# Network analysis to characterize neurodegenerative diseases

Xavier Rodoreda Pitarch

July 1, 2022

**Abstract–** Multiple sclerosis is a neurodegenerative disease that affects the immune system and the central nervous system. Network analysis in multilayer architectures in brains can provide information about a person's condition regarding the disease, even more than singlelayer network analysis. Analysis along with Machine Learning techniques can save time and resources in the classification of healthy and unhealthy brains, and the classification of groups of unhealthy brains. In this project an accuracy of 86% in the classification task was obtained. The analysis may also reflect the most characteristic areas of the brain with respect to the disease.

**Keywords–** Multiple Sclerosis, Graph, Analysis, Multilayer, Machine Learning, Brain

**Resum–** L'esclerosis múltiple és una malaltia neurodegenerativa que afecta el sistema immunitari i al sistema nerviós central. L'anàlisi de xarxes en arquitectures multicapa en cervells pot donar informació sobre l'estat d'una persona respecte a la malaltia, més inclús que l'anàlisi de xarxes d'una sola capa. L'anàlisi junt amb tècniques d'Aprenentatge Automàtic pot estalviar temps i recursos en la classificació entre cervells sans i no sans, i la classificació dels grups dels cervells no sans. En aquest projecte s'ha obtingut una precisió del 86% en la classificació. L'anàlisi també pot reflectir les zones del cervell més característiques respecte a la malaltia.

**Paraules clau–** Esclerosis Múltiple, Graf, Anàlisi, Multicapa, Aprenentatge Automàtic, Cervell

## 1 Introduction

Multiple sclerosis (MS) is a neurodegenerative disease that causes an immune abnormality and affects the central nervous system. Those who suffer from the disease may present it in different ways and with different symptoms such as coordination and balance problems, muscle weakness, visual disturbances, difficulties on thinking and memorizing, itching, stinging or numbness, among other things. It is unknown to cause the disease and is not inherited or contagious. There are 3 different classification groups, depending on the course of the disease in which the patients are. The groups are:

Relapsing-remitting multiple sclerosis (RRMS): is characterized by clearly defined attacks of new or increasing neurologic symptoms. These attacks – also called relapses or exacerbations – are followed by periods of partial or complete recovery (remissions). During remissions, all symptoms may disappear, or some symptoms may continue and become permanent. However, there is no apparent progression of the disease during the periods of remission. [9]

Primary progressive multiple sclerosis (PPMS): is characterized by worsening neurologic function (accumulation of disability) from the onset of symptoms, without early relapses or remissions. PPMS can be further characterized as either active (with an occasional relapse and/or evidence of new MRI activity over a specified period of time) or not active, as well as with progression (evidence of disability accrual over time, with or without relapse or new MRI activity) or without progression. [10]

Secondary progressive multiple sclerosis (SPMS): follows an initial relapsing-remitting course. Some people who are diagnosed with RRMS will eventually transition to a secondary progressive course in which there is a progressive worsening of neurologic function (accumulation of disability) over time. SPMS can be further characterized as either active (with relapses and/or evidence of new MRI activity during a specified period of time) or not active, as well as with progression (evidence of disability accrual over time, with or without relapses or new MRI activity) or with-

• Contact E-mail: xavier.rodoreda.pitarch@gmail.com

• Project tutored by: Jordi Casas Roma (Àrea de Ciències de la Computació i Intel·ligència Artificial)

• Course 2021/22

out progression. [11]

It is possible to represent the brain of a person with the anatomical parcellation thanks to a representation in the form of graph applying certain metrics of the brain. It is known that the connections of certain parts of the brain, in some metrics, may be different in a healthy person than in a person with MS, and certain differences can be perceived by applying network analysis to the brain. If we also put together several representations in a graph with different layers, a multilayer graph, we think that we could extract more information from a network analysis than with single graph network analysis.

