UCLouvain

École polytechnique de Louvain

# Vagus nerve stimulation therapy outcome prediction using brain connectivity estimators

Author: **Igor Teixeira Casatti**
Supervisors: **Michel Verleysen, Riëm El Tahry**
Readers: **Marcos de Sales Guerra Tsuzuki, Dounia Mulders**
Academic year 2021–2022
Master [120] in Data Sciences Engineering

# Acknowledgements

*I would link to thank Michel Verleysen and Riëm El Tahry for guiding me through this work. Having them as supervisors allowed me to learn a lot about how a scientific work should be conducted.*

*I would also like to thank Simone Vespa for sharing his knowledge about functional connectivity and its applications related to VNS therapy. It was an incredible experience to work with someone who shows so much interest and knows so much about the subject.*

*Finally, I would like to thank Venethia Danthine for collecting the EEGs and for helping me with the data preprocessing and clinical information. This master's thesis would not be possible without her work.*

# Abstract

Epilepsy is a neurological disorder that affects millions of people worldwide. The most common treatments for epilepsy are the use of antiepileptic drugs (AEDs) and surgery in more severe cases, but these treatments may not be sufficient to treat some patients and then vagus nerve stimulation (VNS) therapy can be used as an add-on treatment to reduce seizures. However, a question that arises with the use of this therapy is the prediction of patient response to treatment, which would avoid unnecessary implantation of VNS in non-responders. In this master's thesis we used functional connectivity measures in combination with graph measures and machine learning models to predict the response of patients to the therapy. Although the study of the influence of the VNS on functional connectivity is not new, this work is distinguished by using only electroencephalograms recorded before VNS implantation and a wider range of functional connectivity and graph measures. After calculating the measures in a population of 37 patients, we found significant differences ($p < 0.05$) between the global efficiency, average clustering coefficient, and modularity of responders and non-responders, indicating that these measures might work as biomarkers to predict the response to the therapy. Moreover, using a subset of the measures as features we were able to train a machine learning model that achieved a validation accuracy of 0.87 and a test accuracy of 0.86.

# Contents

# Acronyms

**AED:** antiepileptic drug
**CC:** clustering coefficient
**DA:** degree assortativity
**DTF:** directed transfer function
**EEG:** electroencephalogram
**GE:** global efficiency
**GP:** Gaussian process
**GRC:** global reaching centrality
**K-NN:** K-nearest neighbors
**MEG:** magnetoencephalography
**MVAR:** multivariate autoregressive model
**PDC:** partial directed coherence
**PLI:** phase lag index
**SVM:** support vector machine
**wPLI:** weighted phase lag index

# 1 Introduction

Epilepsy is a chronic neurological disorder that affects 50 million people worldwide, which makes it one of the most common neurological disorders in the world. Epilepsy is characterized by recurrent seizures, that are episodes of electrical disturbances in the brain, causing changes in behavior, feelings, or involuntary movements. The duration and intensity of the seizures can vary from brief lapses of attention and small muscle jerks to long episodes of convulsions. Also, the frequency of these episodes can vary from less than one seizure per year to many per day [1].

The main treatment for epilepsy is the use of antiepileptic drugs (AEDs) by patients. This treatment is the most common and can control the seizures of approximately 70% of the patients that have epilepsy, making them seizure free. However, some patients have drug resistant epilepsy, also known as refractory epilepsy, that cannot be treated or controlled with medication [1]. In that case, another options of treatment emerge, like surgery or vagus nerve stimulation (VNS) therapy.

The VNS therapy is a low-risk surgical option for patients with refractory epilepsy. This treatment consists in a device formed of two parts, the stimulator (also called generator) and the wire (also called lead), that are placed during a surgical procedure. The wire is wrapped around the vagus nerve and is connected to the generator, so the device works by sending regular electrical stimulation to the vagus nerve and consequently stimulating areas of the brain. This procedure is recommended to patients who have not had success with drug treatment or surgery [2].

Nonetheless, the vagus nerve stimulation therapy is considered as an add-on treatment, because the use of AEDs is not interrupted after the implantation, that is, the VNS is used jointly with the treatment with medications. Moreover, this treatment does not cure epilepsy, but works reducing the frequency and the intensity of the seizures [2].

One difficulty that arises with this treatment is to predict whether a patient will have a positive outcome with the therapy, in short, predict if the VNS implantation will reduce the frequency or the intensity of the seizures. The outcome prediction would allow a prior evaluation and avoid unnecessary implantation of VNS in patients who do not have a good response to the therapy. But this prediction proved to be a hard task, mainly because VNS mechanism of action is still unknown and the way it controls the seizures is still discussed in the literature [3].

However, while there is no consensus on how the VNS works in the brain, many researches suggests that it may work by changing the functional connetivity of the brain and modulating its synchrony [4, 3]. Consequently, many works in the field suggested the use of functional connectivity measures in combination with network analysis to explain the action of VNS in the brain and how the outcome of this therapy could be predicted.

The functional connectivity measures aim at estimating the relation between the signals of an EEG and the information flow in the brain. In short, these measures find relations between the EEG channels and in that way can estimate the connectivity between the brain regions. In sequence, with the estimated connectivity between the regions it is possible to create a network that summarizes and explain the interaction between them. Finally, with those networks it is possible to extract network measures and consequently analyze the EEG of the patients using a diversified set of quantitative measurements derived from the graph theory.

With the recent advances in the fields of information theory and computer science, the interest in representing the behavior of the brain using those networks increased significantly in the last years, with this technique being applied in the study of Alzheimer's disease, schizophrenia and epilepsy. In that context, many works studied the network changes induced by VNS and how they could explain the action of this therapy in epilepsy treatment. But it is important to highlight that there are many ways to estimate the functional connectivity of the brain, because many methods have already been proposed, with each one generating a different result. Then, the process of estimating the functional connectivity for a specific application involves the selection of the most appropriate methods among a wide range of options.

Apart from the selection of the most appropriate functional connectivity estimators, it is also necessary to explore the connectomes using a wide range of graph measures. For each connectome, many network measures might be used to generate a large set of quantitivative measures (features) and some of them might work as important variables to predict the VNS therapy outcome. In the other hand, these features alone might poorly explain the outcome, but have a great

prediction power combined and that is where the use of machine learning models can be useful. With the use of machine learning models a set of measures can be used together to predict the outcome of the VNS therapy and this can significantly increase the accuracy of the predictions.

## 1.1 Objectives of this work

In this master's thesis we explored the use of functional connectivity estimators in the sensor space in combination with graph measures in the prediction of the outcome of VNS therapy. We assessed the power of these measures in two ways, first we analyzed their predictive power alone, using a statistical test to compare differences between responders and non-responders to the therapy, and in sequence we trained machine learning models to asses the capacity of the set of measures in predicting the response to the treatment.

While analyzing the variables derived from connectomes and functional connectivity we looked for biomarkers, which are variables significantly different in responders and in non-responders to VNS therapy. In other words, biomarkers are features that might work as good classifiers of patients as responders and non-responders.

In brief, to perform this study we used interictal EEG recordings (without seizures activity) of patients before the VNS implantation and looked for indicators that could predict the response to the VNS therapy. For each patient we had two EEG recordings, one made with the patient awake and another one with the same patient sleeping. Then, with each EEG we defined connectomes in different frequency bands and extracted graph measures from them. Additionally, in this work we estimate the brain connectivity using three different functional connectivity measures: direct transfer function (DTF), partial directed coherence (PDC), and the weighted phase lag index (wPLI). In relation to the graph measures, we calculated five for each connectome: global efficiency (GE), average clustering coefficient (Avg. CC), modularity, degree assortativity (DA), and global reaching centrality (GRC).

Consequently, one of the objectives of this work is to propose a pipeline to extract features based on functional connectivity and network measures using EEGs. Furthermore, the other objective consists in to explore the predictive power of these features in classifying responders and non-responders to VNS therapy. In order to do that we adopted a methodological approach, comparing the most used methods in the literature and the results we obtained with them. Then, with this work we expect to find a pipeline configuration that works well in that task and to find

variables that can classify responders and non-responders with a good performance.

This manuscript is organized in seven chapters, counting with this introduction as the first chapter. In the second chapter we bring a brief introduction to the concept of functional connectivity and connectomes, the uses of it and some points that are still discussed in the literature. In the third chapter we discuss works related to VNS therapy outcome prediction and why this work extends what has already been done. In the fourth chapter we talk about the materials used in this work, including the patients data and the software and scripts used. In the fifth chapter we present the methods used in the pipeline and in the machine learning application, justifying the choices that were made and bringing examples to elucidate the use of the methods. In the sixth chapter we assess the performance of the features obtained from the pipeline and the machine learning models that were trained. In the seventh chapter we elaborate on top of the features that we found to be relevant in the classification problem, finding support on the literature about why they could be important. Finally, the work finishes with the conclusion, where we summarize the results obtained, the conclusions we drew from this work and points that could be improved and explored in future works.

# 2 Introduction to connectomes

It is known for a long time that the brain structure is formed by a complex network, that is thought to provide and explain the information flow within the brain. With the significant increase in computational power and the expressive growth of the area known as network science in the last decades, the interest in studying the behavior of the brain using graphs and networks has also increased.

Networks, or graphs, are composed by a set of nodes where some pairs of them are linked together with edges. The nodes may represent objects, entities, or many other things, and the edges a possible interaction or relation between them. For example, social interactions between people can be represented by social networks, where nodes represents individuals and edges a social relation between them.

The representation of phenomena as networks may be very insightful in many ways, because it allows the calculation of network properties, that is, quantification of many characteristics and structures of networks. Hence, by defining networks, one may be able to calculate measures based on the characteristics of the networks, which may be used in a posterior quantitative analysis.

For brains, those networks (also called connectomes) represent the relation between brain regions or neurons, where the edges represent physical or functional relation between those. Like for other networks, the creation of connectomes allow the calculation of graph measures, that make possible the study of brain phenomena. Therefore, with the popularization of this method, connectomes have already been used as tool to understand behaviors, mental or neurological disorders, and neurodegenerative diseases. However, although many studies have already been carried out on the connectivity of the brain, many findings still remain unclear and do not seem to fully describe the behavior of this organ [5].

One of the biggest discoveries in the field of brain connectivity, is the fact that the phenomenon of “small-worldness” observed in other kinds of networks

was also perceived in researches on brain networks. Although a little bit simple, this phenomenon tells that the brain networks tend to be organized in clusters of strongly connected components and also that the expected distance between two random chosen nodes grows proportionally to the logarithm of the number of nodes. Moreover, apart from this kind of organization, many other graph measures and properties may be studied from a functional or structural network, such as modularity, centrality, hierarchy and the distribution of hubs [6, 7].

Before proceeding with the interpretation of networks measures in brain connectivity, we need to distinguish the two different kinds of networks that can be defined for the brain: structural networks and functional networks (also known as functional connectomes). Structural networks are defined and extracted from histological or imaging data, where regions of interest can be defined as nodes and edges as the probability of connection between two regions. On the other hand, functional networks are directly extracted from time series data, where the measuring regions are defined as nodes and the coupling between the signals defines the weight of the edges [5]. The figure 2.0.1 represents the differences in the creation of functional and structural networks. Given the context and the purpose of this work, we will focus our analysis on the functional networks.

The functional segregation is an important ability of the brain networks that is directly derived from the "small-world" phenomenon, which states that brains specialized processing occurs within densely interconnected regions. The existence of clusters in anatomical networks imply in the existence of functional segregation within the brain, what in turn indicates segregated neural processing. In order to measure the functional segregation of brain networks, one may take advantage of segregation measures, that quantify the presence of clusters or modules in networks [8]. Figure 2.0.2 shows examples of modules and triangles in networks.

The second ability that is important in the study of functional networks is the functional integration, that characterizes the brain ability to rapidly combine information from different brain regions. The measures capable of quantifying this characteristic are based on the path length between the nodes, as shown in the figure 2.0.2, and the most known are the characteristic path length and the global efficiency [8].

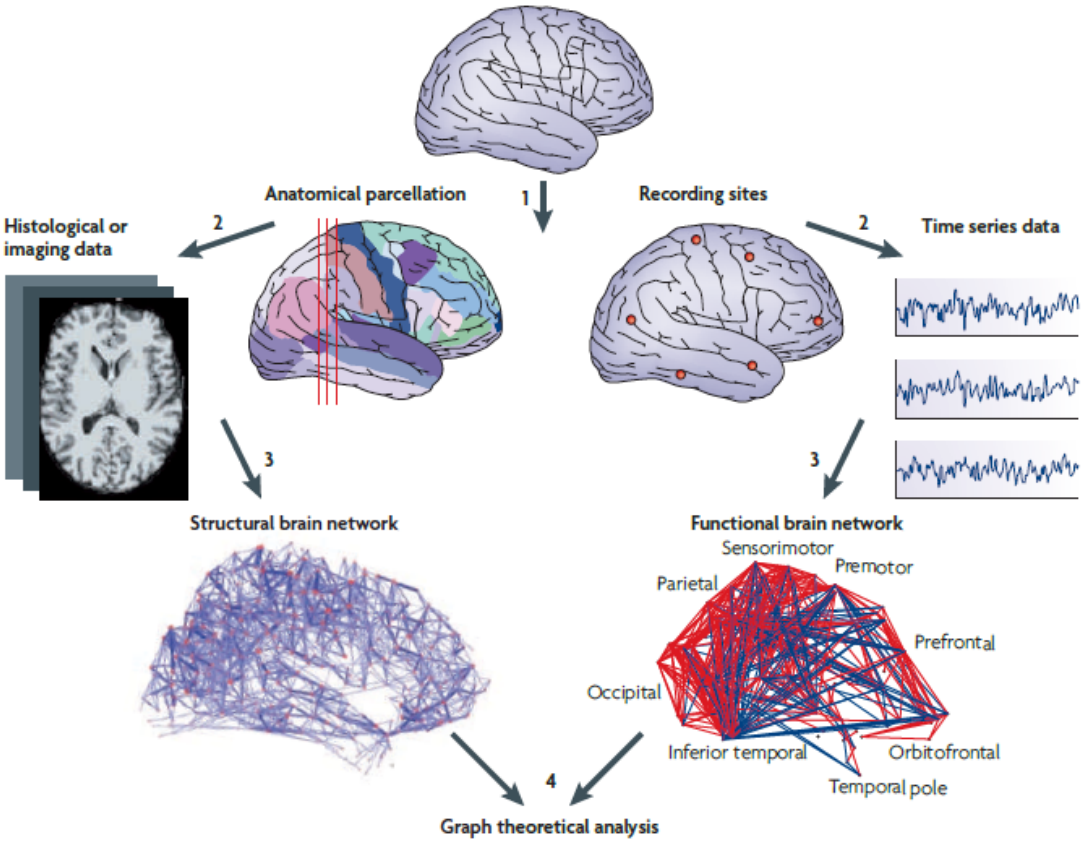


Figure 2.0.1: Representation of the creation of structural and functional networks. Extracted from [5]

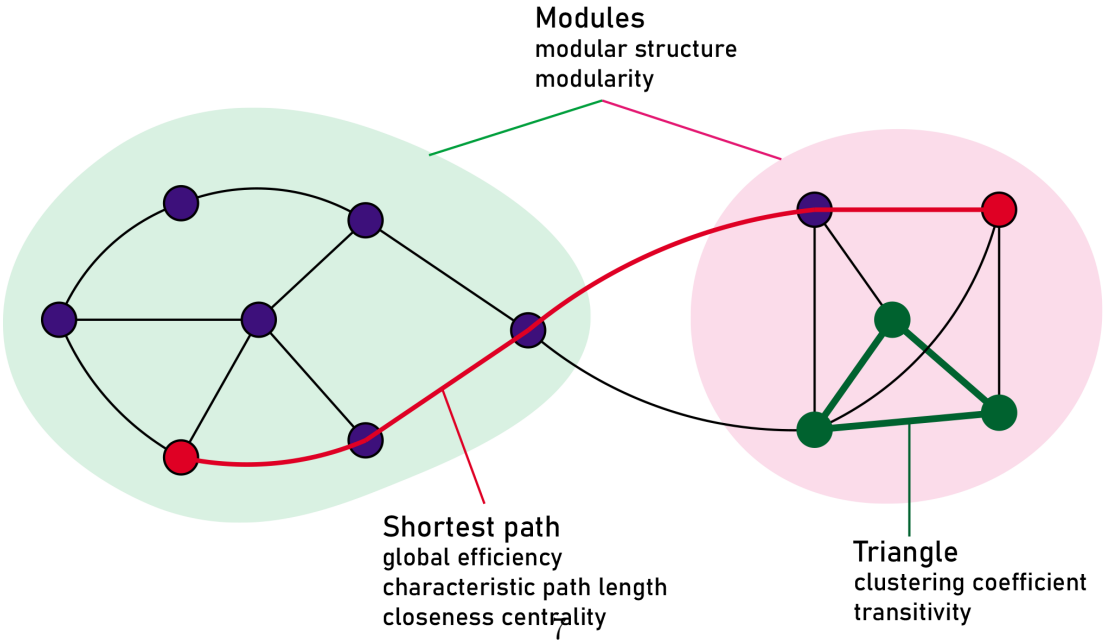


Figure 2.0.2: Examples of a triangle, a shortest path and modules. Adapted from [8]

A third kind of characteristic to be measured in functional networks is the resilience, which quantifies the vulnerability of networks to external influences. For example, in the Alzheimer's disease the severity of the functional deterioration is directly linked to the capacity of anatomical networks to resist to degenerative changes caused by the disease. The two main measures of resilience are the degree distribution and the assortativity [8]. The figure 2.0.3 shows an example of degree calculation in a directed network, where all the edges in red can count for the degree (in-degree and out-degree) of the highlighted node.

A fourth characteristic important for functional networks is the centrality of the nodes, which portrays the role and the existence of central nodes (also referred as hubs) in those networks, as shown in figure 2.0.3. Important nodes may play a determinant role in the resilience of networks and in their response to external stimuli or to degradation. Thereat, measures of centrality quantify the importance of a node to a given network, and the most common measures are the degree of the node and the betweenness centrality. The biggest part of the measures of centrality take into account the participation of the nodes in shortest paths, that is, the amount of shortest paths that pass through each node, which quantifies the importance of each node in the functionality of the network [8].

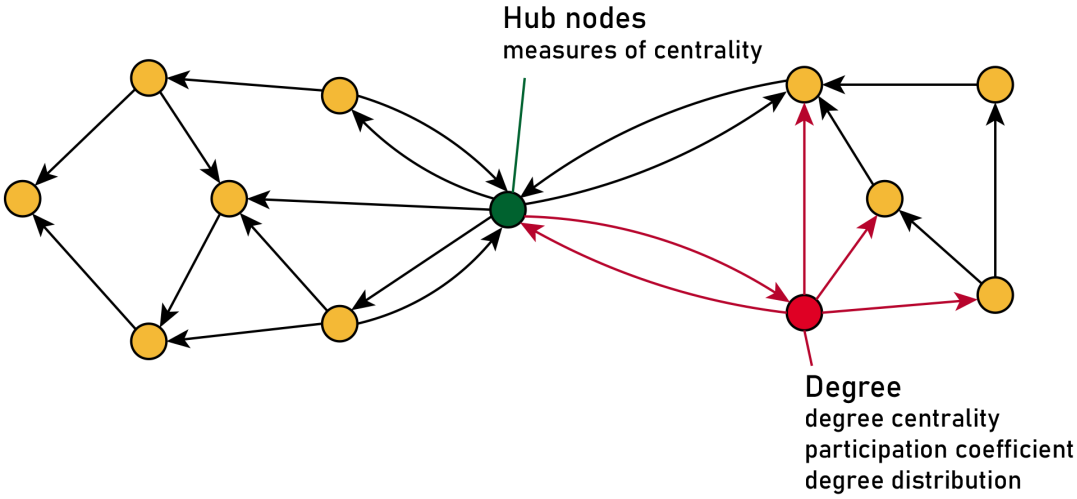


Figure 2.0.3: Examples of a hub node and degree of a node. Adapted from [8]

The visual inspection of the connectomes is also a possibility, but in the cases where one is dealing with big networks or a large amount of patients, this might be a challenging task. Hence, the graph measures based on the characteristics and abilities of brain networks are useful in the search of patterns or structures that

are common to a group of patients. Those patterns and common structures can work as biomarkers to diagnose diseases or to predict the response of treatments.

## 2.1 Building functional networks

Many ways were already proposed to generate functional networks from EEGs, where the simplest ones are based in calculating the Pearson correlation coefficient or the mutual information between pairs of signals and the most complex ones based on the concepts of Granger Causality or phase synchronization. Furthermore, it is important to highlight that different functional connectivity estimators lead to different resultant networks, so different estimators can lead to completely different results [9]. Moreover, those estimators can be classified and characterized with respect to the information they can retrieve from the relation between the signals.

The first classification that can be made for the estimators regards the domain where the coupling of the signals is given, that can be in the time domain or in the frequency domain. On the time domain the coupling between the signals is defined for every time point and the relation of the signals is as a function varying over the time. On the frequency domain the coupling is estimated as a function of the frequency, that is, for each frequency point the signals have a defined coupling.

In sequence, the estimators can be classified with respect to the linearity or nonlinearity of the relations measured between the signals. A linear estimator measures linear relations between signals, always using the amplitude of them in the estimation. On the other hand, non-linear estimators look for non-linear relation between signals and can use the amplitude or the phase of the signals in the coupling estimation.

The third classification is somewhat related to the linearity of the estimators, because it concerns the property of the signals that is used to estimate the coupling, that can be the phase or the amplitude of the signals. The estimation based on the amplitude looks for joint oscillations or changes in signal amplitudes to estimate the relation between signals, while the estimation based on the phase uses the difference between the phase of the signals to estimate the coupling.

The fourth classification is related to the way the measures are formulated and calculated, because the measures can be defined for a pair or to the whole set of signals. The measures defined for a pair of signals are bivariate and can only estimate the coupling of signals two by two, finding only direct coupling between signals. The measures defined for a set of signals are multivariate measures and estimate the coupling between all the signals in the set at once, what make possible

the finding of indirect coupling between them.