In this project we aim to study the differences between healthy brains and patients brains that suffer multiple sclerosis by network analysis using a multilayer graph architecture and machine learning.

## 2 State of the Art

The graph multilayer architectures is a relative recent concept and at the moment it is not used too much. However, the mathematical formulations of the concept have been around for several years, as we can see in this paper [12]. Moreover, in other disciplines, graph multilayer architecture have been used since some years ago [13].

Related to MS, there are some research articles where the authors use data science with the aim of solve problems related to the disease. There are articles that explain the possibility of classifying MS patients into different groups through machine learning [1], [8], works where brains with MS are analyzed from network analysis perspective [2] and articles that compare different groups of MS patients across multilayer architectures [3].

What sets this work from other published projects is that it aims to use a multilayer graph architecture to extract graph node metrics that could be used to build a machine learning classifier model capable of classifying a person's brain in specific MS group. To the best of our knowledge, there is currently no published paper attempting to use a multilayer graph architecture of brains to classify MS control groups from machine learning. There are research that uses multilayer graph analysis in brains with MS, and research that classify control groups through machine learning, but neither uses both concepts at once.

## 3 Objectives

The main objective of this project is to analyze brain dysfunction in the context of multiple sclerosis. This primary objective could be divide in the following objectives:

1. Study the information contained in each of the three layers.
2. Analyze whether it is possible to distinguish between healthy people and patients (and classify patients into groups) using the information from the three layers.
3. Develop a multilayer model to join the information of three layers.
4. Adapt graph metrics in the three-layer model to work with the developed architecture.

## 4 Methodology

The methodology that will be used to carry out this project will be Crisp-DM, the most typical methodology in data mining. It is a flexible methodology that always allows us to go back in its phases, if necessary. It will contain the following phases:

1. Business understanding (problem understanding): Understand the problem to be solved, how we want to solve it and what results we expect to obtain.
2. Data understanding: Once we have the data, we need to understand exactly what they mean, with the help of statistical analysis and domain experts.
3. Data Preparation: Make the necessary transformations to the data from the conclusions of the previous phase.
4. Modeling: Creation of a multilayer model, extract graph theoretical metrics and train a classification model.
5. Evaluation: Evaluate the model and analyse the results.

All code will be implemented in Python, and publicy shared via GitHub. There will be regular meetings every two weeks between the tutor and the student to track the project [1].

## 5 Planning

According to the methodological steps discussed above, we planned the following tasks to fulfill the project (see figure 1):

1. Explore and understand data.
2. Network analysis through graph metrics to identify which areas of the brain, graphs and metrics are most significant for predicting the state of the disease.
3. Implementation of a multilayer graph architecture that join, if possible, the three layers.
4. Network analysis of the multilayer graph with the aim of extracting metrics of some brain parts that are significant for the construction of the classification model.
5. Extraction of data from the multilayer graph to construct the predictive model.
6. Construct a classification model capable of distinguishes between brains that suffers from MS and brains that not suffer from it, and in case the brains suffers from MS, classify in what group belongs to, using machine learning.

In figure 2 we can see a diagram that describes more clearly all the steps followed to develop the project, separated according the sections of the project: blue boxes references the steps done on section 7, green boxes references the steps done on section 8 and orange boxes references the steps done on section 9.

[1] https://github.com/xavierRodo/TFGMalaltiesNeurodegeneratives/

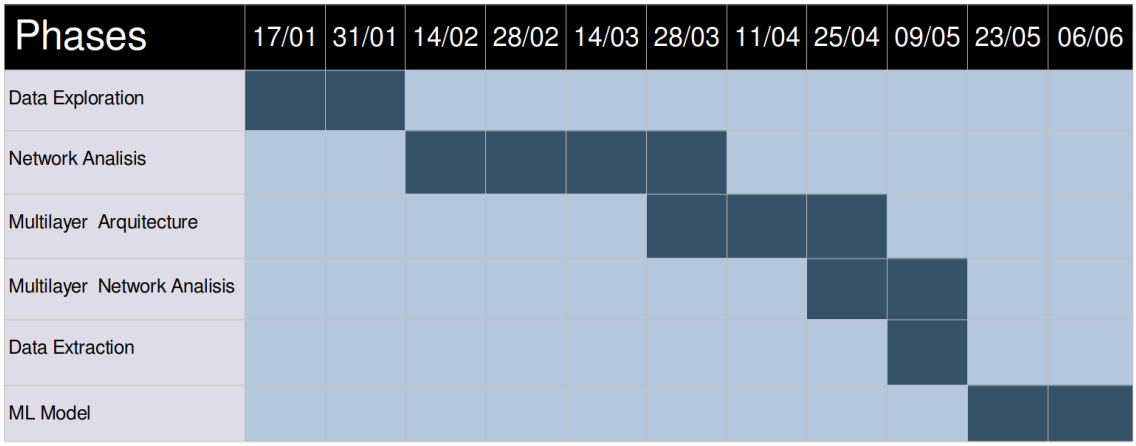


Fig. 1: Grantt diagram

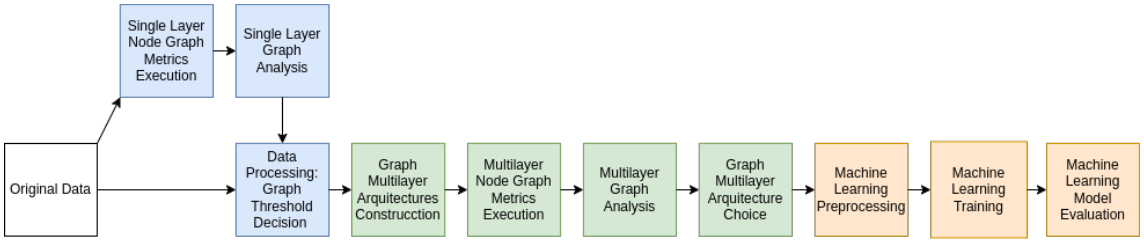


Fig. 2: Data pipeline diagram

## 6 Dataset

The data have been provided by the Hospital Clínic de Barcelona, for the realization of this project. It represents information about 165 people who belong to one of the following groups: the Healthy volunteers (HV), that are the group of volunteers that do not suffer the disease, the Relapsing-remitting multiple sclerosis (RRMS) group, the Primary progressive multiple sclerosis (PPMS) group and the Secondary progressive multiple sclerosis (SPMS) group.

In addition to the group to which they belong, we have the following information for each patient: id, age, sex and the years he/she has been suffering from the disease.

Separately, the brain adjacency matrices of all subjects representing weighted unidirectional graphs, of the measures Fractional anisotropy (FA), Gray Matter (GM) and Resting State (RS) have been provided.

The anatomical parcellation was extracted from the Desikan-Killiany-Tourville (DKT) atlas (Desikan et al., 2006). The nodes of the three brain networks constructed are the 76 brain regions depicted. Thus, the same parcellation is used within each network.

The FA measure is in range [0, 1] and represents the movement of water molecules in the brain. A value close to 0 would mean an isotropic motion of water molecules, and a value close to 1 would mean an anisotropic motion of water molecules. The GM measure has a range of 0 to 1. It indicates the volume correspondence between zones. The RS measure is the correlation between the activation of two areas of the brain and has a range of -1 to 1. The activation of areas of the brain is calculated from the blood flow of the areas of the brain. A value close to 1 represents a strong correlation between two zones, a value close to -1 represents a strong opposite correlation, and a value of 0 would represent a null correlation between two zones. The absolute value have been applied to the matrix.

An age and gender correction has been applied to all matrices.

Regarding the areas of the brain, the following data have been provided: id, name, group, corner and area.