Finally, the last classification concerns the information retrieved by the coupling estimation, that can be only the intensity of the coupling or the direction of the coupling too. The undirected measures can only estimate the intensity of the coupling, so the connectomes generated with those measures have undirected edges. The directed measures estimate the intensity and the direction of the coupling, so they can be used in the creation of directed networks.

Those different characteristics of functional connectivity estimators comes from the way they are calculated and derived from another metrics. Consequently, depending on the formulation and how the measure is calculated it can have a different set of features. Then, according to the formulation, the measures can be classified as: measures derived from correlation, phase synchronization measures, information theory measures, and the measures based on the Granger causality.

The simplest functional connectivity defined for a set of signals is the Pearson correlation coefficient, so this type of measure is pretty straightforward and measures the instantaneous correlation between two signals according to their amplitude. A variant of this measure is the cross-correlation, that calculates the similarity between two signals based on their displacement, making this estimation a good fit to quantify the relation between two signals that are time shifted. As an alternative, the cross-correlation can also be defined in the frequency domain, where it receives the name coherency and measures the coupling between two signals at a given frequency [9, 10]. Those are the main measures derived from the correlation and have as main characteristic the capacity of estimating only linear relation between signals.

Another way to compute the functional connectivity is by exploring the phase relation between signals, those measures are known as phase synchronization measures and the most known measures of this kind are the phase locking value (PLV) and the phase lag index (PLI) [9]. Those measures use the phase of the signals to estimate the coupling between them, that is, the phase of the signals is estimated and then it is used to find coupling between pairs of signals. Phase synchronization measures are able of finding nonlinear relation between signals and are bivariate (can only study the relation of signals two by two). One important fact is that this type of measure is of paramount importance for the study of functional networks in epilepsy, because epileptic networks are characterized by a high level of synchronization [4].

A different group of functional connectivity estimators are the ones originated from the information theory, like the mutual information and the transfer entropy

[9]. Those estimators differ from the fact they can find nonlinear coupling between signals by using their amplitude. Between those metrics, the mutual information has already been used in many application for its capacity in measuring not only linear relations between signals but also nonlinear ones [11].

The last group of functional connectivity estimators are the ones based in the concept of Granger causality. Most part of those measures are based on time varying autoregressive models and multivariate autoregressive models in which a data point can be calculated as combination of previous samples [9]. Those estimators distinguish themselves from the last ones for the fact they consider the whole set of signals in the computation of the connectivity, not only a pair of signals like the measures mentioned previously. Then, the estimators based on Granger causality are multivariate measures and are able of finding indirect information flows (cascade flows) in a set of signals [12].

The table 2.1.1 summarizes the characteristics of each connectivity estimator.

| | Undirec. | Direc. | Bivar. | Multivar. | Ampl. | Phase. | Lin. | Nonlin. | Time. | Freq. |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correlation** | X | | X | | X | | X | | X | |
| **Coherency** | | X | X | | X | | X | | | X |
| **Cross-correlation** | | X | X | | X | | X | | X | |
| **Directed coherence** | | X | | X | X | | X | | | X |
| **Directed transfer function** | | X | | X | X | | X | | | X |
| **Granger causality index** | | X | X | | X | | X | | X | |
| **Mutual information** | X | | X | | X | | | X | X | |
| **Partial coherence** | X | | | X | X | | X | | | X |
| **Partial directed coherence** | | X | | X | X | | X | | | X |
| **Phase locking value** | | X | X | | | X | | X | X | |
| **Phase lag index** | X | | X | | | X | | X | X | |
| **Transfer entropy** | | X | X | | X | | | X | X | |

Table 2.1.1: Table summarizing the characteristics of each connectivity estimator. Each column represent a set of antagonist characteristics. Table adapted from [9].

However, as mentioned before, the functional connectivity estimators can produce really different results, thus the connectomes generated with each one of these estimators can be really different. In the literature, it is possible to find references and uses of all those estimators in many applications. Then, the chose of the best estimators is usually subject to the application of the functional connectivity estimation, that is why there are some references in the literature comparing the performance of those measures in different applications.

Moreover, after using the functional connectivity estimators in a set of EEG channels one should obtain a measure of the coupling between each pair of channels, which can be used later to build a connectome. However, the process of transforming the coupling between channels into a connectome may need an intermediate processing step, the thresholding of the coupling in order to calculate the edges of the connectomes. This processing step might be important to remove spurious and non-relevant connections that can lead to wrongful results. However, in the literature there is no consensus about which method is the most appropriate for each application, so the techniques to do that are still largely discussed [13].

# 3 Novelty of this work

During the last decades many studies have been devoted to searching for biomarkers capable of predicting the efficacy of therapy using VNS in epileptic patients. The first implantation of VNS in humans dates back to 1988, when Penry et al. [14] performed an implantation of VNS to act as an adjunctive treatment in a patient with refractory epilepsy. Since then, many studies have been carried out seeking to understand the mechanism of action of this therapy and how the outcome of this treatment could be predicted.

The first study addressing the response to VNS therapy was published by Penry et al. [14] in 1990, where authors studied and evaluated the response of the first patients to receive the VNS implantation. Nonetheless, this study did not investigate mechanisms and characteristics that could lead to a treatment outcome prediction, only the efficacy of this therapy in a small group formed by four patients.

The main studies looking for predictive factors for the outcome of the therapy came years after the first implantation, after VNS showed to be a reliable method to control refractory epilepsy. Then, in 2001 Schermann et al. [15] published a research based on the results of the implantation of VNS in 95 patients. This paper reported the seizure reduction of the patients and looked for clinical factors that could be possible predictors for the outcome of the VNS therapy. However, this work failed in finding any variable that could explain the outcome of the therapy.

Sequentially, in a study published in 2005, Janszky et al. [16] conducted a research on a population of 47 patients that received VNS implant, where the main goal was to find factors that were able to predict the effectiveness of treatment using VNS. This study used EEG and magnetic resonance imaging data from clinical evaluation of the patients, and the prediction power of variables extracted from those data was analyzed. Consequently, in that work the authors found two variables that could be related to the positive outcome of the VNS therapy, the absence of

interictal epileptform discharges (IED) on EEGs and the presence of malformation of cortical development. A highlight of this work is the discrimination power that the absence of interictal epileptform discharges presented, with a sensitivity of 0.83 and a specificity of 0.80.

Concerning the type of epilepsy and the seizure focus, in 2011 Burakgazi et al. [17] conducted a retrospective study comparing the seizure focus of patients and the response they presented to the treatment. That work concluded that VNS was more effective in patients with frontal lobe epilepsy than in patients with temporal lobe epilepsy. In this study the outcome of 46 patients was analyzed and it was found that 65% of the patients with frontal lobe epilepsy (FLE) and 15% of the patients with temporal lobe epilepsy (TLE) had a satisfactory seizure reduction, and a hypothesis test confirmed a difference in the outcome of patients with TLE and FLE.

More recently, Bràzdil et al. [18] performed a study where the reactivity of EEG to external stimuli was used to evaluate the response of epileptic patients to VNS therapy. In that work the authors performed a power spectral analysis on pre-operative EEG and calculated the changes induced on this measurement by different stimuli (eyes opening and closing, photic stimulation, and hyperventilation). Following, the team used the variables obtained from the measurements as features for machine learning models and managed to obtain a logistic regression classifier that achieved an accuracy of 86.7%, a sensitivity of 88.6%, and a specificity of 84%. This work showed that a good prediction accuracy could be achieved through the combination of features extracted from EEG recordings.

However, despite the success of the aforementioned studies in finding variables capable of predicting the outcome of VNS therapy, many researches were also done in parallel with those, also with the intention of looking for variables able to predict the outcome of therapy with vagus nerve stimulation, but with features derived from functional connectivity. This is mainly because epilepsy may be seen as a functional disorder of the brain, associated with high synchronization of neurons, leading the brain to an hypersynchronous state. In addition, the main mechanism of action of VNS may also be associated with the functional connectivity of the brain, as it is believed that this therapy treats epilepsy by promoting desynchronization and changing brain connectivity to a less synchronous state [19, 20].

In this line of research, in 2013 Fraschini et al. [3] conducted a study to find variables capable of predicting the outcome of this therapy. In that study the group of researchers used the phase lag index (PLI) to compare the synchronization of responders and non-responders before and after the VNS implantation. In that work the synchronization measure was used to measure the average synchronization

of EEG channels in the frequency bands of interest, working as a global estimator of synchronous brain activity. Thereafter, in this research it was found that VNS induced desynchronization in the gamma band was significantly different between responders and non-responders, where responders had a smaller average synchronization after the VNS implantation.

In another work about VNS therapy, Babajani-Feremi et al. [21] used the phase locking value (PLV) to build connectomes from rs-MEG and performed graph analysis to explore the topography of the networks of responders and non-responders to the therapy. Moreover, like in the work mentioned before, the networks were built for each frequency band of interest, so the topology of the graphs was explored on each one of them. In that work it was found that responders to the therapy had lower transitivity and characteristic path length and higher modularity in alpha, beta and theta bands when compared to non-responders.

In a more recent work, Vespa et al. [4] also explored the VNS-induced changes in brain functional connectivity as possible biomarkers for VNS therapy outcome prediction. In that paper the authors explored the mean synchronization and the global efficiency in the frequency bands of interest using EEGs recorded in sleeping state and in wakefulness state. In order to do that the researchers measured the average synchronization using the weighted phase lag index (wPLI) and measured the global efficiency from connectomes generated with the partial directed coherence (PDC). This research reported significant differences between responders and non-responders in the theta band for the VNS-induced desynchronization and global efficiency decreasing on sleeping state EEGs.

This master's thesis aims to extend the researches mentioned above about VNS therapy outcome prediction, but distinguish itself from those works in some key aspects. First of all, this work uses a broader set of functional connectivity measures and look for biomarkers in the connectomes generated with each one of them. By doing that we have as objective compare which functional connectivity estimator can lead to more enlightening results in the VNS therapy outcome prediction. Moreover, we test multiple datasets in machine learning algorithms to assess the discriminating power of those measures in the classification task of responders and non-responders to the therapy.

In second place, this work offers a methodological approach about the creation and thresholding of connectomes, because the thresholding procedure of connectomes is still discussed in the literature and different techniques may lead to different results given a task. Consequently, this work differs itself in this aspect by providing a comparison between three setups for building connectomes and comparing the results obtained with each one of them.

Furthermore, this work also explore the use of machine learning models in the VNS-therapy outcome prediction. Although machine learning models are already extremely diffuse in the literature, in this work we adopt a different approach by testing different feature selection methods and machine learning models and comparing the features that are more important for each one of them. This approach has as objective to compare the features that are more important for the machine learning models with the biomarkers found.

Finally, another important aspect of this master's thesis concern the data we used for the patients, because in this study we performed a retrospective study and calculated the functional connectivity measures using only pre-implantation EEG recordings. Hence, from the point of view of a clinical application, the method proposed by this work makes more sense, as it would allow the evaluation of patients with just simple EEGs, without any kind of implantation or stimulation.

In short, this work differs itself from the other works in the literature by the use of only pre-implantation data and by its methodological approach, where we compare different functional connectivity measures in the EEG connectivity estimation, different thresholding methods in the connectomes generation, and different feature selection and machine learning models in the outcome prediction task.

# 4 Materials

## 4.1 Patients data

The data used for this study consisted in anonymized restropective EEG data from 37 patients, recorded at "Cliniques universitaires Saint-Luc" and "Centre hospitalier neurologique William Lennox" in Belgium. The EEG recordings were made using the 10-20 system and between 19 and 23 electrodes were available for each patient. To generate connectomes with the same number of nodes for all the patients we considered only the 19 electrodes placed in the scalp for all the patients, discarding the channels obtained from other electrodes. The 19 electrodes that were considered are shown in the figure 4.1.1.

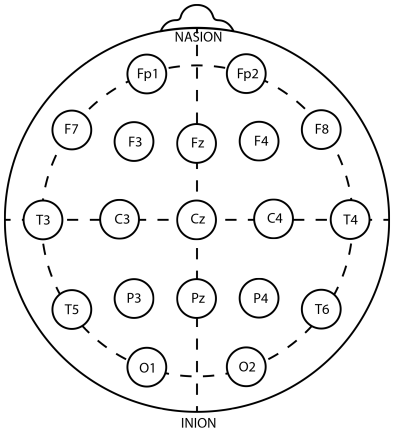


Figure 4.1.1: Electrodes considered for the EEG recordings.

Patients who showed a seizure reduction of at least 50% after VNS implantation were classified as responders, otherwise they were classified as non-responders. Then, from the 37 selected patients, 22 were classified as responders and 15 as

non-responders to VNS therapy. The records were performed before the start of VNS therapy and for each patient two records of 60 minutes were available, one performed in sleeping state (with the patient asleep) and another one performed in wakefulness state (patient awake). The data was treated by neurologists, that selected for each EEG recording 10 epochs of 10 seconds without ictal activity. More details about the patients data can be found in table A.0.1 in the appendix.

## 4.2 Software used and implementation details

All the codes and scripts used in this master's thesis were written in the Python programming language. The scripts and codes were written on top of the MNE-Python library [22], as extensions to the functions that were already implemented in this package. MNE-Python is an open source library that allows users to load, save, preprocess, visualize and perform operations with human neurophysiological data (EEG, sEEG, MEG, etc.). The first main advantage of this library is the availability in Python, so the data structures of this library can be combined with the others libraries that are available in Python, which is useful for complex applications. The second advantage of this package is the integration of this package with libraries that extend the basic functionalities of it, like the set of libraries called MNE-Tools.

MNE-Python allows the users to load EEG data directly from `.edf` and many other file formats that support annotations and storage of recording information. The data we used was recorded in `.edf` format, so this functionality was used to load the data and access the events annotations made by the neurologists. In sequence, with the events annotation we could easily transform the whole EEGs in epoched EEGs with the functions of MNE. In addition, all the EEG data structures of MNE-Python allow the users to retrieve information such as the sampling rate and the channels name of the EEGs, which turned out to be really useful in the implementation of the additional functions we made, because we can pass just one data structure as main argument and retrieve all the necessary information from it. Moreover, the use of MNE also made possible the use of the spectral connectivity functions made available in the connectivity extension of MNE-Tools, that we used to calculate the synchronization measures (wPLI) for the patients.

To calculate the multivariate functional connectivity measures we used the Python library named Source Connectivity Toolbox (SCoT) [23]. Although this library computes the connectivity in the source space, we adapted the functions to compute the connectivity in the sensor space. The main advantage of this library lies in the routines to fit the multivariate autoregressive models, that are fast and numerically stable. Moreover, this library also includes functions to generate

surrogate data, that were useful to perform non-parametric statistical tests in the connectivity measures, like the thresholding using the surrogate data test. However, this library does not directly support the data structures from MNE-Python, so some adaptions were needed to integrate both libraries.

After calculating the connectivity between EEG channels in all the frequency bands, we transformed the connecitivity measures into connectomes. In Python the connectomes were created with the library NetworkX [24], that support the creation and operation with graphs. This library is made for general purpose use, so some modifications were needed to use the connectivity matrices with this library. The biggest advantage of this library is that it has a lot of graph measures already implemented, which made the task of calculating multiple graph measure easier. With exception of the modularity, that was computed with the library iGraph [25], all the other graph measures were calculated with the implementations available in NetworkX.

The visualization of the connectomes and other plots were made with scripts and routines that we wrote using the libraries Matplotlib [26] and Seaborn [27]. Those libraries do not support the plot of networks, so the scripts to plot the connectomes were implemented by us using some resources available in that libraries.

Following, for the statistical tests and biomarkers recognition, we used the libraries SciPy and Statsmodel [28]. Scipy and Statsmodel have a lot of statistical tests already implemented that allow the users to use them using high level commands. The Statsmodel library was used for the Mann-Whitney U test and FDR correction in the bioamrkers identification and Scipy was used for the other tasks.

Finally, for the machine learning part, we used the library Scikit-learn [29] to run the models and perform the model selection. Scikit-learn is a library designed for machine learning, that counts with the implementation of many routines for it, like machine learning models, feature and model selection routines, and functions to assess the performance of the models. The main advantage of this library is that it allows the use and training of various machine learning models using simple high level functions, which makes the test of many different models easy.

# 5 Methodology

As mentioned in the introduction, the main objective of this master's thesis relies in the extraction of measures from EEGs using functional connectivity estimators and graph measures. Then, in this section we discuss and present the methods used in the pipeline we used to extract the features from the EEGs and the methods used in the machine learning task.

## 5.1 Data preprocessing

The data used on the pipeline was already recorded, so the first step of the whole pipeline consists in opening the data, preprocessing it and dividing each raw EEG into 10 epochs. The recorded data was saved in the format `.edf` that can be loaded in Python with the library MNE-Python. The used file format supports annotations of events, so each raw EEG also counted with annotation that were used to divide the data into epochs.

### 5.1.1 Signals re-referencing

After opening the data, we re-referenced the EEG channels to a common average. This preprocessing step is necessary because the EEG signals are measured across the scalp with respect to a reference electrode, so activity detected at the reference electrode may influence the signal measured at the other electrodes. In a perfect condition, the signals would be measured against a common perfectly neutral average, but that is impossible in a live recording situation, so the common method used is to re-reference the signals offline to a common reference that may reduce the noise induced by the reference electrode [30].

The method of re-referencing the EEG electrodes to a common average is largely applied in many biomedical applications, including in the studies of brain connectivity. Despite of counting with the strong assumption that the set of

electrodes capture the whole electrical activity of the brain, this method already showed great results in the analysis of brain connectivity, that is why we re-referenced the EEG signals using this method [31, 30].

### 5.1.2 Filtering

The filtering procedure is a critical step into the pipeline, because it is an essential step to remove undesired noise and frequencies, but at the same time the application of badly designed filters can induce errors in the results of connectivity estimators. Furthermore, it has already been shown that multivariate causality measures can be severely affected by data preprocessing and that small time shifts between time series can introduce error in the connectivity estimations [32].

Consequently, to avoid the introduction of erroneous estimations into our calculations, we opted for a simple filtering setup, only band-pass filtering the signals with a phase invariant filter. The lower and upper cutoff frequency of the filter were set to 0.5 Hz and 30 Hz respectively [4] and the filter was designed with a second order Butterworth filter. In order to make the filter phase invariant, we applied the filter forward and backward, which doubled the resulting order of the filter used.

## 5.2 Phase synchronization measures

Some of the features that will be extracted from the proposed pipeline will be based on phase synchronization measures, that are measures that rely on the phase difference between two signals to estimate their coupling. In the context of EEG study, those measures are used to estimate the phase difference between two EEG channels, and by performing this calculation across all pairs of electrodes it is possible to obtain a matrix that describes the global synchronization of the signals.

Many studies have already reported biomarkers to VNS response found with the use of synchronization measures [33, 4, 3], so we decided to include this metric in our pipeline. Furthermore, this choice is also supported by the possible mechanism of action of VNS, which is believed to work by promoting a desynchronization in functional connections. Hence, the features extracted from synchronization estimators might contain useful information to predict the outcome of VNS therapy.

In this work we used a variant of the traditional Phase Lag Index (PLI) [34], the weighted Phase Lag Index (wPLI), as a synchronization measure. The wPLI is a synchronization measure introduced by Vinck et al. [35], known for being more robust to uncorrelated noise sources and more sensitive to detect changes in

phase-synchronization than the non-weighted PLI.

Given a pair of time series $x(t)$ and $y(t)$ defined on $n$ epochs (trials) and their cross-spectral density $S_{xy,t}$ at a determined frequency evaluated on an epoch $t$, we can define the $PLI_{xy}$ between those signals as [36]:

$$PLI_{xy} = \left| \frac{\sum_{t=1}^{n} sgn(\Im\{S_{xy,t}\})}{n} \right| \tag{5.1}$$

with $\Im\{S_{xy,y}\}$ denoting the imaginary part of the cross-spectral density and *sgn* the sign function.

Then, we can define the wPLI as:

$$wPLI_{xy} = \left| \frac{\sum_{t=1}^{n} |\Im\{S_{xy,t}\}| \, sgn(\Im\{S_{xy,t}\})}{\sum_{t=1}^{n} |\Im\{S_{xy,t}\}|} \right| \tag{5.2}$$

It is important to notice that the cross-spectral matrix is defined for a specific frequency, and so do the wPLI and the PLI. Consequently, in order to calculate the wPLI and the PLI over a frequency band, it is necessary to compute those metrics over many frequency bins specified on this interval (a discretization of the frequency interval) and then average the values.

To compute the wPLI in this work we used the function available on MNE-Python, using the multitaper method also implemented on this library to compute the cross-spectral matrix.

In the figure 5.2.1 it is possible to see the connectivity matrices calculated for one patient with the use of the wPLI. In that figure it is possible to notice that the matrices are symmetric, because the wPLI is not a directed measure, that is, $wPLI_{xy} = wPLI_{yx}$

Furtermore, it is important to highlight that in this master's thesis we do not use the wPLI as a functional connectivity estimator to build connectomes, but as an estimator of the synchronization of the EEG channels during the recording. Because of that, to have a global view of the recording, we get the average value from each connectivity matrix obtained, obtaining the average synchronization on each frequency band.