## 7 Data Processing

We proposed to remove the edges that represented a low value in each measurement of the graphs in order to gain sparsity. To have a criterion to know from what value we could discriminate an edge, we executed different measures of nodes by each type of graph and by discriminator value of the edges from 0 to 0.95 in steps of 0.05. The results of each measurement for each node in patients belong to the same MS control group were joined and the four populations were compared with each other in pairs using a student's T-test from which we extracted the p-value to evaluate the statistical significance. In case the p-value obtained was lower than the established threshold, the following data were saved in a list: matrix name, node name, metric name, index of the groups being compared, and p-value obtained. If the p-value was lower than the threshold it indicated that the two populations were significantly different according to that confidence threshold. Through the indexing of the list we could know what discriminator value of the edge it was. In this process three different p-values were applied: 0.05 (95% confidence), 0.01 (99% confidence) and 0.001 (99.9% confidence).

From this data, it was extracted for each configuration of the matrix, discriminant value and p-value, the following information: number of populations with a lower p-value,

| MS type | Number | Age, Years | Females,n(%) | EDSS |
|---|---|---|---|---|
| HV | 18 | 36.62±9.60 | 15(83.33%) | – |
| RRMS | 125 | 45.66±9.48 | 90(72.00%) | 2.11±1.10 |
| SPMS | 16 | 56.82±9.95 | 10(62.50%) | 5.81±0.93 |
| PPMS | 6 | 56.72±3.85 | 4(66.66%) | 5.66±1.08 |

TABLE 1: DATASET RESUME

number of node metrics where more than one pair of populations have a lower p- value, number of node metrics where all population pairs have a lower p-value, the three nodes with more population pairs with a lower p-value, and the metric with more population pairs with a lower p-value.

The most significant datum to decide which discriminator was most appropriate was the number of populations with the p-value below the confidence threshold. The results of the analysis were separated by matrix type in figures (3), (4) and (5). The vertical axis represents the pairs of lower p-values, the horizontal axis the discriminant value, and each line the results with a different p-value.

The results were analyzed with three different p-values (instead of 1) because if the chosen p-value was a large value, when analyzing the results visually with boxplot it was observed that many pairs of populations were not different, despite they had passed the test. For very small p-values very few tests were passed, insufficient to be able to make decisions. The combination of three different p-values allowed us to found a threshold that gave us a certainty that most populations, applying a boxplot, would be different, through the correlation observed visually in the graphics, without having to analyze the boxplots of all populations. The highest p-value, 0.05, was used to find the remarkable threshold values. We selected all the points where the threshold with the highest p-value had more passed student's T-test than the previous and next thresholds. These points were taken into account, as well as their previous and subsequent points. The lower p-value, 0.001, was the one that were used to decide the best threshold, and the middle threshold was used to decide the threshold when the lower thresholds were almost equal.

For the FA matrix, as shown in figure (3), the best discriminant value was 0.45. Threshold 0.20 had more passed student's T-test, but we consider that 0.20 is still a low threshold for the meaning of the measure of the matrix FA. Threshold 0.75 had considerably lower passed student's T-test than threshold 0.45.

For the GM matrix, as shown in figure (4), the best discriminant value is 0.10. Other threshold remarkable point that have a very similar passed student's T-test with p-value=0.001 and p-value=0.01 was the threshold 0.60, but GM matrix had very low edge weights in general terms. With a threshold of 0.60 the matrix would have had very low edges.

For the RS matrix, as shown in figure (5), the best discriminant value was 0.85. Is not a remarkable point, but a previous point of a remarkable point. That point had more passed student's T-test with the lowest p-value than the remarkable point.