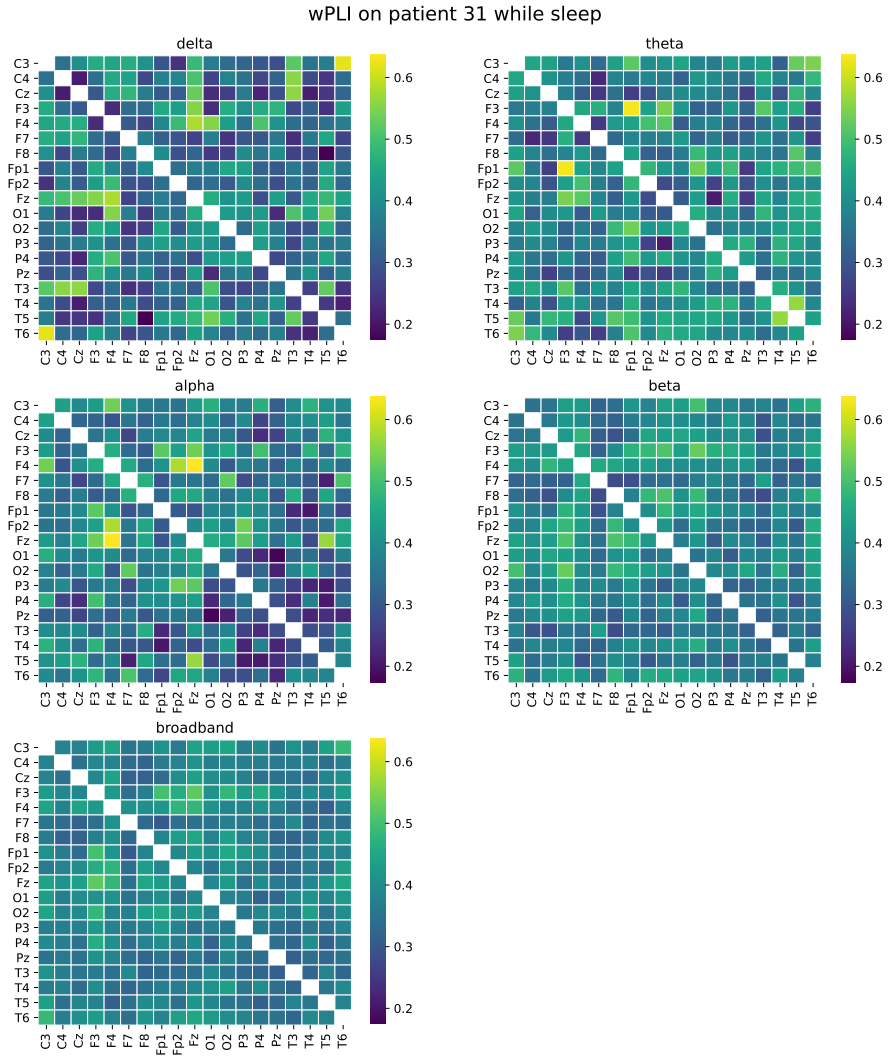


Figure 5.2.1: Connectivity matrices obtained for one patient using the wPLI.

## 5.3 Directed brain connectivity estimators

The directed brain connectivity estimators are important elements of the pipeline proposed on this work, because with those estimators we built connectomes that were used to generate the features based of graph measures. The directed brain connectivity estimators differ from the undirected on the capacity to estimate not only the intensity of the coupling between two signals, but also the direction of the coupling, so with them it is possible to create more informative networks, that contain information about the intensity of the coupling and its direction. Nonetheless, between the directed brain connectivity estimators, it is necessary to choose between bivariate and multivariate estimators.

As already mentioned before, the bivariate methods can only find the relation between a pair of signals, so to create a connectome with this methods it is necessary to compare the EEG signals two by two, finding the connection between all pairs of channels.

In contrast, the multivariate directed connectivity estimators treat all the channels as members of one system, so they are able to find relations between multiple channels at a time. This characteristic gives to them the property of taking into account not only the direct connections between nodes, but also indirect connections [12]. On the context of brain functional connectivity, this characteristic might be a strong advantage in favor of those methods, because they are able to find cascade flows between EEG channels. In the literature, the most common multivariate methods used to analyze the connectivity between EEG channels are the partial directed coherence (PDC) [37] and the direct transfer function (DTF) [38].

Moreover, the PDC and the DTF have already been used in many studies about epilepsy, being used to identify seizure onset zones, epileptic focus, responders to VNS therapy and many other applications on the field [9, 4]. Since both estimators appear frequently on the literature and have already proven to provide important results, in this work we chose to work with both estimators, so we calculated connectomes using DTF and PDC for each frequency band.

### 5.3.1 Multivariate autoregressive models

In order to calculate the estimators chosen, the first step is to fit a multivariate autoregressive (MVAR) model to the channels data. The MVAR model assumes that a data sample over $k$ channels, at a time $t$ can be expressed as a linear combination of the $p$ last data samples (the order of the model) plus an uncorrelated white noise.

So, it is possible to define a data point as:

$$X(t) = [x_1(t), x_2(t), ..., x_{k-1}(t), x_k(t)]^T \tag{5.3}$$

And the MVAR model is defined as:

$$X(t) = \sum_{m=1}^{p} A(m)X(t-m) + E(t) \tag{5.4}$$

Where $A(m)$ is the $k \times k$ matrix with the model coefficients and $E(t) = [e_1(t), e_2(t), ..., e_{k-1}, e_k]^T$ is the uncorrelated white noise vector at time $t$. The order of the model determines how many previous samples will be used for the estimation of the current data point and the matrix $A(m)$ estimates the influence of the previous samples into the current data point. Moreover, each element $A_{ij}(m)$ contains information of the influence of point $x_j(n-m)$ on the current data point $x_i(n)$, so the coefficients matrix provide an estimation of the information flow of the system [9].

Before fitting the model, it is necessary to determine the model order $p$ that will be used on the MVAR model. The most common approach is the estimation of the model order through the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC), defined as [9]:

$$\mathrm{AIC}(p) = \ln|\Sigma_e(p)| + \frac{2pk^2}{N} \tag{5.5}$$

$$\mathrm{BIC}(p) = \ln|\Sigma_e(p)| + \frac{\ln(N)pk^2}{N} \tag{5.6}$$

Where $\Sigma_e(p)$ is the covariance matrix of the residuals, $k$ is the number of channels, $N$ is the number of time samples and $p$ is the order of the model. With those criteria, the model order can be selected as the value that minimizes the chosen criterion [9].

Alternatively, one can set a fixed model order and fit the model using the Ridge regression, optimizing the regularization term. The idea behind this technique relies in setting a model order reasonably high and using the regularization parameter to avoid the overfitting of the model. Consequently, the optimization of the ridge penalty term can be made with the minimization of the generalization error [23].

In this work we opted to use this method to fit the MVAR models, because the Python library used to fit the models (SCoT) had an implementation of this method that had a computational speed superior to the traditional methods mentioned previously.

In sequence to the model fitting, we evaluated the quality of the fits on each epoch and to do that we used the relative error variance (REV) [39], defined as:

$$REV = \frac{MSE}{MSS} \tag{5.7}$$

Where $MSE$ is the mean squared error and $MSS$ is the mean squared signal. This metric should be preferred over the $MSE$ because it is normalized by the $MSS$, what makes the comparison between different datasets possible. The $REV$ values lie on the interval $]0, 1[$ and a value closer to 0 indicates a good explanation of the signals by the model, while a value closer to 1 indicates a poor performance of the model [39]. In our models, the REV ranged from $2 \times 10^{-9}$ to $7 \times 10^{-2}$, mantaining an average of $5 \times 10^{-4}$, showing that the MVAR models describe well the behavior of the systems [40].

### 5.3.2 Directed transfer function (DTF)

The directed transfer function (DTF) was introduced by Kaminski and Blinowska in 1991 [38] as an estimator of the information flow between multiple signals. The first step needed to calculate the $DTF$ is to transform the MVAR model to the frequency domain, so the first step to obtain the $DTF$ is to apply a Fourier Transform to equation (5.4) as follows:

$$E(f) = A(f)X(f) \tag{5.8}$$

Where:

$$A(f) = \sum_{m=0}^{p} A(m)e^{-j2\pi f \Delta t m} \tag{5.9}$$

With $\Delta t$ being the time interval between two samples and $A(0) = -I$.

Then one can rewrite equation (5.8) as:

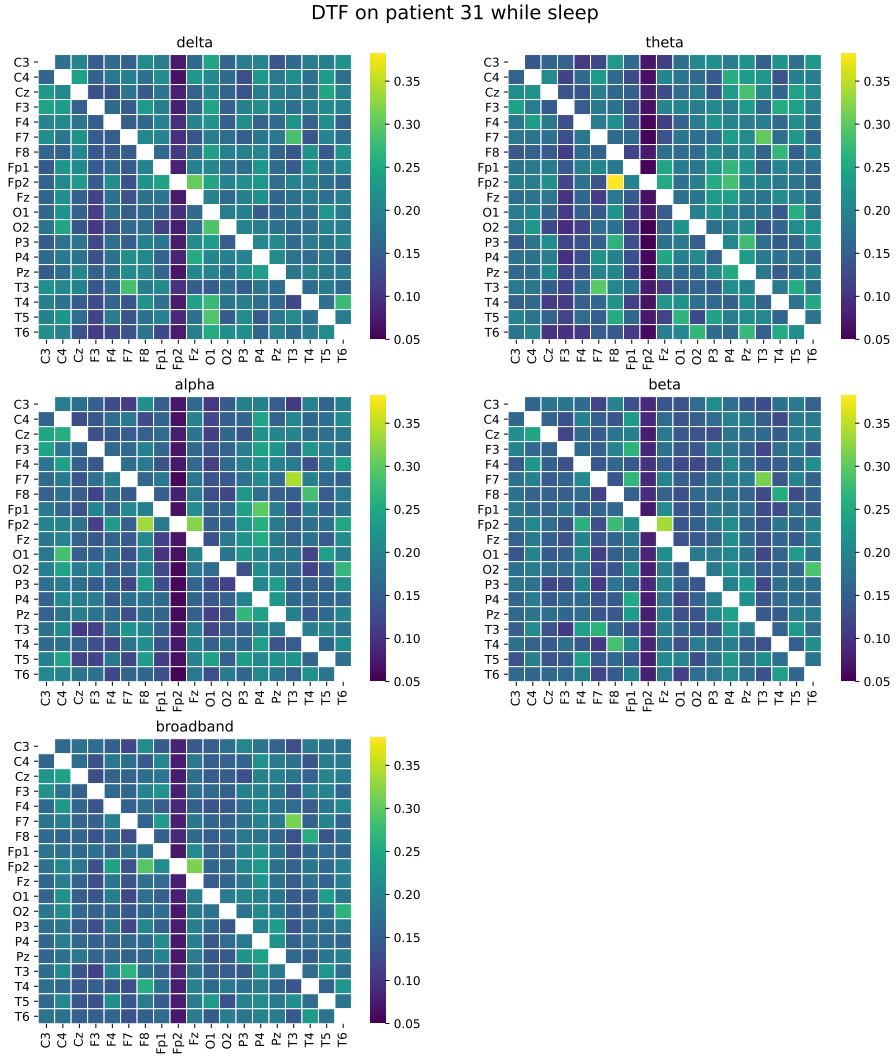


Figure 5.3.1: Connectivity matrices obtained for one patient using DTF. The horizontal axis represent the sources and the vertical the sinks.

$$X(f) = A^{-1}(f)E(f) = H(f)E(f) \tag{5.10}$$

With $H(f)$ being called the transfer function of the system, where each element $H_{ij}(f)$ represents the connectivity between the j-th input and the i-th output at a given frequency $f$.

Finally, the $DTF$ can be defined through the transfer function as:

$$\gamma_{ij}^2(f) = \frac{H_{ij}(f)}{\sqrt{\sum_{m=1}^{k} |H_{im}(f)|^2}} \tag{5.11}$$

With respect to the normalization condition:

$$\sum_{m=1}^{k} \gamma_{im}^2(f) = 1 \tag{5.12}$$

In the figure 5.3.1, it is possible to see the connectivity matrices calculated for one patient. The normalization of the DTF is made with respect to the sources, so it is possible to notice vertical patterns in the connectivity matrices, indicating that one source is similarly coupled to all the sinks.

### 5.3.3 Partial directed coherence (PDC)

The partial directed coherence (PDC) was defined by Baccalá and Sameshima in 2001 [37] and is defined directly from the Fourier Transform of the coefficients matrix of the MVAR model. Given that $A(f)$ is the Fourier transform of the coefficients matrix, as shown in equation 5.9, the PDC can be defined as:

$$\pi_{ij}(f) = \frac{A_{ij}(f)}{\sqrt{\sum_{m=1}^{k} |A_{mj}(f)|^2}} \tag{5.13}$$

With respect to the normalization:

$$\sum_{m=1}^{k} \pi_{mj}^2(f) = 1 \tag{5.14}$$

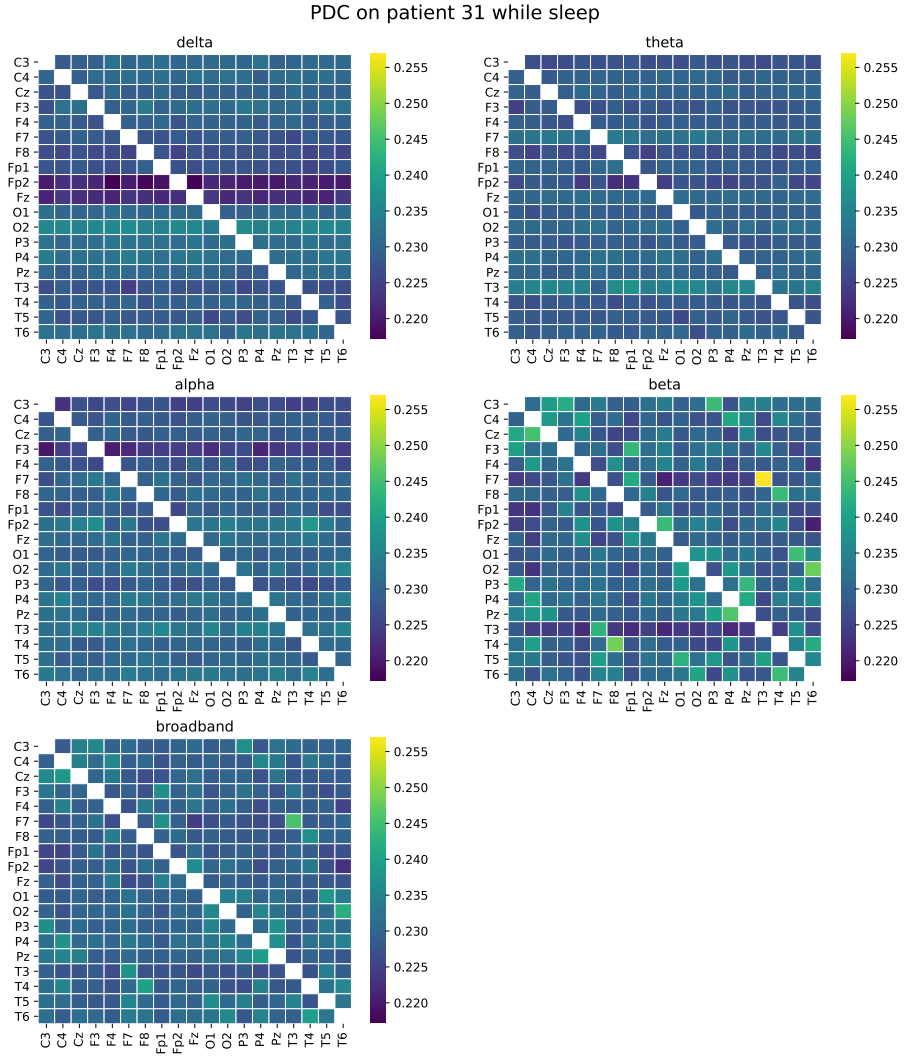


Figure 5.3.2: Connectivity matrices obtained for one patient using PDC. The horizontal axis represent the sources and the vertical the sinks.

In the figure 5.3.2 it is possible to see the connectivity matrices obtained with the use of the PDC for one patient, where it is possible to notice some horizontal patterns in the delta and theta band. This patterns occur due to the normalization adopted by the PDC, that is made with respect to the sinks.

### 5.3.4 Surrogate data test

After estimating the connectivity between EEG channels, a problem that may surge is to assess the significance of the connections found, that is, verify if the found coupling between the signals is indeed a result of the interactions between the channels. To assess the significance of the results, one may perform a hypothesis tests, under the null hypothesis of no causal coupling between a pair of signals on the specified direction.

Nevertheless, the PDC and the DTF have a nonlinear relation with the data they are derived, so the distribution of those estimators is not well defined, what cancels the possibility of using parametric hypothesis tests (which would assume a specific statistical distribution for the estimators). Hence, one way to perform this hypothesis test is by using a non-parametric hypothesis test, like the surrogate data test, that uses an empirical distribution of the estimators to perform the test [41].

To generate the surrogate data from the original data, the following procedure was adopted [42]:

- given the set of time series defined in equation (5.3), apply the Fourier Transform to each time series $x_n(t)$, obtaining the set of complex functions $X(f) = B(f)e^{j\phi(f)}$;
- next, the FT is phase-randomized, by adding a value $\Phi(f)$ drawn uniformly from the interval $[0, 2\pi[$ to each frequency $f$, obtaining $\tilde{X}(f) = B(f)e^{j(\phi(f)+\Phi(f))}$;
- finally, by applying the inverse FT on the phase-randomized transform, we can obtain a time series $\tilde{X}(t)$ with the same power spectrum as $X(t)$.

With the generated surrogate data, it is possible to generate an empirical distribution for the DTF and the PDC and use it to test the hypothesis of no causal coupling between the signals. When using this approach on this work, we considered a connection between two channels at a given frequency relevant when the estimator value was above the $95^{\text{th}}$ percentile of the empirical distribution obtained by means of surrogate data.

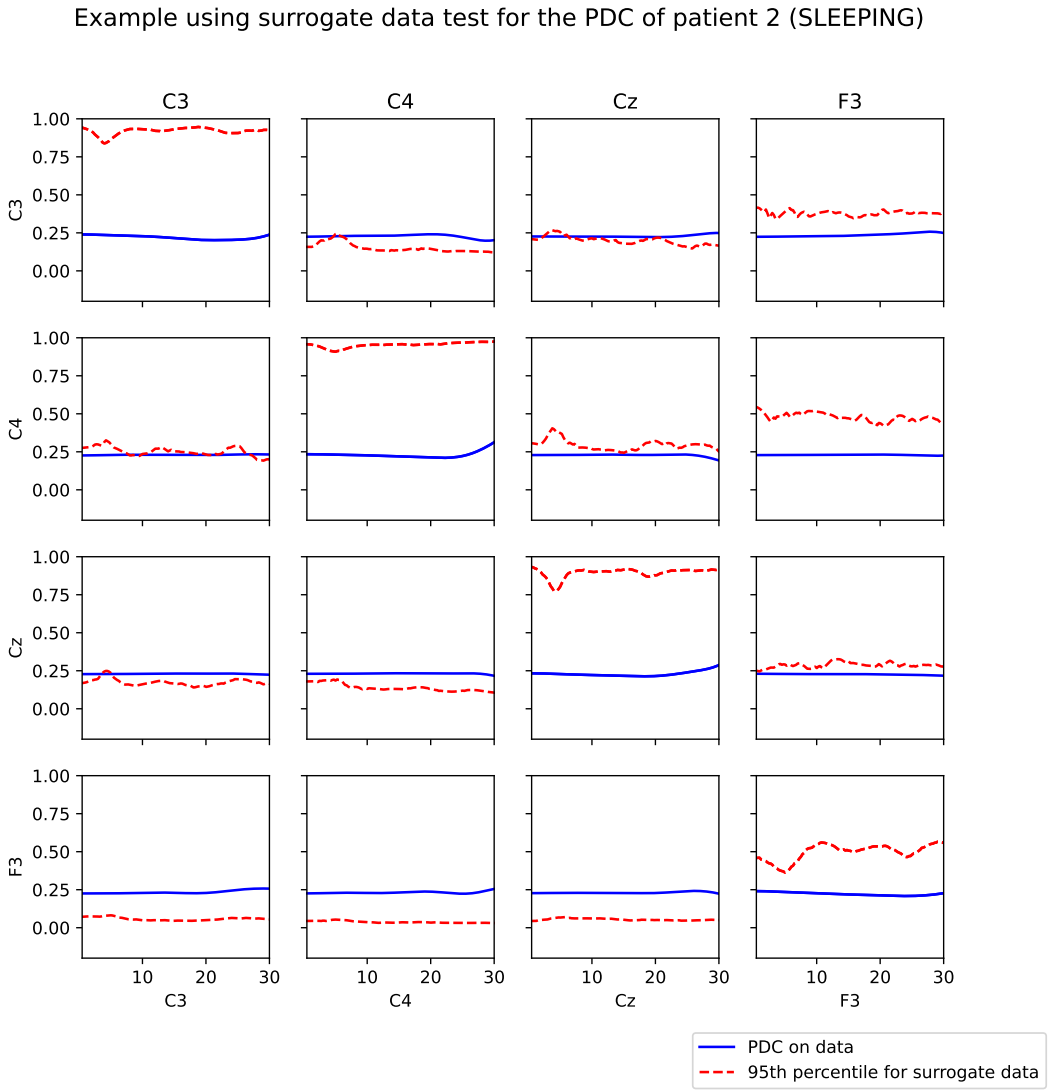


Figure 5.3.3: Example of connectivity obtained from EEG (solid blue line) and the 95$^{\text{th}}$ percentile of connectivity obtained from surrogate data (dashed red lines). The horizontal axes represent the frequencies and the vertical axes the connecitivity between the channels.

In the figure 5.3.3, a comparison between the coupling obtained from EEG data and by means of surrogate data is shown. In the regions where the connectivity is below the $95^{\text{th}}$ percentile of the connectivity obtained by means of surrogate data we set the connectivity to 0, because the surrogate data test is indicating that this connection is not significant.

## 5.4 Connectomes generation

After estimating the connectivity values with the PDC and DTF, we need to convert the obtained values into a graph, transforming the connectivity values between channels into a connectome. This conversion implies in two procedures, the representation of the measured connections at a given frequency band and the thresholding of them to create the edges of the connectomes.

The PDC and the DTF are represented as 3-D matrices with dimensions $channels \times channels \times frequency\,bins$, because the connectivity values are calculated on frequency bins (a discretization of the frequency domain). This implies that for each frequency bin we have a 2-D array representing the connectivity between the channels at this given frequency.

Consequently, in order to obtain the connectivity between two channels at a given frequency band, we opted to get the average connectivity measured on the frequency bins within this frequency band. Moreover, since we used 10 epochs for each patient, we also averaged the frequency bands connection matrices within the patients [40], obtaining one averaged connection matrix for each frequency band at a recording condition (awake or sleeping state EEG).

However, the matrices generated with this process generate dense networks, where the number of edges is equal the maximum possible number of edges ($19 \times 18$ edges disregarding the self-loops) and all edges have considerable weights, so the thresholding process may be an important step to retrieve important information from the connection matrices. In the literature, many methods were already described to perform that procedure [13], so we used three different setups and compared the results obtained with them. The three setups adopted were:

1. not using a threshold and generating the connectomes directly from the averaged matrices, creating weighted directed graphs, as shown in the figure 5.4.1;

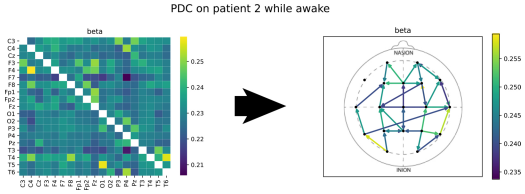


Figure 5.4.1: Representation of the steps to create a connectome without threshold. Just the connections above the 70$^{\text{th}}$ percentile are being show on the graph.

2. performing the surrogate data test (section 3.3.4) and setting the not relevant connections to 0 on each frequency bin before averaging the matrices, creating weighted directed graphs, as shown in the figure 5.4.2;

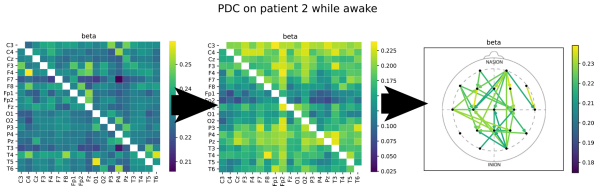


Figure 5.4.2: Representation of the steps to create a connectome using the surrogate data test. Just the connections above the 70$^{\text{th}}$ percentile are being show on the graph.

3. setting a fixed edge density of 50% for each connectome and setting the strongest connections to 1 and the weakest to 0, creating binary directed graphs, as shown in the figure 5.4.3.

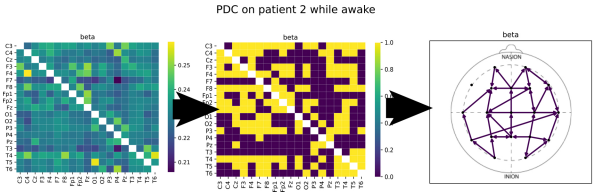


Figure 5.4.3: Representation of the steps to create a connectome using the binarization. Just the connections that were above the 70$^{\text{th}}$ percentile are being show on the graph.

## 5.5 Graph measures

From the connectomes (generated with PDC and DTF), it is possible to extract graph measures that will be used as features for the classification problem. For this work we opted for global measures, because we were more interested in changes in the whole network and not only in a specific region. For each connectome we calculated the global efficiency, global reaching centrality, modularity, average clustering coefficient and the degree assortativity.

### 5.5.1 Global efficiency

Global efficiency [43] is a measure of functional integration of the brain and is largely used in the study of brain functional connectivity. In the literature, the measure has already been used as a biomarker to identify patients with a good post-surgical outcome [44]. Also, VNS induced changes in global efficiency were proven to be a biomarker for VNS therapy outcome prediction [4].

The global efficiency $E$ is defined as:

$$E = \frac{1}{n} \sum_{i \in N} \frac{\sum_{j \in N, j \neq i} d_{ij}^{-1}}{n-1} \tag{5.15}$$

Where $N$ denotes the set of nodes, $n$ the amount of nodes, and $d_{ij}$ the distance to go from node $i$ to node $j$. To calculate the distance between two nodes, we used the length of the edges, that we defined as the inverse of the connectivity measures, because a higher connectivity would lead to a smaller distance and vice-versa [8].

The global efficiency is higher in networks with more and well distributed connections, for example, in the figure 5.5.1, the network on the left has a smaller global efficiency compared to the network on the right, because the average path length between two nodes is higher, leading to an inefficient network configuration.

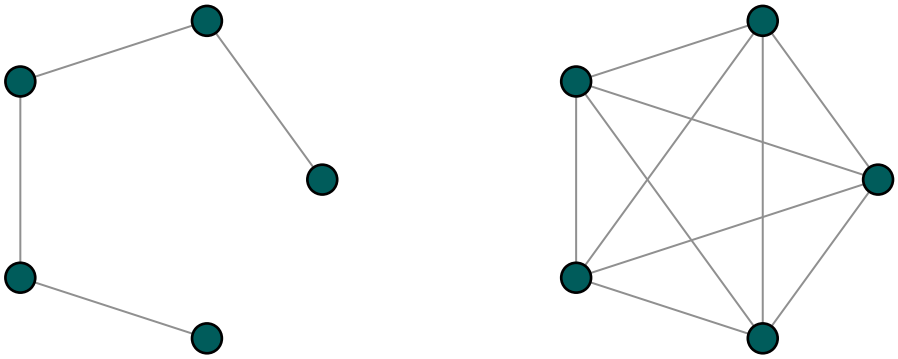

Figure 5.5.1: Comparison between the label and actual change of state

### 5.5.2 Average clustering coefficient

The clustering coefficient [45] is a local measure of functional segregation based on the number of connections between the neighbors of a node, so to obtain a global perception of the network we got this measure averaged across all the nodes. The choice of this measure is supported by its references in the literature, where studies on surface EEG have already shown changes in the clustering coefficient on interictal networks for patients with temporal lobe epilepsy [46]. Moreover, this measure is strictly related with the "small-world" phenomenon mentioned previously, because it quantifies the formation of clusters in a network.

An alternative to this measure would be the use of the transitivity as a measure of functional segregation, that has already been used on MEG-based networks to predict the efficacy of VNS treatment [21]. The main difference between the transitivity and the average clustering coefficient lies on the normalization adopted for each one of them. While the transitivity is normalized with respect to the whole network, the clustering coefficient is normalized for each node individually, so the transitivity is less influenced by nodes with few neighbors. However, due to the references to the use of averaged clustering coefficients on networks generated on surface EEG and to the fact that some of the networks used on this work are densely connected, we opted to use the average clustering coefficient instead of the transitivity.

The clustering coefficient $C_i$ for a node $i$ in a weighted directed networks is defined as [47]:

$$C_i = \frac{[W^{\frac{1}{3}} + (W^T)^{\frac{1}{3}}]^3_{ii}}{2[d_i^{tot}(d_i^{tot} - 1) - 2d_i^{\leftrightarrow}]} \tag{5.16}$$

Where $W$ is a matrix with each element $w_{ij}$ representing the weight of the edge pointing from $i$ to $j$, $W^{\frac{1}{3}}$ is the cube root performed on each element of $W$, $d_i^{tot}$ is the total degree (sum of the in-degree and the out-degree) of node $i$, and $d_{ij}^{\leftrightarrow}$ is the bilateral degree between nodes $i$ and $j$. For a weighted graph, $d_i^{tot}$ and $d_{ij}^{\leftrightarrow}$ are defined as:

$$d_i^{tot} = \sum_{j \in N: j \neq i} (w_{ij} + w_{ji}) \quad \text{and} \quad d_{ij}^{\leftrightarrow} = \sum_{j \neq i} w_{ij} w_{ji} \tag{5.17}$$

The clustering coefficient is bigger for nodes that are inside a densely connected region, where the neighbors are densely connected between themselves. In the figure 5.5.2, it is possible to notice three subgraphs with different average clustering coefficients.

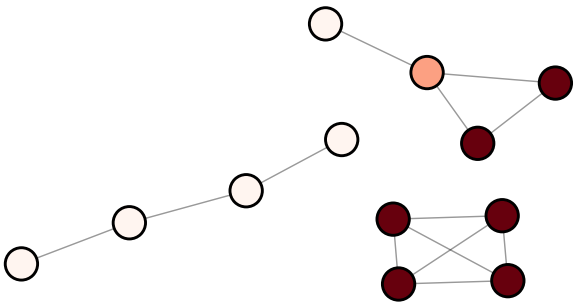

Figure 5.5.2: Subgraphs showing the clustering coefficients for different nodes. Red nodes have higher clustering coefficient and white ones have lower values.

### 5.5.3 Modularity

The modularity [48, 49] is a measure of functional segregation that measures how well a predetermined community configuration is meaningful for a network. In the literature, this measure proved to be an important biomarker to classify responders to treatment with canabidiol [50] and to VNS therapy [21].

The formulation of the modularity $Q$ for a weighted directed networks is given by:

$$Q = \frac{1}{m} \sum_{i,j \in N} \left[ w_{ij} - \frac{d_i^{in} d_j^{out}}{m} \right] \delta_{c_i,c_j} \tag{5.18}$$

Where $m$ is the total number of edges in the network, $w_{ij}$ is the weight of the edge linking $i$ to $j$, and $\delta_{c_i,c_j}$ is a function that is equal to 1 when the nodes $i$ and $j$ belong to the same community and 0 otherwise. Moreover, $d_i^{in}$ and $d_i^{out}$ represent the in-degree and the out-degree of a node $i$ respectively, defined for a directed weighted network as:

$$d_i^{in} = \sum_{j \in N : j \neq i} (w_{ij}) \quad \text{and} \quad d_i^{out} = \sum_{j \in N : j \neq i} (w_{ji}) \tag{5.19}$$

In order to calculate the modularity of the connectomes, we defined a community division, as shown in the right part of the figure 5.5.3. Since we used low-density EEG to build the connectomes, the community division proposed is different than the most common divisions found in the literature, because some regions were merged in a way that each community had at least three nodes. Starting from the division proposed in the left part of the figure 5.5.3, we merged groups of electrodes and made some permutations to achieve the communities division used in the calculations.

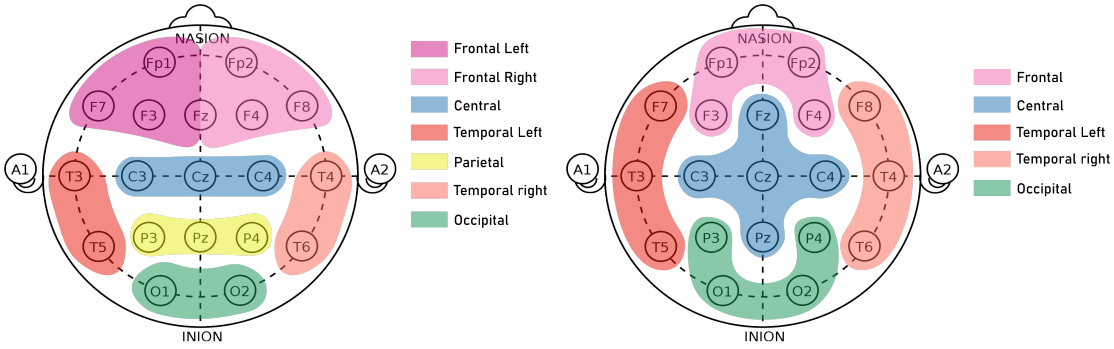


Figure 5.5.3: On the left the distribution of EEG electrodes and brain regions adapted from [51]. On the right the communities configuration used on the modularity calculation.

The modularity is bigger for networks where the connections between nodes from different communities are weaker. For example, in the figure 5.5.4 the first

network has a higher modularity, because the biggest part of the connection are set between nodes from the same community. On the other hand, in the second network, a lot of connections exist between the communities, what reduces the modularity.

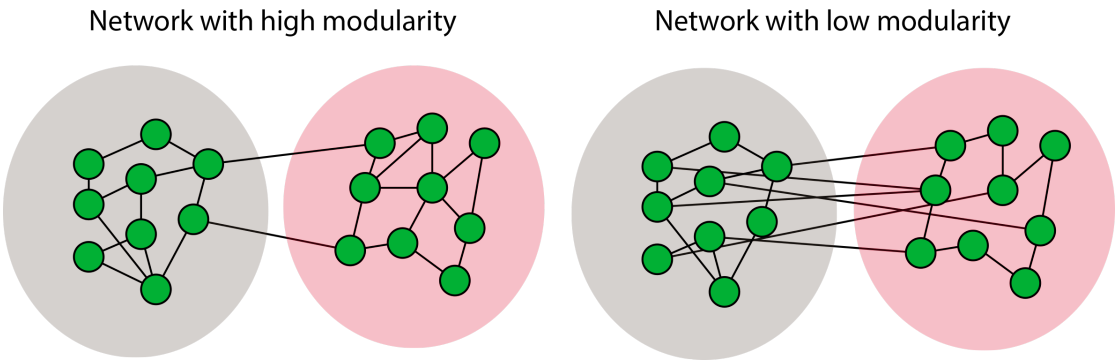


Figure 5.5.4: Two networks with different modularity. The first network has higher modularity because the biggest part of the connections occur within the communities (defined by the colored zones).

### 5.5.4 Global reaching centrality

The global reaching centrality [52] is a measure of the centrality of a network, an important type of measure to identify the existence of hubs in networks. In the literature, it has already been shown that a decreased hub-value in interictal states may be linked to the localization of the epileptic zone (EZ), suggesting a functional isolation of the EZ in the intericatal state [53].

Although the major findings about hubs in interictal networks are not directly related to the objective of this work, we decided to explore those types of measures, as the presence or absence of hubs may be associated with the response to VNS therapy.

Before defining the global reaching centrality, it is necessary to define the local reaching centrality for a node $i$, given by:

$$RC_i = \frac{1}{n-1} \sum_{j:0<l^{out}(i,j)<\infty} \left( \frac{\sum_{k=1}^{l^{out}(i,j)} w_{ij}^{(k)}}{l^{out}(i,j)} \right) \tag{5.20}$$

Where $l^{out}(i,j)$ denotes the number of edges in the directed path going from node $i$ to node $j$, $w_{ij}^{(k)}$ represents the weight of the $k$-th edge along this path, and

$n$ represents the total number of nodes in the network. If two nodes $i$ and $j$ are connected by more than one directed path, then the one with the maximum weight should be considered.

From the local reaching centrality, it is possible to define the global reaching centrality as [52]:

$$GRC = \frac{\sum_{i \in N}[RC^{max} - RC_i]}{N - 1} \tag{5.21}$$

Where $RC^{max}$ is the maximum local reaching centrality in the network and $RC_i$ is the reaching centrality of node $i$.

For example, in the figure 5.5.5, the network on the left has a smaller global reaching centrality when compared to the network on the right, because the first network does not have any central node, while the second one clearly has a node working as a hub.

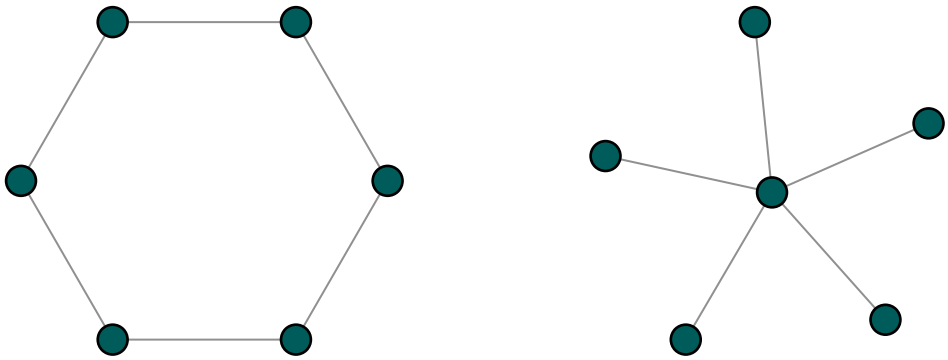

Figure 5.5.5: Example of two graphs with different global reaching centrality values.

### 5.5.5 Degree assortativity

The degree assortativity [54] is a measure of the resilience of a network, that measures the similarities of the connections on the graph with respect to the degree of the nodes that are being linked. On the context of brain functional connectivity, the study of epileptic seizures showed an increase of the degree assortativity during the evolution of a seizure, so this measure can be used to quantify the ictal activity of the brain [55].

Even though this measure is commonly associated to the study of ictal networks, we decided to explore the relation between this measure and the VNS therapy outcome in interictal networks, because some ictal activity might still be present in those networks.

Let $\alpha, \beta \in \{in, out\}$ be the degree type being used, and $j_i^\alpha$ and $k_i^\beta$ be the $\alpha$-degree and $\beta$-degree of the source and target node of the edge $i$. Then, the degree assortativity for a weighted directed network [56] is given by:

$$r(\alpha, \beta) = \frac{E^{-1} \sum_{i \in E} \left[ (j_i^\alpha - \bar{j^\alpha})(k_i^\beta - \bar{k^\beta}) \right]}{\sigma^\alpha \sigma^\beta} \tag{5.22}$$

with:

$$\bar{j^\alpha} = E^{-1} \sum_i j_i^\alpha, \ \bar{k^\beta} = E^{-1} \sum_i k_i^\beta,$$

$$\sigma^\alpha = \sqrt{E^{-1} \sum_i (j_i^\alpha - \bar{j^\alpha})^2} \text{ and } \sigma^\beta = \sqrt{E^{-1} \sum_i (k_i^\beta - \bar{k^\beta})^2}$$

where $E$ is the number of edges in the network.

To calculate the assortativity on this work we used $\alpha = out$ and $\beta = in$, so we used the out-degree of the source nodes and the in-degree of the target nodes to compute the measure. To use the weights of the edges we defined the in-degree of a node as the sum of the weights of the edges pointing to this node and the out-degree in a similar way, but considering the edges leaving the node.

In the figure 5.5.6, it is possible to notice two networks with different degree assortativity. In the network on the left the nodes with similar degree are linked together, which indicates a network with high degree assortativity (assortative network). In the network shown in the right, the nodes are linked with others that have different degree, which characterizes a network with low degree assortativity (disassortative network).

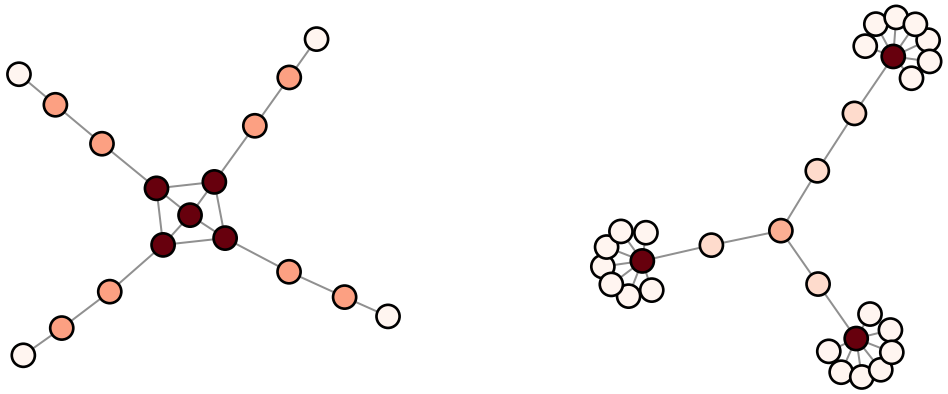

Figure 5.5.6: Example of two graphs with different assortativity. Nodes in darker red have higher degree and nodes in lighter red have lower degree.

## 5.6 Feature extraction pipeline overview

After defining all the processes used in the feature extraction process it is possible to have a general view of how the pipeline works. In the figure 5.6.1 it is possible to notice the steps of the pipeline.

In the first stage the data arrives in the format of `.edf` files, containing the raw records of the EEGs, and is transformed into MNE-Python raw EEG data structure. It is important to highlight that for each patient two EEGs are provided, one recorded during sleep, and another one recorded with the patient awake.

In the second stage, the EEG channels are re-referenced to a common average, filtered, and split into epochs. The division into epochs is made with the annotations set by neurologists and this kind of data is managed with the epoched data structure from MNE-Python.

In the third stage the connectivity matrices are created for each EEG. To obtain the wPLI we used the spectral connectivity function implemented on MNE-Python, that is fully compatible with the epoched EEGs. The calculation of the connection matrices for PDC and DTF are made with SCoT-Python, but the scripts were adapted to be compatible with MNE-Python data structures and to

return the connectivity on the sensor space (the default for SCoT is the sensor space connectivity). Also, it is on this step where the surrogate data test is performed when it is used as thresholding method.

In the fourth stage the connection matrices calculated with PDC and DTF are transformed into graphs and the average synchronization measure is calculated for the wPLI matrices. To build the graphs and extract the graph measures, we used Networkx-Python with the non-thresholded or thresholded matrices.

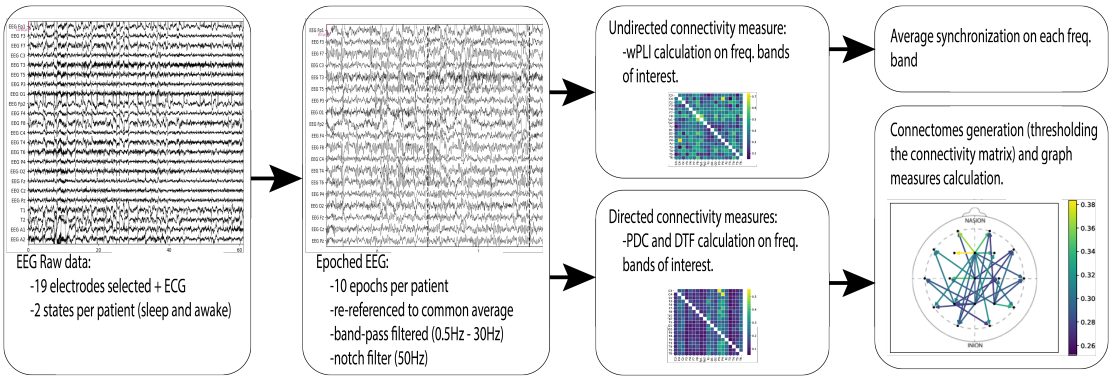


Figure 5.6.1: Scheme representing the pipeline used for feature extraction

At the end of the pipeline we obtain 110 features for each patient, as exemplified by (5.23), where we have $n_{rec.\,status} = 2$ (sleep and wake), $n_{dir.\,conn.} = 2$ (PDC and DTF), $n_{freq.\,bands} = 5$ (delta, theta, alpha, beta and broadband), and $n_{wPLI\,matrices} = 5$ (one averaged matrix for each frequency band).

$$n_{feat.} = n_{rec.\,status}\left[n_{dir.\,conn.} \times n_{graph\,measures} \times n_{freq.\,bands} + n_{wPLI\,matrices}\right] \quad (5.23)$$

## 5.7 Machine learning models

After extracting all the features, we trained machine learning models for the binary classification task, that consists in classifying a patient in responder or non-responder to VNS therapy using the 110 features extracted. To do that we selected the best features for each model, tuned their hyperparameters and tested them in a test set.