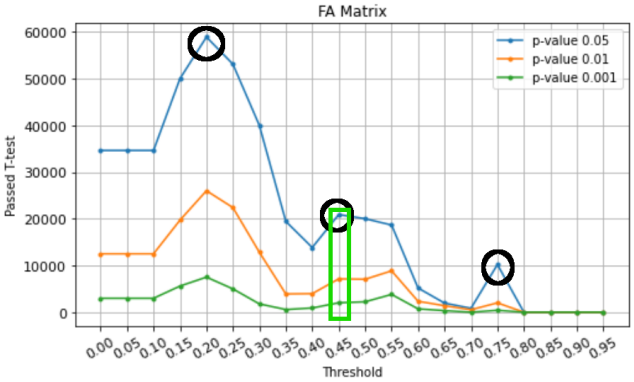


Fig. 3: Comparison of passed student's T-test for different p-values and different thresholds of the matrix FA

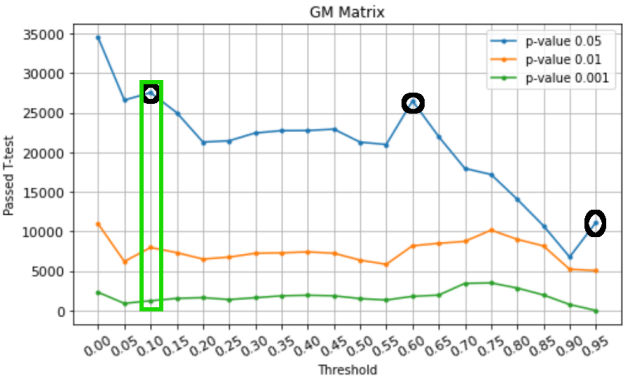


Fig. 4: Comparison of passed student's T-test for different p-values and different thresholds of the matrix GM

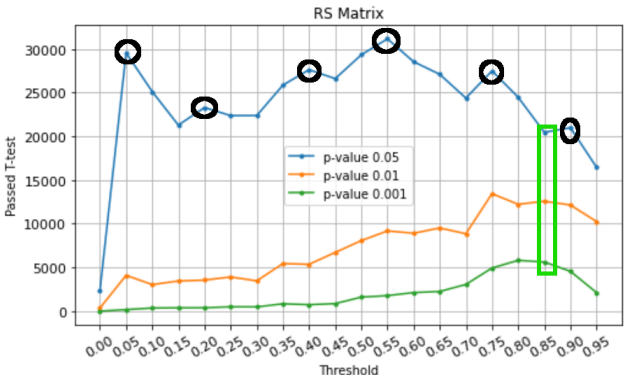


Fig. 5: Comparison of passed student's T-test for different p-values and different thresholds of the matrix RS

## 8 GRAPH MULTILAYER ARCHITECTURE

The objective of this section was to investigate which graph multilayer architecture, where the junction between layers was only from nodes to their self in another layer, was more

suitable for extracting useful node metrics for the classification of different control groups.

Given this constraint we have seven possible architectures:

1. Use only two layers (three possibilities because there are three single layer graphs). An architecture is shown in figure (6).
2. Use three layers and connect all the layers together (one possibility).
3. Use three layers but only one connected to the other two layers (three possibilities). An architecture is shown in figure (7).

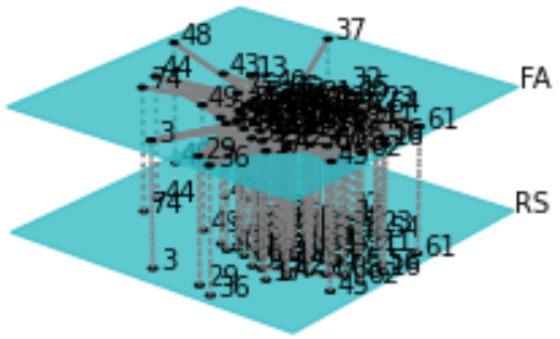


Fig. 6: 2 layer multilayer graph representation by pymnet example.