### 5.7.1 Train-test split and validation procedure

The first step adopted on the machine learning models training was the split of the data into a training set and a test set. The training set is used for training

the models and tuning theirs hyperparameters, with the purpose of getting the configurations that lead to the best validation score on that dataset. After tuning the models, it is necessary to assess their performance in a situation similar to a real application, that is, data that was not used in the model training, so we use the test set to measure the model accuracy.

The first point to be decided while splitting the data is which percentage of data should be used for the test of the models, however, there is no agreement on the literature about what is the optimal split, since it should be adapted for each problem individually. While spliting the data, one should keep in mind the trade-offs of each split configuration, for example, a bigger test set can give a better overview of the true performance of the model, but can drastically reduce the training dataset and impair the model performance.

The dataset used to train the models in this work counts only with 37 patients, which might be a strong limitation for some machine learning models. Then, in order to keep a reasonable amount of data for the training of the models, we decided to select only 7 patients to form the test set, leaving the other 30 patients for the training set. This split favors the training and validation of the models, but reduces the confidence of the models test performance estimation. The final data split adopted is represented in the figure 5.7.1.

For the validation of the models we used a 6-Fold cross-validation, which consists in splitting the data into 6 equal parts (folds) and at each time using one of the parts to validate the model performance and the other ones to train it. The final validation accuracy is the average accuracy obtained for each validation fold. The cross-validation technique was used in the adjustment of hyperparameters and in the sequential feature selection.

One problem that may occur during or after this split is the data leakage, when information is shared between the training and test set and make the final assessment of the performance biased. Then, to avoid that problem we performed the split of the dataset at the very beginning of the models training.

### 5.7.2 Feature selection

Another difficulty particular from our application is the great amount of features present in our dataset, that is way bigger than the amount of samples we have (110 features against 30 samples for training). This lack of balance between the number of features and the number of samples can harm the performance of the model, because in machine learning applications with many features a bigger amount of data is needed to ensure that all the values combination are present in the dataset.

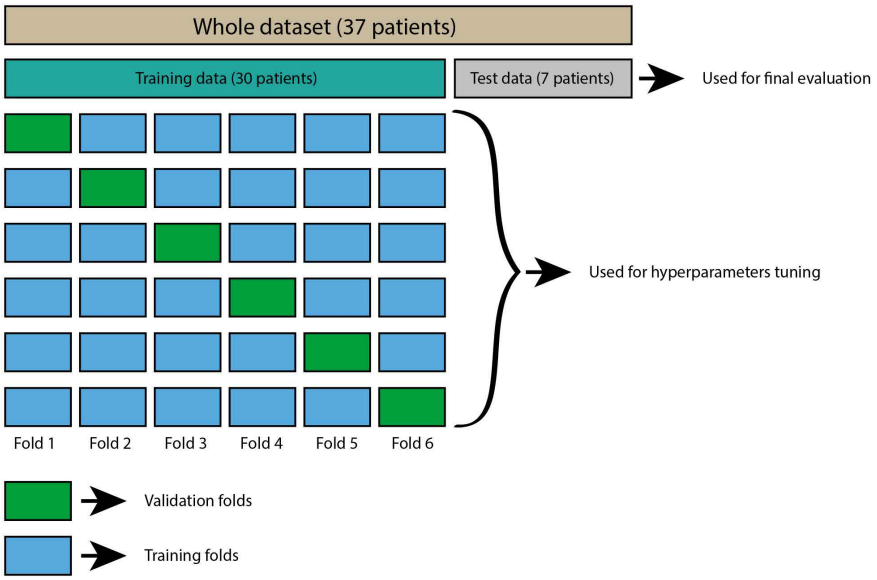


Figure 5.7.1: The data splitting used to train and assess the machine learning models

In short, more samples are needed to guarantee the generalization of the model when more features are used.

In addition, the predictive power of a model starts to decrease when inserting additional features to the optimal subset, this phenomenon is widely know as the "Peak Phenomenon" or the "Curse of dimensionality". This phenomenon states that the inclusion of features that are redundant or not important into the optimal subset of features can reduce the model's performance for a given task [57].

Hence, we took advantage of feature selection methods to reduce the original set of features to a smaller one that presents a better accuracy in our classification task. To do so, we used two filter methods, the Maximum relevance and Minimum redundancy (MrMr) and the correlation based feature selection, and one wrapper method, the sequential feature selection. There are several differences between filter and wrapper methods, with each showing an advantage on different situations, so we used both kind of methods to compare the results we could obtain with them.

The filter methods select a subset of features based on a score function or some statistic, so the features selected by those methods are independent of any machine learning model. In order to achieve that, the score functions adopted in those methods usually try to reduce the redundancy of the features in the subset while maximizing the relation between the subset and the target variable. The main

advantage of those methods is their computational efficiency, so they are largely used in large datasets, where greedy algorithms would take too much time to select a good subset of features. Also, another possibility is the use of filter methods as a preprocessing step for some other feature selection algorithm.

On the other hand, wrapper methods are model specific, that is, the subset of features found by that methods are different for each model. Usually those methods work by starting with a set of features and removing or adding the features that increase the validation accuracy. The biggest drawback of those methods is their computational complexity, since to find a subset those methods need to perform many fits with the model, which might be a demanding task depending on the size of the dataset and the complexity of the model used.

The first filter method we chose for the task of feature selection is the correlation based feature selection [58], that tries to find a subset that maximizes the score function shown in equation 5.24. In that equation, $k$ represents the number of features in the subset, $\bar{r_{cf}}$ the average feature-class correlation, and $\bar{r_{ff}}$ the average feature-feature correlation.

$$Merit_s = \frac{k\bar{r_{cf}}}{\sqrt{k + k(k-1)\bar{r_{ff}}}} \tag{5.24}$$

In this work the features are continuous, so we calculated the correlation between them using the Pearson correlation coefficient. For the features-target correlation it is not possible to use the Pearson correlation, because the target variable is binary, so to compute the correlation between the features and the target we used the Point-biserial correlation.

To find the best subset, the algorithm starts with an empty subset and calculate for each feature the score of being added to the subset (that is empty at that stage) and add the feature that leads to the biggest merit. This implies that the first feature added to the subset is the one with biggest correlation with the target variable, a result of using $k = 1$ on equation 5.24.

The process continues in an iterative way, adding at each iteration a feature that leads to merit increase. Whenever a subset cannot be improved anymore, the algorithm performs a backtrack to the best non-expanded subset and starts the search again adding a different feature. A search performed without backtracking limitation would lead to a search in all the subsets space, testing the merit of all possible subsets. Then, to simplify the process we limited the number of backtracks to 5, so the process ends when the backtracks limit is reached and returns the

subset with the biggest score among the explored possibilities [59].

The other filter method used in this work is the Maximum relevance and Minimum redundancy (MrMr) method [60]. This method also tries to find a features subset that maximizes a score function, but now the score for a feature $f$ at the $i$-th iteration is given by equation 5.25 [61]. The function $F(f, target)$ is a function that returns the F-score between the feature $f$ and the target variable and $corr(f, s)$ is the Pearson correlation between the features $s$ and $f$.

$$score_i(f) = \frac{F(f, target)}{\sum_{s \in features\ selected\ until\ i-1} |corr(f, s)|/(i-1)} \tag{5.25}$$

Similarly to the correlation based feature selection, this method starts with an empty subset of features and at each iteration add to the subset of selected features the one that has the biggest computed score. However, the stopping criterion is different for this method, because the user needs to define a number of features $K$ to be selected, then the algorithm stops when the subset reaches the predefined size. Instead of using a fixed $K$ for all the models, we decided to explore for each model the $K$ ranging from 1 to 20 that led to the best performance [61].

The third method used for feature selection was the sequential feature selection, which is a wrapper method and should be applied for every model. This method works in a simple way selecting at each time a feature that increases the cross-validation score of the model. Then, for each model we start with an empty subset of features and add the feature that lonely lead to the best cross-validation accuracy. Hence, in the next iterations all the possible features are tested inside the subset and the one that gives the best cross-validation accuracy is added to the subset. This process is repeated until the cross-validation performance cannot be increased with the addition of a new feature.

### 5.7.3 Data re-scaling, normalization and standardization

The features we used to train the machine learning models are contained in the most variable ranges, what might be a problem for some models, so we performed transformations on the data to make them a best fit for the models. In short, we used three data transformation techniques for our data and tested which one of them worked better for a given model.

One of the transformations used is the normalization of the data, which consists in normalizing each sample by its magnitude. This method considers each sample as an array and works by dividing each component of the array by the norm of the

array, transforming each sample into an unit vector, as exemplified by equation 5.26.

$$[x_1{}', x_2{}'...x_n{}'] = \frac{[x_1, x_2...x_n]}{x_1^2 + x_2^2 + ... + x_n^2} \tag{5.26}$$

This method might be useful when the objective of the data transformation is to re-scale the data preserving the cosine similarity between the data points, because we are preserving the angle of the sample arrays in the space formed by the features.

The second transformation adopted was the standardization of the data, which standardize each feature by subtracting its mean and scaling to unit variance. The standardization of a feature $f$ in a sample $i$ is given by the equation 5.27, where $\mu_f$ is the mean and $\sigma_f$ the standard deviation of the feature $f$.

$$x_{fi}{}' = \frac{x_{fi} - \mu_f}{\sigma_f} \tag{5.27}$$

This method is particularly useful for machine learning methods that assume the data has zero mean and unit variance (normally distributed around 0), such as support vector machines.

The last transformation tried was the re-scaling of the data, which consists in considering the minimum and the maximum value of a feature to make it lie within the range $[0, 1]$. The re-scaling is made according to the equation 5.28, where $f_{min}$ represents the minimum and $f_{max}$ the maximum values of the feature.

$$x_{fi}{}' = \frac{x_{fi} - f_{min}}{f_{max} - f_{min}} \tag{5.28}$$

This method is commonly used as an alternative to the standardization in cases where it does not perform well, for example when the standard deviation is too small or when the preservation of zero entries in the data is important.

Given that, we decided to try the three transformations on each model where it could affect the result. It is important to highlight that the decision tree based models (decision tree, random forest and AdaBoost with decision tree) are invariant to those transformations, so the data used in these models was not transformed.

### 5.7.4 Models used

To find the best model for our classification problem, we tested many models for the task, to do that we selected the best features using the three methods mentioned before and tuned the hyperparameters that are specific for each model, however, some models include particularities and those were taken into account while performing the tests. We performed the test with 8 different models: support vector machine with linear kernel, support vector machine with polynomial kernel, support vector machine with radial basis function kernel, Gaussian process, k-nearest neighbors, decision tree, random forest, and AdaBoost.

#### Support vector machine: linear kernel

Support vector machine (SVM) is a machine learning model originally formulated for binary classification problems. Assuming data points with $n$ features and divided into two classes, the classical support vector machine formulation consists in finding an hyperplane capable of splitting the data into two in the $n$-dimensional space according to the class attributed to each data point [62].

The hyperplane defined during the fit of the SVM is a maximum margin separator, that is, it lies within the biggest possible distance between the data points. However, for generalization purposes or data conditioning, one may want to allow data points to violate that boundary, so the original loss function can be modified to include a regularization hyperparameter that "softens" the margin, allowing points to lie within the margin [62].

Consequently, the hyperparameters that we tuned for this model were the margin regularization and the tolerance for the stopping criterion.

#### Support vector machine: polynomial kernel

One difficulty that the SVM with linear kernel may face is the non-linearity of the data, in a way that the classes cannot be divided by the hyperplane, so it is necessary to appeal to modifications in the original formulation of the SVM to increase its performance in different data. One modification of the original formulation is the use of kernels to transform the data to a feature space where it should be linearly separable [62].

The first non-linear kernel we tried in the SVM model is the polynomial kernel and for this model we tuned the degree of the polynomial kernel, the scale of it, the independent term of the polynomial, and the hyperparameters that were also tuned for the linear SVM (margin regularization and the tolerance for the stopping criterion).

### Support vector machine: radial basis function kernel

Another possible and widely used non-linear kernel for the support vector machines is the radial basis function. This kernel just have one parameter to be tuned, the scaling of the kernel function, so we tuned this hyperparameter and the ones that were also tuned for the linear SVM model.

### Gaussian process

A Gaussian process specified by its mean function and covariance function, that is, the Gaussian process can be seen as a generalization of the Gaussian distribution, that has the mean defined as a vector and the covariance as a matrix. Then, given an input, this model should return a probability distribution for the possible output values [63].

One characteristic of this method is the way the covariance matrix may be defined, where a kernel can be used to provide information about the prior distribution of the data on the space. Hence, in this work we tried to find the best kernel for the Gaussian process models, trying linear combinations of different kernels or the product between them [63]. Namely, we tested the linear combination of the kernels dot product, white, and radial basis function. We opted for these kernels because they were already implemented in the Sci-kit Learn library.

### K-Nearest neighbors

The K-nearest neighbors model is one of the simplest machine learning models, because it works by attributing one data point to the class of its $k$ nearest neighbors. In other words, the training data is positioned in the $n$-dimensional space and when a point with unknown class is positioned into that space it is attributed to the same class as its $k$ closest neighbors.

The two main parameters to be tuned in that model are the number of neighbors used in the class attribution, that is, the number $k$ of points used in the classification, and the distance metric used to compute the distance between the data points.

### Decision tree

The decision tree is a model based on simple decision rules that can be inferred from the dataset and its features. The final shape of a decision tree is a tree-shaped workflow, where at each node the value of one feature is used to take a decision, until the end node is reached and a label is attributed to the data. To establish this decision flow, the algorithm builds a tree recursively, using the features that

provide the biggest information gain, that is, have greater capacity to separate the data points between the classes.

The greatest disadvantage of decision trees is the overfitting of the model to the training data, because the tree building algorithms can build over-complex trees that lacks generalization, working well just with the training data. To avoid the overfitting we tuned a max depth parameter for the models, which limits the depth the trees are able to grow and consequently avoids the creation of over-complex models.

An important aspect of decision trees is that those models perform a feature selection by themselves and are invariant to data scaling, so we used the raw data into those models, without any kind of transformation or feature selection method.

### Random forest

The random forest model is based on the concept of meta estimators, which consists in a set of weak learning algorithms working together. The random forest model is composed by many shallow decision trees, working as weak learners, built from random samples drawn from the training dataset. Because of that, in a classification problem each decision tree will have a different class prediction for the same data point, so the final prediction of the random forest will be the class that occurred the most among this set of predictions.

For this model we tuned the maximum depth parameter of the trees and the number of trees that composed the random forest.

### AdaBoost

The AdaBoost is a model that uses a set of weak learners to predict the final label of the data, but uses a method called boosting iteration to improve the performance of the model in particular difficult data points. The model starts by fitting one weak learner on the whole training data and then modifying the weight attributed to the samples that were misclassified by this weak-learner, then iteratively new weak learners are fit to the data and the weights are modified, until the maximum number of weak learners is reached. By doing this, each subsequently weak learner is forced to perform better in the samples that were particularly difficult for the previous learners. Finally, after all the weak learners are fit, the model can predict the label of a data point similarly to the random forest, by taking the majority of the labels predicted by the weak learners [64].

For this model we used shallow decision trees as weak learners and we tuned the

maximum depth of those trees and the number of estimators used by the AdaBoost model.

# 6 Results

After running the pipeline, collecting the features and training the machine learning models, we evaluated the results obtained, which we will show in this chapter. It is important to notice that in this chapter we mention three datasets, each one generated with one of the thresholding setups specified in section 5.4. Hence, we have one dataset of features that were extracted from connectomes that were not thresholded, one dataset of features extracted from connectomes thresholded with the surrogate data test, and another one composed by features extracted from binarized connectomes with an edges density of 50%. In addition, the synchronization measure was not thresholded, so the 10 features extracted from this measure are equal in the three datasets.

## 6.1 Most important features

To find possible biomarkers to the response to VNS therapy, we looked for differences between the populations of responders and non-responders for all the features. The first step to perform this analysis consisted in testing the normality of the features for both populations, because it is important to define which hypothesis tests we can use to compare the populations. Indeed, the features were not normal, and since we have a small amount of data, the use of parametric tests, like the t-student test, may lead to erroneous results.

Then, one alternative to the parametric tests are the non-parametric statistical tests, that does not make assumptions about the data distribution. Therefore, to study the difference between the populations we used the Mann-Whitney U test, with the null hypothesis that the populations are equal and the alternative hypothesis that the populations are not equal.

However, each hypothesis test is performed for features defined at a frequency band and the test of features from the same family over all the frequency bands

consists in a multiple comparison problem, where we have a set of 5 statistical inferences happening at the same time. The tests involving multiple comparisons can increase the probability of type I errors (rejection of a true null hypothesis), so we need to correct the p-values defining a false discovery rate. Hence, after doing the Mann-Whitney U test for all the variables, we corrected the p-values for each family of features using the Benjamini-Hochberg procedure [65] using a false discovery rate of 5%.

After performing the procedures described above in the three datasets, we found five features that can work as possible biomarkers to predict the outcome of VNS therapy. In the dataset of the features generated from the connectomes thresholded with the surrogate data test, we found significant differences between the populations in the global efficiency (beta band and in the broadband) and in the average clustering coefficient (delta band and in the broadband) for the networks generated with the PDC in sleeping state. In the dataset generated from binarized connectomes, the modularity (alpha band) in the networks generated with PDC in awake recordings showed some power to discriminate responders from non-responders. The corrected p-values for the aforementioned features can be found in the table 6.1.1 and the corrected p-values for all the features are in the appendix, in the tables B.0.1, B.0.2, and B.0.3.

| **Feature** | **Thresholding method** | **p-value** |
|---|---|---|
| Avg. CC on delta band while sleeping (PDC) | surrogate data test | 0.024 |
| Mod. on alpha band while awake (PDC) | 50% density binarization | 0.029 |
| Avg. CC on broadband while sleeping (PDC) | surrogate data test | 0.029 |
| GE on beta band while sleeping (PDC) | surrogate data test | 0.044 |
| GE on broadband while sleeping (PDC) | surrogate data test | 0.049 |

Table 6.1.1: P-values of the relevant features found by the Mann-Whitney U test.

After performing the hypothesis test for all the features, we decided to explore the discriminating power of the most relevant ones. To do that we used the receiver operating characteristic (ROC) curve to find the best threshold for the features on the classification task, then we explored the sensitivity, the specificity and the overall accuracy of those features.

The ROC curve shows how the performance of a feature varies according to

the variation of its discrimination threshold, that is, for different values of the feature, the curve shows the division obtained for the dataset in responders and non-responders. The best threshold in the classification problem is found as the farthest point on the diagonal that crosses the graph. Furthermore, one important measure derived from that curve is the area under the curve (AUC), that measures the overall performance of the feature. A big AUC indicates good discriminating power for the feature, while a smaller value indicate the opposite.

With the best found threshold we assessed the sensitivity, specificity and overall accuracy of the features. The sensitivity (true positive rate) measures the capacity of the features in identifying responders, the specificity (true negative rate) measures the capacity of the features in identifying non-responders, and the accuracy measures the percentage of correct classifications the features can perform (considering positive and negative cases).

### 6.1.1 Use of wPLI as synchronization measure

The synchronization already showed great power in predicting the outcome of VNS therapy in previous studies. Also, this kind of measure showed to be very important in the explanation of the action of VNS in the brain [4, 3, 33]. However, the features based on the wPLI did not show significant differences between responders and non-responders in this work, as shown by the boxplots in the figure 6.1.1 and the table 6.1.2.

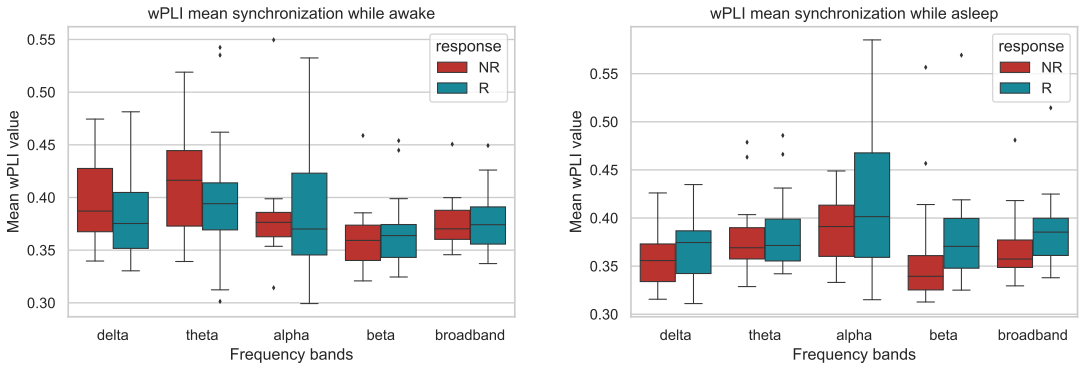


Figure 6.1.1: Boxplots for the wPLI obtained from EEGs in awake state (left plot) and in sleeping state (right plot).

| | P-values for the mean wPLI in different freq. bands | | | | |
|---|---|---|---|---|---|
| | **Delta** | **Theta** | **Alpha** | **Beta** | **Broadband** |
| **Awake** | 0.47 | 0.47 | 0.49 | 0.49 | 0.49 |
| **Sleep** | 0.23 | 0.38 | 0.27 | 0.06 | 0.06 |

Table 6.1.2: P-values obtained for the wPLI based features.

Despite the lack of significant differences between responders and non-reponders in the hypothesis test, the mean wPLI showed discrimination power in the beta band of sleeping state EEGs. In the figure 6.1.2 it is possible to see the ROC curve for this feature and the confusion matrix obtained with the best threshold.

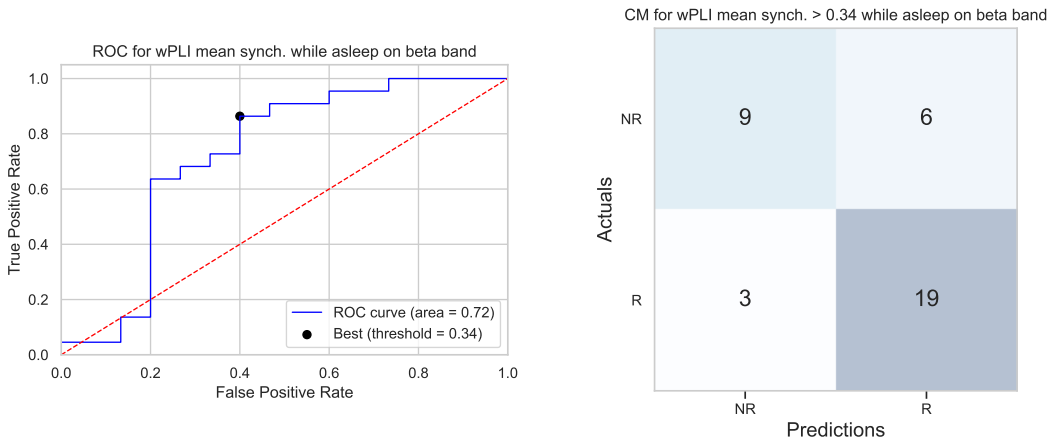


Figure 6.1.2: The ROC for the mean wPLI on beta band and the confusion matrix for the best threshold obtained.

By setting as responders the patients with a mean wPLI above 0.34 we obtained the classification performance shown in the table 6.1.3.

| GE on beta band while sleeping (PDC) | | |
|---|---|---|
| **Sensitivity** | **Specificity** | **Accuracy** |
| 0.86 [0.65, 0.97] | 0.60 [0.32, 0.84] | 0.76 [0.59, 0.88] |

Table 6.1.3: Performance of the mean wPLI on beta band for the specified threshold.

Even with not significant difference found by the hypothesis test, the mean wPLI on beta band on sleeping state EEGs showed a great discrimination power and might be an important feature for the machine learning models.

### 6.1.2 Avg. CC on PDC connectome while sleeping

According to the p-values presented before, the average clustering coefficient on the surrogate data test dataset showed some capability in differentiating responders and non-responders to VNS therapy in two frequency bands, the delta band and the broadband. The figure 6.1.3 shows the average clustering coefficient on sleeping state obtained in all the frequency bands for the connectomes generated with PDC. From the boxplots it is possible to notice the difference on the average clustering coefficient between responders and non-responders on the delta band and on the broadband.

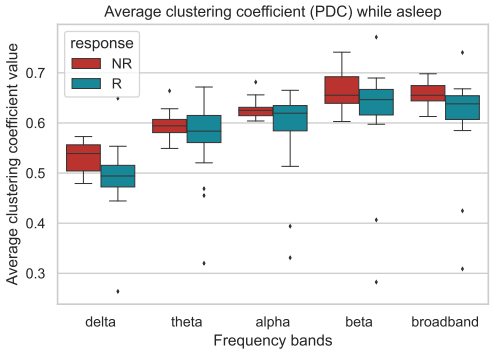


Figure 6.1.3: Boxplot for the avg. CC on PDC connectomes while sleeping

The ROC curve obtained for this feature in the delta band is shown in the figure 6.1.4, with the confusion matrix obtained with the best threshold according to the curve.

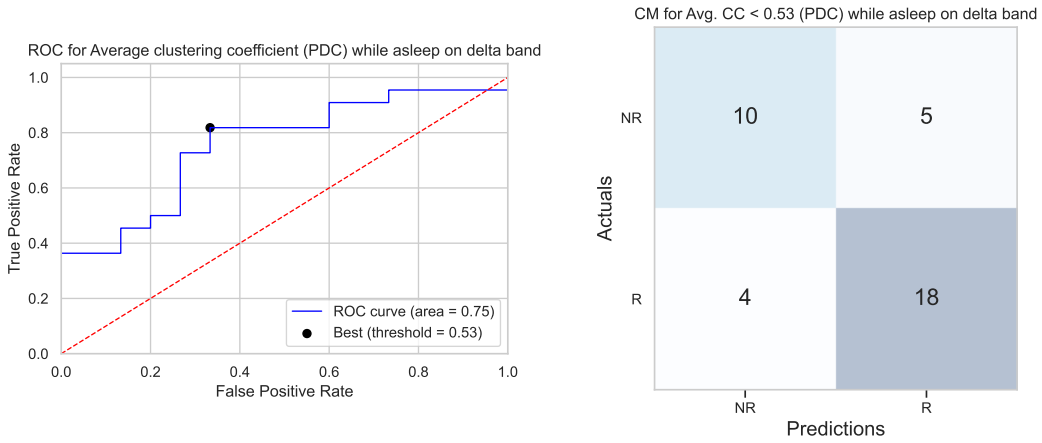


Figure 6.1.4: The ROC for the avg. CC on delta band and the confusion matrix for the best threshold obtained.

From this feature we got that the best threshold setup for the classification is to set as responders the patients with an average clustering coefficient below 0.53. The performance of this threshold is summarized in the table 6.1.4.

| **Avg. CC on delta band while sleeping (PDC)** | | |
|---|---|---|
| **Sensitivity** | **Specificity** | **Accuracy** |
| 0.82 [0.60, 0.67] | 0.67 [0.38, 0.88] | 0.76 [0.59, 0.88] |

Table 6.1.4: Performance of the avg. CC on delta band for the specified threshold.

Performing this same analysis with the avg. CC on the broadband, we obtained the ROC curve and the confusion matrix shown in the figure 6.1.5.

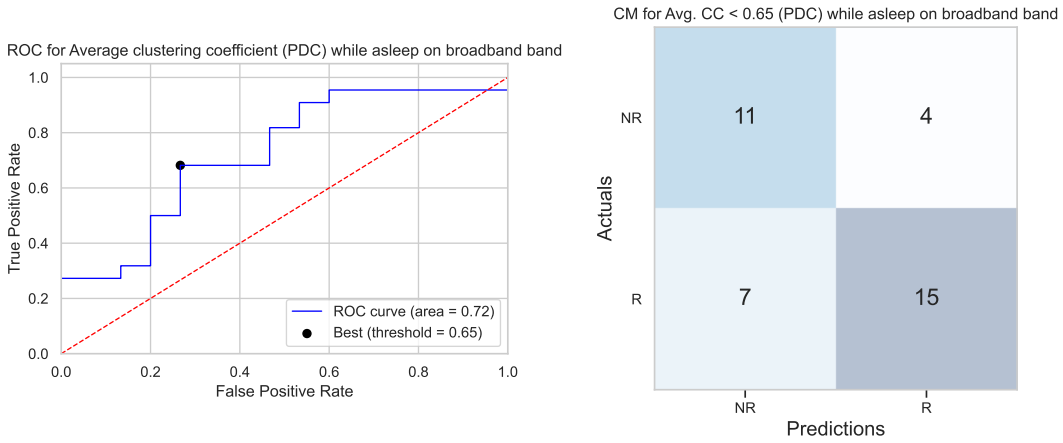


Figure 6.1.5: The ROC for the avg. CC on broadband and the confusion matrix for the best threshold obtained.

For this feature we got that the best threshold was to set as responders the patients with an average clustering coefficient below 0.65, and the performance obtained is shown in the table 6.1.5.

| **Avg. CC on broadband while sleeping (PDC)** | | |
|---|---|---|
| **Sensitivity** | **Specificity** | **Accuracy** |
| 0.68 [0.41, 0.83] | 0.73 [0.45, 0.92] | 0.70 [0.53, 0.84] |

Table 6.1.5: Performance of the avg. CC on broadband for the specified threshold.

Following, to study the relation between those two features we computed the correlation between them and obtained a value of 0.85, indicating a strong linear relation. The fact that the same graph measure presents significant differences between responders and non-responders in two frequency bands might be linked to the way the connectomes are generated. This is due to the fact that connectomes are generated by averaging connection matrices over frequency bins, so the topography of a connectome in a frequency band can have a strong influence over the topography of the connectome in the broadband.

### 6.1.3 Global efficiency on PDC connectomes while sleeping

Like the average clustering coefficient, the global efficiency demonstrated capacity to classify responders and non-responders in two frequency bands for the connectomes generated with the PDC and thresholded with the surrogate data test. The boxplot showing the distribution of this measure for all the patients is shown in the figure 6.1.6.

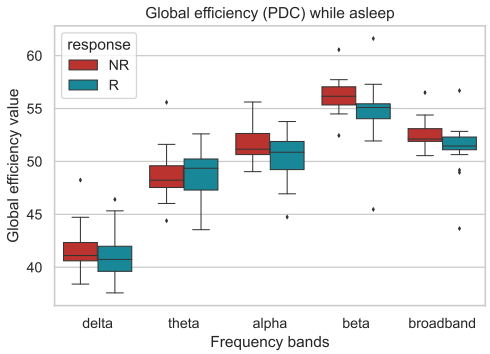


Figure 6.1.6: Boxplot for the GE on PDC connectomes while sleeping

The ROC curve and the confusion matrix for the best threshold for the global efficiency in the beta band is shown in the figure 6.1.7.

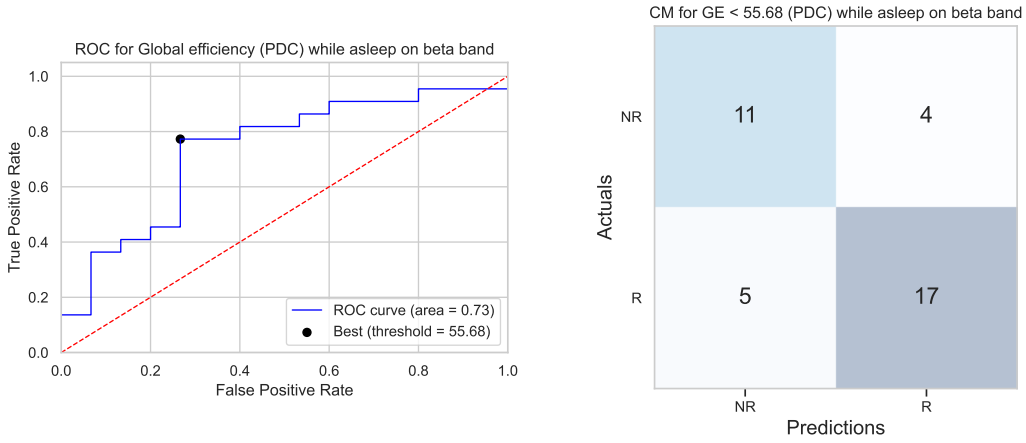


Figure 6.1.7: The ROC for the GE on beta band and the confusion matrix for the best threshold obtained.

For this feature the best split between responders and non-responders occur

by classifying as responders the patients with a global efficiency below 55.68. The sensitivity, specificity and accuracy for this feature are represented in the table 6.1.6.

| **GE on beta band while sleeping (PDC)** | | |
|---|---|---|
| **Sensitivity** | **Specificity** | **Accuracy** |
| 0.77 [0.55, 0.92] | 0.73 [0.45, 0.92] | 0.76 [0.59, 0.88] |

Table 6.1.6: Performance of the GE on beta band for the specified threshold.

Doing the same analysis for the global efficiency in the broadband, we obtained the ROC curve and the confusion matrix in the figure 6.1.8.

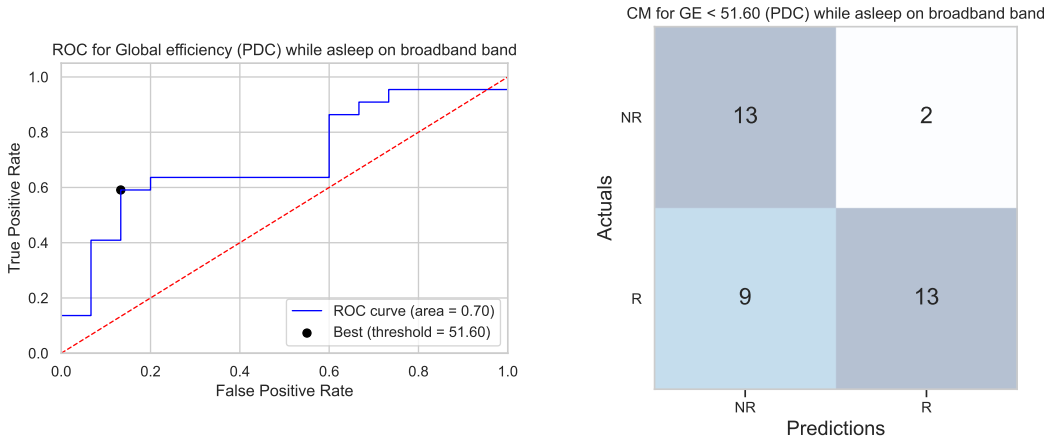


Figure 6.1.8: The ROC for the GE on broadband and the confusion matrix for the best threshold obtained.

The best threshold found for the GE in the broadband was to set as responders the patients with a GE below 51.6. The performance for this threshold is summarized in the table 6.1.7.

| GE on broadband while sleeping (PDC) | | |
|---|---|---|
| **Sensitivity** | **Specificity** | **Accuracy** |
| 0.59 [0.36, 0.79] | 0.87 [0.60, 0.98] | 0.70 [0.53, 0.84] |

Table 6.1.7: Performance of the GE on broadband for the specified threshold.

Once again, we studied the relation between the two features based on the same graph measure, and we obtained a correlation of 0.95 between them, which shows a strong linear relationship between the calculated measure in both frequency bands. Like for the average clustering coefficient, this linear relation might be explained by the way the connectomes were generated, that is, the topology of the beta band is reflected on the broadband.

### 6.1.4 Modularity on PDC connectomes while awake

Differently from the other features presented above, this one was obtained from connectomes generated with the binarization. Also, this graph measure showed significant differences between responders and non-responders in just one frequency band. The boxplot for this measure on the binarized connectomes is shown in the figure 6.1.9.

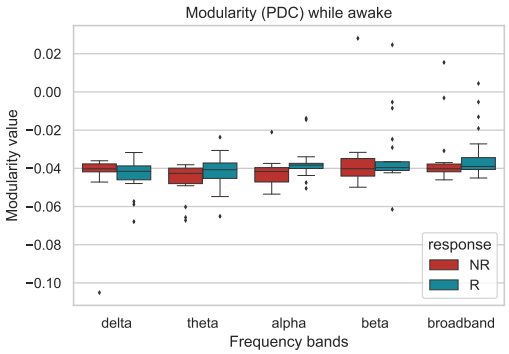


Figure 6.1.9: Boxplot for the modularity on PDC connectomes while awake

The boxplot shows that the modularity values are negative in most of the cases, which shows that the modular configuration adopted for the calculation of this measure does not divide the connectomes very well, that is, the connections happens more often between communities, not within them.

The figure 6.1.10 shows the ROC curve for the modularity in the alpha band and the confusion matrix obtained with the best threshold.

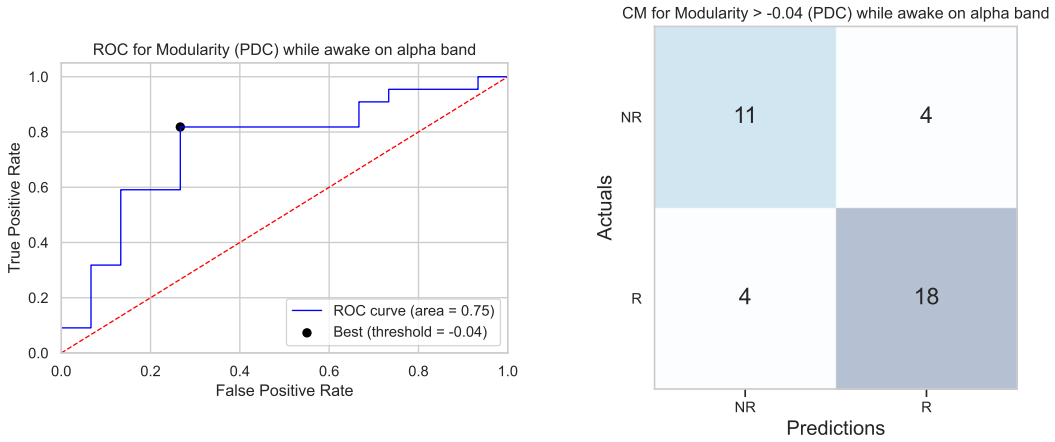


Figure 6.1.10: The ROC for the modularity on alpha band and the confusion matrix for the best threshold obtained.

The best classification setup consists in classifying as responder to VNS therapy the patients with a modularity higher than $-0.04$. The performance of this threshold is summarized in the table 6.1.8.

| **Modularity on alpha band while sleeping (PDC)** | | |
|---|---|---|
| **Sensitivity** | **Specificity** | **Accuracy** |
| 0.82 [0.60, 0.95] | 0.73 [0.45, 0.92] | 0.78 [0.62, 0.90] |

Table 6.1.8: Performance of the mod on alpha band for the specified threshold.

Differently from the other measures mentioned above, the effect of this measure is not seen on the broadband. This might be explained by the fact that the connectomes are binarized after averaging the connection matrices over the frequency bins, what reduces the influence of narrow frequency bands in the broadband. Moreover, another difference is the recording state of the EEGs used to generate the connectomes, because the other graph measure had bigger discrimination power on the sleeping state recordings, while the modularity showed a better performance in the EEG recordings performed on wakefulness state.

## 6.2 Comparison between the thresholding methods

Decide between the most adequate thresholding method for a connectome might be a difficult task, that may vary according to the application of the connectome. In this work we proposed three types of connectomes (not thresholded, thresholded with the surrogate data test, and binarized with a 50% density), so we performed two tests to decide which one of the connectomes we should use to extract the features of the machine learning models.

For the first comparison we calculated the AUC for the features obtained from each one of the connectomes and the sensitivity, specificity, and accuracy obtained with best threshold of this feature, similarly to what was done in the previous section. This analysis shows the individual capacity of each feature, that is, the capacity of the feature alone in classifying responders and non-responders.

The table C.0.1 in the appendix shows the 5 features with the highest AUC generated from each thresholding configuration. In the table 6.2.1 we summarize the general performance of the features obtained from each dataset by getting the average of the metrics used to measure the performance.

| **Thresholding method** | **Avg. AUC** | **Avg. specificity** | **Avg. Sensitivity** | **Avg. accuracy** |
|---|---|---|---|---|
| **Not thresholded** | 0.56 | 0.62 | 0.58 | 0.60 |
| **Surrogate data test** | 0.57 | 0.61 | 0.62 | 0.61 |
| **Binarized** | 0.57 | 0.60 | 0.62 | 0.61 |

Table 6.2.1: Averaged metrics for the features obtained from each thresholding method.

From those tables we can see that the relevant features from the connectomes generated with the surrogate data test thresholding method are more relevant individually for the classification method, a fact that is supported by the hypothesis test we performed in the previous section. However, the average performance of the features is similar for the features from this dataset and the dataset binarized according to a fixed density.

Following, to measure the capacity of the combined features in the machine learning task, we trained, validated the models, and compared their validation accuracy. For this assessment we used the whole datasets, not only the graph

measures, that is, each dataset used was composed by the graph measures extracted from the connectomes and the synchronization measures. With this dataset the feature selection methods described previously were used and the feature subset that lead to the best result was used.

The results obtained with this procedure are displayed in the table 6.2.2.

| | **Thresholding methods** | | |
|---|---|---|---|
| **Model** | **Not thresholded** | **Surrogate data test** | **Binarized with 50% density** |
| **SVM linear** | 0.800 | 0.867 | 0.867 |
| **SVM polynomial** | 0.833 | 0.933 | 0.867 |
| **SVM rbf** | 0.800 | 0.900 | 0.867 |
| **Gaussian process** | 0.800 | 0.900 | 0.833 |
| **K-NN** | 0.867 | 0.933 | 0.900 |
| **Decision Tree** | 0.700 | 0.667 | 0.633 |
| **Random Forest** | 0.633 | 0.667 | 0.700 |
| **AdaBoost** | 0.733 | 0.733 | 0.733 |

Table 6.2.2: Best validation accuracy obtained from the training on each dataset.

The objective of this comparison is to assess the capacity of the features set in predicting the outcome of the VNS therapy and analyze which thresholding method is the more appropriated for this task. From the table it is possible to notice that the dataset with the features extracted from the connectomes thresholded with the surrogate data test had the best validation accuracy in the biggest part of the models.

Consequently, given the performance obtained by the features individually and the performance of the whole dataset on the machine learning task, we decided to use the connectomes thresholded with the surrogate data test to build the final machine learning model.

## 6.3 Machine learning model selection

To train the first models we performed the feature selection methods and found the best hyperparameters for each model according to the dataset re-scaling or transformation adopted. At this stage we kept all the features in the dataset and

left the feature selection procedure entirely to the feature selection methods. As mentioned before, the hyperparameters were tuned according to the cross-validation accuracy obtained on the training set, and the test accuracy was obtained by testing the model performance on the 7 patients in the test set.

The models that obtained the best validation accuracy are shown in the table 6.3.1, with the test accuracy, and the best feature selection method.

| **Model** | **Val. acc.** | **Test acc.** | **Best feat. selec. method** | **Number of feat. selected** |
|---|---|---|---|---|
| **SVM linear** | 0.867 | 0.43 | MrMr | 11 |
| **SVM polynomial** | 0.933 | 0.57 | CBFS | 10 |
| **SVM rbf** | 0.900 | 0.43 | CBFS | 10 |
| **Gaussian process** | 0.900 | 0.57 | CBFS | 10 |
| **K-NN** | 0.930 | 0.57 | CBFS | 10 |
| **Decision Tree** | 0.733 | 0.57 | - | 3 |
| **Random Forest** | 0.670 | 0.57 | - | 54 |
| **AdaBoost** | 0.733 | 0.57 | - | 69 |

Table 6.3.1: Performance of the best models found using the whole dataset.

From the table it is possible to notice that the models have a poor test accuracy and a good validation accuracy, which may be a strong indicator of overfitting of the models to the training data. Moreover, the number of features selected is high for a dataset with 30 samples, which may also be an indicator of the overfitting.

One possible problem we noticed when analyzing the features selected by the methods is that some of the features selected were based on DTF connectomes. However, the features explored in section 6.1 were based in PDC connectomes, and according to the table C.0.1, the top 5 features extracted from the connectomes thresholded with the surrogate data test are based in PDC connectomes. This lack of agreement between the most relevant features and the features subset selected by the feature selection methods might be an indicator of poor performance of the methods due to excess of features.

Hence, because of that, we decided to manually remove some features from the original dataset and explore the performance of the models in a reduced dataset. The first reduction we preformed was to drop the features based on DTF connectomes, leaving just the features based on synchronization measures and PDC

connectomes. This elimination reduced the number of features of the whole dataset from 110 to 60.

The performances obtained after doing the same procedures as before on the reduced dataset are summarized in the table 6.3.2.

| Model | Val. acc. | Test acc. | Best feat. selec. method | Number of feat. selected |
|---|---|---|---|---|
| **SVM linear** | 0.800 | 0.71 | MrMr | 7 |
| **SVM polynomial** | 0.867 | 0.71 | CBFS | 7 |
| **SVM rbf** | 0.867 | 0.71 | MrMr | 10 |
| **Gaussian process** | 0.867 | 0.86 | MrMr | 7 |
| **K-NN** | 0.867 | 0.71 | MrMr | 6 |
| **Decision Tree** | 0.767 | 0.43 | - | 3 |
| **Random Forest** | 0.767 | 0.57 | - | 13 |
| **AdaBoost** | 0.800 | 0.57 | - | 43 |

Table 6.3.2: Performance of the best models found using the dataset without the measures from the DTF connectomes.

Indeed, the reduction of the dataset reduced the complexity of the models and the overfitting, because the validation accuracy got reduced while the test accuracy increased for most of the models. Another indicator that can be observed is the reduction in the number of features selected by the feature selection methods.

However, it is still possible to notice the overfitting still persists in some models, because they have a test accuracy lower than the validation accuracy. Also, the number of features is still high for models like the SVM with radial basis function kernel, the random forest, and the AdaBoost.

In a new try to reduce the complexity of the models, we decided to reduce the dataset again, this time removing the features obtained from wakefulness state EEGs. The measures with the highest AUCs in this dataset were extracted from sleeping state EEGs, so we decided to remove the features from awake state in a trial to obtain better models.

This procedure reduced the features number of the dataset from 60 to 30 and the results obtained after this are summarized in the table 6.3.3. With this table it is possible to notice that the models are not overfitted anymore, but the performance of some models is still low.

| Model | Val. acc. | Test acc. | Best feat. selec. method | Number of feat. Selected |
|---|---|---|---|---|
| **SVM linear** | 0.800 | 0.71 | CBFS | 6 |
| **SVM polynomial** | 0.800 | 0.71 | MrMr | 5 |
| **SVM rbf** | 0.830 | 0.71 | MrMr | 5 |
| **Gaussian process** | 0.800 | 0.71 | CBFS | 6 |
| **K-NN** | 0.867 | 0.86 | CBFS | 6 |
| **Decision Tree** | 0.800 | 0.43 | - | 3 |
| **Random Forest** | 0.700 | 0.57 | - | 14 |
| **AdaBoost** | 0.867 | 0.57 | - | 27 |

Table 6.3.3: Performance of the best models found using the dataset without the measures from the DTF connectomes and the measures from awake state EEGs.

In a last attempt to improve the performance of the models we decided to perform a manual feature selection, selecting the features that had an AUC bigger or equal than 0.7. In this method we used a dataset with the features based on synchronization measures, binarized connectomes and connectomes thresholded with the surrogate data test. The results obtained with this dataset are summarized in the table 6.3.4.

| Model | Val. acc. | Test acc. | Best feat. selec. method | Number of feat. Selected |
|---|---|---|---|---|
| **SVM linear** | 0.767 | 0.71 | SFS | 2 |
| **SVM polynomial** | 0.800 | 0.71 | MrMr | 3 |
| **SVM rbf** | 0.733 | 0.71 | SFS | 2 |
| **Gaussian process** | 0.767 | 0.71 | SFS | 3 |
| **K-NN** | 0.867 | 0.71 | MrMr | 5 |
| **Decision Tree** | 0.733 | 0.71 | - | 2 |
| **Random Forest** | 0.833 | 0.71 | - | 8 |
| **AdaBoost** | 0.800 | 0.57 | - | 9 |

Table 6.3.4: Performance of the best models found mixing synchronization measures and graph measures from the binarized connectomes and the connectomes thresholded with the surrogate data test.

The models trained with the dataset that had only the features with AUC

greater or equal than 0.7 performed worse than the other models trained above. Nonetheless, the models based on decision tree experimented a small increase in the performance. It is also possible to notice some models can achieve a reasonable accuracy with just 2 or 3 features. Moreover, the models agree in the importance of some features, because the features GE on beta band while sleeping (PDC) and Avg. CC on delta band while sleeping (PDC) were present in all the features subset, while the mean synchronization in beta band on sleeping state was present with them in the features subset with at least 3 features. Hence, we can conclude that those 3 features are the most important for the classification task, which is in agreement with the analysis from section 6.1.

From all the models tested, we got that the best were the Gaussian process from the dataset without the measures from DTF connectomes (table 6.3.2) and the K-NN from the dataset without measures from DTF connectomes and wakefulness state EEGs (table 6.3.3).

By comparing the best feature subset of those two models we can notice they are similar, as it can be noticed from the table 6.3.5.

| | **Gaussian process** | **K-NN** |
|---|---|---|
| **GE on beta band while sleeping (PDC)** | X | X |
| **GE on broadband while sleeping (PDC)** | X | X |
| **Avg. CC on delta band while sleeping (PDC)** | X | X |
| **Mod. on alpha band while awake (PDC)** | X | |
| **Mean wPLI on beta band while sleeping** | X | X |
| **Mean wPLI on broadband band while sleeping** | X | X |
| **Mean wPLI on alpha band while sleeping** | X | X |

Table 6.3.5: Features selected for each model (marked with an X).

We can see the models agree in the most important features and they have a similar performance regarding the cross-validation score and the test accuracy. Hence, we need to choose between one of those models as our final machine learning model.

## 6.4 Assessment of the final model

The first comparison we made between the models consisted in performing a permutation test with the models. The permutation test returns a permutation-based p-value that measures the chance of the observed performance being obtained by chance. The null hypothesis in this test is that the classifier fails to use patterns of the data to predict labels and the alternative hypothesis is that the model is able to use patterns in the data to predict the labels.

This statistical test is a non-parametric test with the p-value being calculated from a null distribution that is generated by permuting the labels and calculating the cross-validation score of the model on this new data. On each permutation the labels are randomly shuffled and a distribution for the model accuracy is obtained after performing this procedure many times.

We performed this test with 1000 permutations and 6-Fold cross-validation with random shuffle for both machine learning models, obtaining the null distributions shown in the figure 6.4.1.

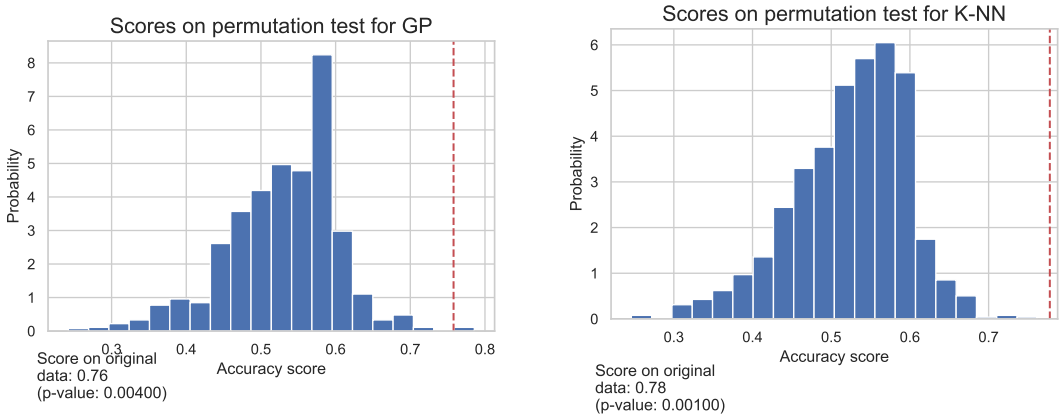


Figure 6.4.1: The null distribution for both models, obtained by shuffling the labels of the samples. The red line indicates the score obtained without permuting the labels.

From this test we can see the K-Nearest Neighbors model obtained a smaller p-value, indicating this model has a smaller chance of having a performance obtained by chance.

Consequently, we chose the K-NN model as the final model to predict the VNS therapy outcome. This model obtained a cross-validation accuracy of 0.867 and a

test accuracy of 0.86. From the 7 samples selected for the test set, 3 were patients with negative outcome and 4 were patients with positive outcome. From the 7 test samples the model just guessed wrong the label of one negative outcome (guessed as positive). Also, it is important to highlight the Gaussian process model guessed wrong the label of the same test sample as the K-NN model.

Finally, the chose of the K-NN over the Gaussian process is also motivated by the computational advantage of the K-NN model over the Gaussian process model. The K-NN is one of the simplest machine learning models and can be trained and tested faster than the Gaussian process, so it might be more interesting to adopt the K-NN model as the final choice.

# 7 Discussion

The results obtained in this work are aligned with the results reported in previous works, moreover, they are also aligned to what is expected from the behavior of epilepsy and VNS therapy. The epilepsy is characterized by an hypersynchronized state in brain's functional connectivity and it is believed that VNS works by promoting desynchronization and thus shifting the brain to a less synchronous state.

Consequently, the first important feature in this work is the synchronization estimator we calculated, the average wPLI, because the level of synchronization demonstrated to be able to classify responders and non-responders with a good performance. However, it is important to highlight that no significant differences were found between the average synchronization of responders and non-responders in the hypothesis tests, but this variable showed to be useful for the machine learning models and presented a good classification performance alone. In the classification of responders and non-responders using only this feature, it was found that classifying as responder the patients with higher levels of synchronization could lead to a reasonable classification performance.

The works from Fraschini et al. [3] and Bodin et al. [33] reported a lower level of synchronization in responders than in non-responders after VNS implantation, which reinforces the theory that VNS may work by inducing a desynchronization in brain functional connectivity. Also, complementing those results, Vespa et al. [4] compared the values of global synchronization on VNS ON and VNS OFF states, finding that a bigger induced desynchronization might be linked to a better response to VNS therapy. Although our results cannot be directly compared to the results of those works, because we used only pre-implantation data, the higher level of synchronization we found in responders might be related to a higher capacity in presenting a synchronization decrease after the VNS implantation.

With respect to the graph measures, the first important finding we made is

the significant difference in the average clustering coefficient of responders and non responders in the delta band and in the broadband. We found that the average clustering coefficient is lower in responders than in non-responders to the therapy, showing that the functional connectivity of non-responders has a more pronounced clustering behavior. This finding tells the functional connections of non-responders tend to occur within clusters, indicating the existence of densely connected regions. Moreover, this finding is in accordance with the findings of Babajani-Feremi et al. [21], that found a higher transitivity in non-responders to VNS therapy. Although the transitivity and the clustering coefficient are not the same measure, they differ by a normalization, with both being directly proportional to the number of triangles in a network, so the results of our work and the one from Babajani-Feremi et al. are in agreement.

Regarding the modularity, we also found significant differences between responders and non-responders for this graph measure, because according to our comparisons we found that non-responders have a higher modularity in the alpha band before VNS implantation. This difference shows that responders to VNS therapy have their functional connections more concentrated within the communities stablished for the modularity calculation, like shown in the figure 5.5.3. Furthermore, this finding is also in accordance to the discoveries made by Babajani-Feremi et al. [21], who also reported a smaller modularity in non-responders than in responders in alpha, beta, and theta bands.

The other possible biomarker we found is the global efficiency, that we discovered to be bigger in non-responders. A higher global efficiency is reflex of higher brain integration, which may be linked to a more pronounced epileptogenic state, so the modulation of the brain connectivity towards a less epileptogenic state may imply in a reduction of network integration [4] and consequently in lower values of GE. Also, this statement is reinforced by the findings of Carboni et al. [66], that found higher levels of GE in epileptic patients in comparison to healthy patients, so a higher level of network integration can indeed characterize a more pronounced epileptogenic state. Then, according to our findings, patients who present a more pronounced epileptic network configuration might be less responsive to VNS therapy, which is indicated by higher values of GE in non-responders to the therapy.

Also, this is also supported by the the study of Babajani-Feremi et al. [21], where the authors compared responders and non-responders to VNS therapy with healthy controls, finding that good responders to the therapy had values of transitivity, modularity, and characteristic path length closer to the ones presented by the healthy controls. This finding shows that good responders to the therapy might present a network configuration similar to the one of healthy patients. Hence, the

difference between responders and non-responders that we observed in GE follows this trend, because non-responders presented higher values of global efficiency, which characterize a network configuration more distant from the one expected for healthy patients.

# 8 Conclusion

The first objective of this master's thesis was to propose a pipeline configuration to extract features based on graph measures and functional connectivity from EEG recordings. The second objective was to explore and study the features extracted from this pipeline, looking for possible biomarkers in the outcome prediction of VNS therapy and assess the importance of those features in a machine learning task. In the search for biomarkers we performed statistical tests and assessed the performance of the features alone in the distinction between responders and non-responders to the therapy. Additionaly, for the machine learning models, we compared many subsets of features generated in multiple ways, with the objective of finding which combination of features would lead to the best results.

The proposed pipeline was implemented in Python and does not count with a guided user interface, which makes it not so friendly for people that are not familiarized with programming. However, this proposed implementation can work as a guideline for a future high level implementation, because it shows the potential of this method in a probable future application.

In this work we explored the different methods and measures that could be used in different steps of the pipeline. Namely, we explored three functional connectivity measures, three thresholding setups for the connectomes and five network measures. From this trials we found out that the most insightful and powerful features were derived from the synchronization estimation with the wPLI and from the PDC connectomes thresholded with the surrogate data test. Moreover, among the two recording states, sleeping and awake, we found out that the biggest part of the features and possible biomarkers came from the sleeping state records, but one possible biomarker could be found in the wakefulness recordings.

While looking for biomarkers in the connectomes thresholded with the surrogate data test, we found out that the global efficiency calculated from PDC connectomes on beta band and on broadband in sleeping state can work as possible biomarker

in the VNS therapy outcome prediction. Similarly, we have found that the average clustering coefficient on delta band and on broadband in sleeping state can also work as a possible biomarker. Additionaly, using the connectomes that were thresholded and binarized based on a fixed density, we have found that the modularity calculated on the alpha band in wakefulness EEG can be a biomarker too.

Furthermore, while performing an analysis with the other features that were not identified as possible biomarkers, we found out that the average wPLI calculated on beta band has a great discriminating power in the classification task and is an useful feature for the machine learning task.

Using the graph measures extracted from the connectomes thresholded with the surrogate data test and the average wPLI, we tested machine learning models and found the feature subsets that could lead to the best validation and test performance. Doing these procedures we trained two different machine learning models and they achieved a cross-validation accuracy of 86.7% and a test accuracy of 86%. The two machine learning models that achieved the best performances were the K-Nearest neighbors and the Gaussian process, then, to choose between one of the models we performed a permutation method, where the K-nearest neighbors method had the best performance. Moreover, the permutation test confirmed the relation between the features and the labels, proving that the labels can be inferred from the features.

In the other hand, this work count with some limitations, like the number of data available on the machine learning task and the fact that this is a retrospective study. To assess the performance of the machine learning models we counted with just 7 patients on the test set, so the assessment of the model test accuracy has a really low confidence. The other limitation comes from the fact we used retrospective data, so we cannot assure that the recordings were made in the exact same conditions.

There are some points that could be improved or explored in future works. The first point to be improved is the implementation of the pipeline, which can be implemented in a more user friendly way in a future work, aiming the adoption by physicians in clinical evaluations. A second point to be improved is the assessment of the machine learning models, because with more data it would be possible to get better assessments of models accuracy and probably better accuracy due to bigger availability of training samples. In the machine learning part we also tested just a few models and performed fast feature selection methods, so it might be interesting to test more models in a future work and greedy feature selection algorithms to confirm the choice of the best features subset.

With respect to the methods of the pipeline, it might be interesting to test different functional connectivity measures in the connectomes generation, because we just made an option for the estimators that were more used in the literature, but different estimators could lead to better results. Also, we tested just three thresholding setups, which is a quite limited number, so in a future work it would be interesting to explore different thresholding methods, mainly the ones based in graph properties, like the thresholding using a fixed "small-worldness" index. About the graph measures, future works can work and test different measures in the connectomes, because in this master's thesis we limited our analysis to five different measures, but many other graph measures exists and could also be used.

Finally, we can conclude this master's thesis achieved its objectives, because the proposed pipeline was used and the features obtained from it presented a great classification power, what was reinforced by the hypothesis tests and the machine learning models. However, there are still many points to be explored regarding the use of functional connectivity and graph measures in the VNS therapy outcome prediction, like the use of different functional connectivity estimators and graph measures.

# A Patients data

| Patient number | Sex | Response to therapy | Etiology | Epilepsy type |
|---|---|---|---|---|
| 1 | F | NR | unknown | focal |
| 2 | F | R | non-lesional | multifocal |
| 3 | F | R | non-lesional | focal |
| 4 | M | R | lesional | multifocal |
| 5 | M | R | non-lesional | generalized |
| 6 | F | NR | unknown | generalized |
| 7 | M | R | unknown | focal |
| 8 | F | NR | lesional | focal |
| 9 | M | R | unknown | focal |
| 10 | M | NR | lesional | focal |
| 11 | M | NR | lesional | multifocal |
| 12 | F | NR | non-lesional | focal |
| 13 | F | R | lesional | focal |
| 14 | F | NR | non-lesional and lesional | generalized and focal |
| 16 | F | R | lesional | unknown |
| 17 | F | R | unknown | focal |
| 18 | M | R | lesional | focal |
| 19 | M | R | lesional | focal |
| 20 | F | R | lesional | bifocal |
| 21 | M | NR | unknown | generalized |
| 22 | F | R | lesional | focal |
| 23 | M | NR | unknown | focal |
| 24 | M | R | lesional | multifocal |
| 25 | M | NR | lesional | focal |
| 26 | F | NR | lesional | focal |
| 27 | M | R | non-lesional | generalized |
| 28 | M | R | lesional | focal |
| 29 | F | NR | lesional | unknown |
| 30 | F | NR | lesional | focal |
| 31 | F | R | unknown | unknown |
| 32 | M | R | lesional | focal |
| 33 | M | R | unknown | generalized |
| 34 | M | NR | unknown | generalized and focal |
| 35 | F | NR | unknown | generalized |
| 36 | M | R | lesional | focal |
| 37 | F | R | unknown | generalized |
| 38 | F | R | unknown | focal |

Table A.0.1: Patients data used in the work. R and NR refer to responders and non-responders to VNS-therapy. The data of patient 15 was dicarded because his response was not determined.

# B P-values obtained for the features

| Not thresholded connectomes | | | | | |
|---|---|---|---|---|---|
| **Measure** | **delta** | **theta** | **alpha** | **beta** | **broadband** |
| **GE while awake (PDC)** | 0.494 | 0.494 | 0.494 | 0.494 | 0.494 |
| **GRC while awake (PDC)** | 0.294 | 0.294 | 0.294 | 0.294 | 0.294 |
| **Mod. while awake (PDC)** | 0.338 | 0.338 | 0.338 | 0.338 | 0.338 |
| **DA while awake (PDC)** | 0.263 | 0.263 | 0.263 | 0.263 | 0.263 |
| **Avg. CC while awake (PDC)** | 0.494 | 0.494 | 0.494 | 0.494 | 0.494 |
| **GE while sleeping (PDC)** | 0.123 | 0.123 | 0.123 | 0.123 | 0.123 |
| **GRC while sleeping (PDC)** | 0.338 | 0.338 | 0.338 | 0.338 | 0.338 |
| **Mod. while sleeping (PDC)** | 0.469 | 0.469 | 0.469 | 0.469 | 0.469 |
| **DA while sleeping (PDC)** | 0.108 | 0.125 | 0.139 | 0.150 | 0.108 |
| **Avg. CC while sleeping (PDC)** | 0.215 | 0.236 | 0.354 | 0.262 | 0.361 |
| **GE while awake (DTF)** | 0.354 | 0.354 | 0.361 | 0.354 | 0.354 |
| **GRC while awake (DTF)** | 0.197 | 0.197 | 0.197 | 0.197 | 0.197 |
| **Mod. while awake (DTF)** | 0.206 | 0.206 | 0.294 | 0.206 | 0.206 |
| **DA while awake (DTF)** | 0.490 | 0.494 | 0.490 | 0.490 | 0.494 |
| **Avg. CC while awake (DTF)** | 0.329 | 0.338 | 0.329 | 0.329 | 0.329 |
| **GE while sleeping (DTF)** | 0.327 | 0.327 | 0.327 | 0.327 | 0.327 |
| **GRC while sleeping (DTF)** | 0.349 | 0.349 | 0.349 | 0.349 | 0.349 |
| **Mod. while sleeping (DTF)** | 0.381 | 0.494 | 0.344 | 0.344 | 0.344 |
| **DA while sleeping (DTF)** | 0.296 | 0.481 | 0.481 | 0.481 | 0.481 |
| **Avg. CC while sleeping (DTF)** | 0.284 | 0.284 | 0.284 | 0.284 | 0.284 |

Table B.0.1: FDR corrected P-values on the Mann-Whitney U test for the features extracted from the not thresholded connectomes.

| Connectomes thresholded with the surrogate data test | | | | | |
|---|---|---|---|---|---|
| **Measure** | **delta** | **theta** | **alpha** | **beta** | **broadband** |
| **GE while awake (PDC)** | 0.494 | 0.494 | 0.494 | 0.494 | 0.494 |
| **GRC while awake (PDC)** | 0.384 | 0.384 | 0.384 | 0.384 | 0.384 |
| **Mod. while awake (PDC)** | 0.361 | 0.118 | 0.118 | 0.179 | 0.118 |
| **DA while awake (PDC)** | 0.349 | 0.349 | 0.349 | 0.349 | 0.349 |
| **Avg. CC while awake (PDC)** | 0.457 | 0.457 | 0.457 | 0.457 | 0.457 |
| **GE while sleeping (PDC)** | 0.170 | 0.361 | 0.166 | 0.045 | 0.050 |
| **GRC while sleeping (PDC)** | 0.481 | 0.481 | 0.481 | 0.481 | 0.481 |
| **Mod. while sleeping (PDC)** | 0.445 | 0.445 | 0.445 | 0.445 | 0.445 |
| **DA while sleeping (PDC)** | 0.189 | 0.179 | 0.179 | 0.179 | 0.179 |
| **Avg. CC while sleeping (PDC)** | 0.024 | 0.316 | 0.179 | 0.133 | 0.029 |
| **GE while awake (DTF)** | 0.349 | 0.349 | 0.349 | 0.349 | 0.349 |
| **GRC while awake (DTF)** | 0.258 | 0.349 | 0.258 | 0.258 | 0.258 |
| **Mod. while awake (DTF)** | 0.374 | 0.384 | 0.384 | 0.374 | 0.374 |
| **DA while awake (DTF)** | 0.269 | 0.338 | 0.250 | 0.250 | 0.250 |
| **Avg. CC while awake (DTF)** | 0.253 | 0.253 | 0.253 | 0.253 | 0.253 |
| **GE while sleeping (DTF)** | 0.396 | 0.396 | 0.396 | 0.396 | 0.396 |
| **GRC while sleeping (DTF)** | 0.230 | 0.408 | 0.408 | 0.408 | 0.408 |
| **Mod. while sleeping (DTF)** | 0.344 | 0.344 | 0.349 | 0.344 | 0.349 |
| **DA while sleeping (DTF)** | 0.481 | 0.201 | 0.481 | 0.481 | 0.494 |
| **Avg. CC while sleeping (DTF)** | 0.420 | 0.420 | 0.420 | 0.420 | 0.420 |

Table B.0.2: FDR corrected P-values on the Mann-Whitney U test for the features extracted from the connectomes thresholded with the surrogate data test. The values below 0.05 are marked in red.

| Binarized connectomes with 50% censity | | | | | |
|---|---|---|---|---|---|
| **Measure** | **delta** | **theta** | **alpha** | **beta** | **broadband** |
| **GE while awake (PDC)** | 0.424 | 0.424 | 0.424 | 0.424 | 0.424 |
| **GRC while awake (PDC)** | 0.176 | 0.485 | 0.485 | 0.123 | 0.176 |
| **Mod. while awake (PDC)** | 0.177 | 0.177 | 0.027 | 0.177 | 0.177 |
| **DA while awake (PDC)** | 0.202 | 0.417 | 0.417 | 0.080 | 0.202 |
| **Avg. CC while awake (PDC)** | 0.365 | 0.365 | 0.365 | 0.365 | 0.365 |
| **GE while sleeping (PDC)** | 0.197 | 0.211 | 0.197 | 0.197 | 0.197 |
| **GRC while sleeping (PDC)** | 0.219 | 0.219 | 0.420 | 0.219 | 0.219 |
| **Mod. while sleeping (PDC)** | 0.223 | 0.133 | 0.315 | 0.361 | 0.361 |
| **DA while sleeping (PDC)** | 0.469 | 0.469 | 0.469 | 0.469 | 0.354 |
| **Avg. CC while sleeping (PDC)** | 0.154 | 0.154 | 0.384 | 0.154 | 0.154 |
| **GE while awake (DTF)** | 0.494 | 0.494 | 0.494 | 0.115 | 0.494 |
| **GRC while awake (DTF)** | 0.488 | 0.488 | 0.488 | 0.102 | 0.488 |
| **Mod. while awake (DTF)** | 0.423 | 0.422 | 0.422 | 0.422 | 0.457 |
| **DA while awake (DTF)** | 0.338 | 0.338 | 0.338 | 0.338 | 0.338 |
| **Avg. CC while awake (DTF)** | 0.247 | 0.494 | 0.247 | 0.247 | 0.247 |
| **GE while sleeping (DTF)** | 0.437 | 0.437 | 0.437 | 0.457 | 0.437 |
| **GRC while sleeping (DTF)** | 0.361 | 0.394 | 0.361 | 0.481 | 0.443 |
| **Mod. while sleeping (DTF)** | 0.361 | 0.361 | 0.361 | 0.361 | 0.445 |
| **DA while sleeping (DTF)** | 0.143 | 0.143 | 0.143 | 0.143 | 0.143 |
| **Avg. CC while sleeping (DTF)** | 0.420 | 0.420 | 0.420 | 0.164 | 0.420 |

Table B.0.3: FDR corrected P-values on the Mann-Whitney U test for the features extracted from the binarized connectomes with 50% edge density. The values below 0.05 are marked in red.

# C Thresholding methods features

| Feature | AUC | Sensitivity | Specificity | Accuracy |
|---|---|---|---|---|
| **Not thresholded** | | | | |
| DA on delta band while sleeping (PDC) | 0.67 | 0.77 | 0.60 | 0.70 |
| DA on broadband while sleeping (PDC) | 0.67 | 0.68 | 0.67 | 0.68 |
| Avg. CC on delta band while sleeping (PDC) | 0.67 | 0.73 | 0.60 | 0.68 |
| DA on delta band while sleeping (DTF) | 0.65 | 0.77 | 0.60 | 0.70 |
| GRC on alpha band while awake (DTF) | 0.65 | 0.82 | 0.53 | 0.70 |
| **Surrogate data test** | | | | |
| Avg. CC on delta band while sleeping (PDC) | 0.75 | 0.82 | 0.67 | 0.76 |
| GE on beta band while sleeping (PDC) | 0.73 | 0.77 | 0.73 | 0.76 |
| Avg. CC on broadband while sleeping (PDC) | 0.72 | 0.68 | 0.73 | 0.70 |
| GE on broadband while sleeping (PDC) | 0.70 | 0.59 | 0.87 | 0.70 |
| MOD on theta band while awake (PDC) | 0.69 | 0.59 | 0.80 | 0.68 |
| Binarized | | | | |
| MOD on alpha band while awake (PDC) | 0.75 | 0.82 | 0.73 | 0.78 |
| DA on beta band while wake (PDC) | 0.70 | 0.73 | 0.60 | 0.68 |
| GRC on beta band while awake (DTF) | 0.70 | 0.91 | 0.53 | 0.76 |
| GE on beta band while awake (DTF) | 0.70 | 0.91 | 0.53 | 0.76 |
| MOD on theta band while sleeping (PDC) | 0.69 | 0.68 | 0.67 | 0.68 |

Table C.0.1: Five features with the biggest AUC for each connectome thresholding setup. The features in red had a p-value below 0.05 in the Mann-Whitney U test.

# Bibliography


[1] Epilepsy, 2022. URL `https://www.who.int/news-room/fact-sheets/detail/epilepsy`. Visited on 20/05/2022.

[2] P.O. Shafer and P.M. Dean. Vagus nerve stimulation (vns) therapy, 2018. URL `https://www.epilepsy.com/treatment/devices/vagus-nerve-stimulation-therapy`. Visited on 20/05/2022.

[3] Matteo Fraschini, Monica Puligheddu, Matteo Demuru, Lorenzo Polizzi, Alberto Maleci, Giorgio Tamburini, Socrate Congia, Marco Bortolato, and Francesco Marrosu. VNS induced desynchronization in gamma bands correlates with positive clinical outcome in temporal lobe pharmacoresistant epilepsy. *Neuroscience Letters*, 536:14–18, mar 2013. doi: 10.1016/j.neulet.2012.12.044.

[4] Simone Vespa, Jolan Heyse, Lars Stumpp, Giulia Liberati, Susana Ferrao Santos, Herbert Rooijakkers, Antoine Nonclercq, André Mouraux, Pieter van Mierlo, and Riëm El Tahry. Vagus nerve stimulation elicits sleep EEG desynchronization and network changes in responder patients in epilepsy. *Neurotherapeutics*, 18(4):2623–2638, October 2021. doi: 10.1007/s13311-021-01124-4.

[5] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, feb 2009. doi: 10.1038/nrn2575.

[6] Danielle Smith Bassett and Ed Bullmore. Small-world brain networks. *The Neuroscientist*, 12(6):512–523, dec 2006. doi: 10.1177/1073858406293182.

[7] Jinhui Wang, Liang Wang, Yufeng Zang, Hong Yang, Hehan Tang, Qiyong Gong, Zhang Chen, Chaozhe Zhu, and Yong He. Parcellation-dependent small-world brain functional networks: A resting-state fMRI study. *Human Brain Mapping*, 30(5):1511–1523, may 2009. doi: 10.1002/hbm.20623.

[8] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059–1069, September 2010. doi: 10.1016/j.neuroimage.2009.10.003.

[9] Pieter van Mierlo, Margarita Papadopoulou, Evelien Carrette, Paul Boon, Stefaan Vandenberghe, Kristl Vonck, and Daniele Marinazzo. Functional brain connectivity from EEG in epilepsy: Seizure prediction and epileptogenic focus localization. *Progress in Neurobiology*, 121:19–35, October 2014. doi: 10.1016/j.pneurobio.2014.06.004.

[10] Mary A.B. Brazier. Spread of seizure discharges in epilepsy: Anatomical and electrophysiological considerations. *Experimental Neurology*, 36(2):263–272, aug 1972. doi: 10.1016/0014-4886(72)90022-2.

[11] Zhe Wang, Ahmed Alahmadi, David Zhu, and Tongtong Li. Brain functional connectivity analysis using mutual information. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, dec 2015. doi: 10.1109/globalsip.2015.7418254.

[12] R. Kus, M. Kaminski, and K.J. Blinowska. Determination of EEG activity propagation: Pair-wise versus multichannel estimate. *IEEE Transactions on Biomedical Engineering*, 51(9):1501–1510, September 2004. doi: 10.1109/tbme.2004.827929.

[13] J. Toppi, F. De Vico Fallani, G. Vecchiato, A. G. Maglione, F. Cincotti, D. Mattia, S. Salinari, F. Babiloni, and L. Astolfi. How the statistical validation of functional connectivity patterns can prevent erroneous definition of small-world properties of a brain connectivity network. *Computational and Mathematical Methods in Medicine*, 2012:1–13, 2012. doi: 10.1155/2012/130985.

[14] J. Kiffin Penry and J. Christine Dean. Prevention of intractable partial seizures by intermittent vagal stimulation in humans: Preliminary results. *Epilepsia*, 31(s2):S40–S43, June 1990. doi: 10.1111/j.1528-1157.1990.tb05848.x.

[15] Judith Scherrmann, Christian Hoppe, Thomas Kral, Johannes Schramm, and Christian E. Elger. Vagus nerve stimulation. *Journal of Clinical Neurophysiology*, 18(5):408–414, September 2001. doi: 10.1097/00004691-200109000-00004.

[16] J Janszky. Vagus nerve stimulation: predictors of seizure freedom. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(3):384–389, March 2005. doi: 10.1136/jnnp.2004.037085.

[17] Ahmet Z. Burakgazi, Evren Burakgazi-Dalkilic, Anthony J. Caputy, and

Samuel J. Potolicchio. The correlation between vagus nerve stimulation efficacy and partial onset epilepsies. *Journal of Clinical Neurophysiology*, 28 (4):380–383, August 2011. doi: 10.1097/wnp.0b013e318227333a.

[18] Milan Brázdil, Irena Doležalová, Eva Koritáková, Jan Chládek, Robert Roman, Martin Pail, Pavel Jurák, Daniel J. Shaw, and Jan Chrastina. EEG reactivity predicts individual efficacy of vagal nerve stimulation in intractable epileptics. *Frontiers in Neurology*, 10, May 2019. doi: 10.3389/fneur.2019.00392.

[19] Premysl Jiruska, Marco de Curtis, John G. R. Jefferys, Catherine A. Schevon, Steven J. Schiff, and Kaspar Schindler. Synchronization and desynchronization in epilepsy: controversies and hypotheses. *The Journal of Physiology*, 591(4): 787–797, January 2013. doi: 10.1113/jphysiol.2012.239590.

[20] Fabrice Bartolomei, Francesca Bonini, Elsa Vidal, Agnes Trébuchon, Stanislas Lagarde, Isabelle Lambert, Aileen McGonigal, Didier Scavarda, Romain Carron, and Christian G. Benar. How does vagal nerve stimulation (VNS) change EEG brain functional connectivity? *Epilepsy Research*, 126:141–146, October 2016. doi: 10.1016/j.eplepsyres.2016.06.008.

[21] Abbas Babajani-Feremi, Negar Noorizadeh, Basanagoud Mudigoudar, and James W. Wheless. Predicting seizure outcome of vagus nerve stimulation using MEG-based network topology. *NeuroImage: Clinical*, 19:990–999, 2018. doi: 10.1016/j.nicl.2018.06.017.

[22] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7(267):1–13, 2013. doi: 10.3389/fnins.2013.00267.

[23] Martin Billinger, Clemens Brunner, and Gernot R. Müller-Putz. SCoT: a python toolbox for EEG source connectivity. *Frontiers in Neuroinformatics*, 8, March 2014. doi: 10.3389/fninf.2014.00022.

[24] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[25] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `https://igraph.org`. Visited on 21/03/2022.

[26] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[27] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021.

[28] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[30] Aleksandra Miljevic, Neil W. Bailey, Fidel Vila-Rodriguez, Sally E. Herring, and Paul B. Fitzgerald. Electroencephalographic connectivity: A fundamental guide and checklist for optimal study design and evaluation. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, November 2021. doi: 10.1016/j.bpsc.2021.10.017.

[31] O Bertrand, F Perrin, and J Pernier. A theoretical justification of the average reference in topographic evoked potential studies. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 62(6):462–464, November 1985. doi: 10.1016/0168-5597(85)90058-9.

[32] Esther Florin, Joachim Gross, Johannes Pfeifer, Gereon R. Fink, and Lars Timmermann. The effect of filtering on granger causality based multivariate causality measures. *NeuroImage*, 50(2):577–588, April 2010. doi: 10.1016/j.neuroimage.2009.12.050.

[33] Clémentine Bodin, Sandrine Aubert, Géraldine Daquin, Romain Carron, Didier Scavarda, Aileen McGonigal, and Fabrice Bartolomei. Responders to vagus nerve stimulation (VNS) in refractory epilepsy have reduced interictal cortical synchronicity on scalp EEG. *Epilepsy Research*, 113:98–103, July 2015. doi: 10.1016/j.eplepsyres.2015.03.018.

[34] Cornelis J. Stam, Guido Nolte, and Andreas Daffertshofer. Phase lag index: Assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources. *Human Brain Mapping*, 28(11): 1178–1193, 2007. doi: 10.1002/hbm.20346.

[35] Martin Vinck, Robert Oostenveld, Marijn van Wingerden, Franscesco Battaglia,

and Cyriel M.A. Pennartz. An improved index of phase-synchronization for electrophysiological data in the presence of volume-conduction, noise and sample-size bias. *NeuroImage*, 55(4):1548–1565, April 2011. doi: 10.1016/j.neuroimage.2011.01.055.

[36] Laura Sophie Imperatori, Monica Betta, Luca Cecchetti, Andrés Canales-Johnson, Emiliano Ricciardi, Francesca Siclari, Pietro Pietrini, Srivas Chennu, and Giulio Bernardi. EEG functional connectivity metrics wPLI and wSMI account for distinct types of brain functional interactions. *Sci. Rep.*, 9(1):8894, June 2019.

[37] Luiz A. Baccalá and Koichi Sameshima. Partial directed coherence: a new concept in neural structure determination. *Biological Cybernetics*, 84(6): 463–474, May 2001. doi: 10.1007/PL00007990.

[38] M. J. Kaminski and K. J. Blinowska. A new method of the description of the information flow in the brain structures. *Biological Cybernetics*, 65(3):203–210, Jul 1991. ISSN 1432-0770. doi: 10.1007/BF00198091.

[39] Alois Schlögl. *The electroencephalogram and the adaptive autoregressive model: theory and applications.* Berichte aus der Medizinischen Informatik und Bioinformatik. Shaker-Verlag GmbH, Germany, 2000. ISBN 3-8265-7640-3.

[40] Ana Coito, Christoph M. Michel, Pieter van Mierlo, Serge Vulliemoz, and Gijs Plomp. Directed functional brain connectivity based on EEG source imaging: Methodology and application to temporal lobe epilepsy. *IEEE Transactions on Biomedical Engineering*, 63(12):2619–2628, December 2016. doi: 10.1109/tbme.2016.2619665.

[41] L. Astolfi, F. Cincotti, D. Mattia, C. Babiloni, F. Carducci, A. Basilisco, P.M. Rossini, S. Salinari, L. Ding, Y. Ni, B. He, and F. Babiloni. Assessing cortical functional connectivity by linear inverse estimation and directed transfer function: simulations and application to real data. *Clinical Neurophysiology*, 116(4):920–932, April 2005. doi: 10.1016/j.clinph.2004.10.012.

[42] L. Faes, A. Porta, and G. Nollo. Surrogate data approaches to assess the significance of directed coherence: Application to EEG activity propagation. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, September 2009. doi: 10.1109/iembs.2009.5332477.

[43] Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19), October 2001. doi: 10.1103/physrevlett.87.198701.

[44] M. Carboni, M. Rubega, G.R. Iannotti, P. De Stefano, G. Toscano, S. Tourbier, F. Pittau, P. Hagmann, S. Momjian, K. Schaller, M. Seeck, C.M. Michel, P. van Mierlo, and S. Vulliemoz. The network integration of epileptic activity in relation to surgical outcome. *Clinical Neurophysiology*, 130(12):2193–2202, December 2019. doi: 10.1016/j.clinph.2019.09.006.

[45] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998. doi: 10.1038/30918.

[46] Maher A. Quraan, Cornelia McCormick, Melanie Cohn, Taufik A. Valiante, and Mary Pat McAndrews. Altered resting state brain dynamics in temporal lobe epilepsy can be observed in spectral power, functional connectivity and graph theory metrics. *PLoS ONE*, 8(7):e68609, July 2013. doi: 10.1371/journal.pone.0068609.

[47] Giorgio Fagiolo. Clustering in complex directed networks. *Physical Review E*, 76(2), August 2007. doi: 10.1103/physreve.76.026107.

[48] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70(5), November 2004. doi: 10.1103/physreve.70.056131.

[49] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Physical Review Letters*, 100(11), March 2008. doi: 10.1103/physrevlett.100.118703.

[50] David E Anderson, Deepak Madhavan, and Arun Swaminathan. Global brain network dynamics predict therapeutic responsiveness to cannabidiol treatment for refractory epilepsy. *Brain Communications*, 2(2), 2020. doi: 10.1093/braincomms/fcaa140.

[51] Xinyun Hu and Gabriel Lodewijks. Detecting fatigue in car drivers and aircraft pilots by using non-invasive measures: The value of differentiation of sleepiness and mental fatigue. *Journal of Safety Research*, 72:173–187, feb 2020. doi: 10.1016/j.jsr.2019.12.015.

[52] Enys Mones, Lilla Vicsek, and Tamás Vicsek. Hierarchy measure for complex networks. *PLoS ONE*, 7(3):e33799, March 2012. doi: 10.1371/journal.pone.0033799.

[53] Eric van Diessen, Judith I. Hanemaaijer, Willem M. Otte, Rina Zelmann, Julia Jacobs, Floor E. Jansen, François Dubeau, Cornelis J. Stam, Jean Gotman, and Maeike Zijlmans. Are high frequency oscillations associated with altered

network topology in partial epilepsy? *NeuroImage*, 82:564–573, November 2013. doi: 10.1016/j.neuroimage.2013.06.031.

[54] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2), February 2003. doi: 10.1103/physreve.67.026126.

[55] Stephan Bialonski and Klaus Lehnertz. Assortative mixing in functional brain networks during epileptic seizures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(3):033139, September 2013. doi: 10.1063/1.4821915.

[56] Jacob G. Foster, David V. Foster, Peter Grassberger, and Maya Paczuski. Edge direction and the structure of networks. *Proceedings of the National Academy of Sciences*, 107(24):10815–10820, May 2010. doi: 10.1073/pnas.0912671107.

[57] G. V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):306–307, jul 1979. doi: 10.1109/tpami.1979.4766926.

[58] Mark A. Hall. Correlation-based feature selection of discrete and numeric class machine learning. Technical report, University of Waikato, Department of Computer Science, 2000. Working Paper.

[59] Johannes S. Fischer. Correlation-based feature selection in python from scratch, Aug 2021. URL `https://johfischer.com/2021/08/06/correlation-based-feature-selection-in-python-from-scratch/#:~:text=The%20correlation%2Dbased%20feature%20selection%20(CFS)%20method%20is%20a,the%20name%20already%20suggest%3A%20correlations`. Visited on 17/04/2022.

[60] Zhenyu Zhao, Radhika Anand, and Mallory Wang. Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, oct 2019. doi: 10.1109/dsaa.2019.00059.

[61] Samuele Mazzanti. "mrmr" explained exactly how you wished someone explained to you, Feb 2022. URL `https://towardsdatascience.com/mrmr-explained-exactly-how-you-wished-someone-explained-to-you-9cf4ed27458b`. Visited on 17/04/2022.

[62] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.

[63] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced*

*Lectures on Machine Learning*, pages 63–71. Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-28650-9_4.

[64] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997. doi: 10.1006/jcss.1997.1504.

[65] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. ISSN 00359246. URL `http://www.jstor.org/stable/2346101`.

[66] Margherita Carboni, Pia De Stefano, Bernd J. Vorderwülbecke, Sebastien Tourbier, Emeline Mullier, Maria Rubega, Shahan Momjian, Karl Schaller, Patric Hagmann, Margitta Seeck, Christoph M. Michel, Pieter van Mierlo, and Serge Vulliemoz. Abnormal directed connectivity of resting state networks in focal epilepsy. *NeuroImage: Clinical*, 27:102336, 2020. doi: 10.1016/j.nicl.2020.102336.

**UNIVERSITÉ CATHOLIQUE DE LOUVAIN**
**École polytechnique de Louvain**
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**