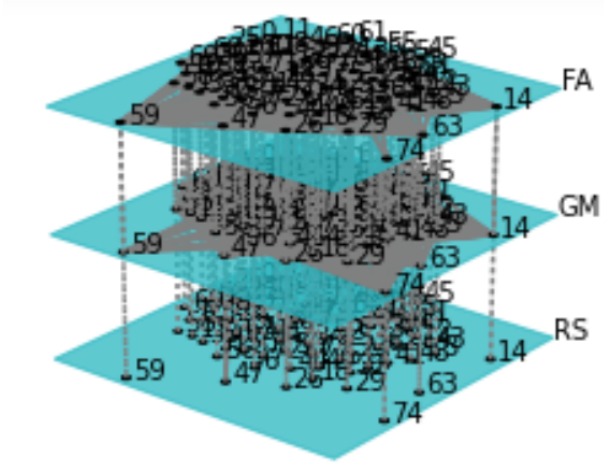


Fig. 7: 3 layer multilayer graph representation by pymnet example.

The seven architectures have been built in Python. For each adjacency matrix the edges between nodes with a lower weight have been removed as stated in section (7. The weight assigned between the nodes of different layers was 0.2. To use the metrics that involucrates the shortest path, the weight of the intralayer edges has been modified by $\frac{1}{weight}$, because in all intralayer graphs, the nodes that represent a greater connection or relationship are represented with a value close to 1 (higher) and nodes with little connection or relationship with a value close to 0 (lower). It would not make sense to use the shortest path with the less correlated/connected nodes.

The metrics that have been used are: degree, strength, closeness centrality, betweenness centrality, clustering and local efficiency.

For each architecture, all metrics were extracted for all nodes in each layer, and a student's T-test of each population separated by control groups was performed in the same way as in section (7), to extract the p-value that indicates to us if two populations were differentiable between them.

To decide which architecture was the best for the feature extraction, we search which architecture obtained the most differentiation between groups, having separation in all groups and especially promoting the separation between HV and patients suffering from the disease (in any of the degrees ) using the p-value threshold of 0.001. The summary of the results presented in table 2.

The architecture that best fits this requirement, by far, was the three-layer architecture, all layers connected to each other, with 1,245 student's T-tests passed, student's T-tests passed by all control groups and many more student's T-tests passed involving the HV than the others student's T-tests: 531 differentiating HV and RRMS, 379 differentiating HV and SPMS, 302 differentiating HV and PPMS, 16 differentiating RRMS and PPMS, 10 differentiating RRMS and SPMS and 10 differentiating SPMS and PPMS.

## 8.1 Code Implementation for Graph Multilayer Architecture

There is currently no complete multilayer graph library on Python. The ones that exist are very simple and precarious. Due to this, almost all the code has had to be implemented for the construction of the graphs and execution of the metrics. However, pymnet [7] was used, a Python library to create graph multilayer architectures.

It was necessary to build a function that passes the nodes and edges to the library one by one from the adjacency matrices, and to implement the node metrics that would be executed. The Dijkstra algorithm was also implemented to calculate the shortest path of all nodes, required for certain metrics. The degree and strength metrics were already implemented, and the closeness centrality, betweenness centrality, clustering, and local efficiency metrics had to be implemented.

Though Dijkstra algorithm is not the search algorithm with the lower time complexity, specifically the complexity of the algorithm is $u(|V^2|)$, being $V$ the number of vertices on the graph, we decided to implement Dijkstra because it was not going to suppose a significant difference in the execution time to use an algorithm with a lower complexity.

## 8.2 Node Metrics

The following metrics were considered in our work to evaluate node's characteristics:

1. **Degree**: Is the number of edges connected to the layer. Intra edges and inter edges.
2. **Strength**: Is the sum of all the weights of the edges connected to the node. Intra edges and inter edges.
3. **Closeness Centrality**: Is a way of detecting nodes that are able to spread information very efficiently through

| Control Group | 2 layers, FA and GM | 2 layers, FA and RS | 2 layers, GM and RS | 3 layers 3 interlayer connections | 3 layers and GM connection in the middle | 3 layers and RS connection in the middle | 3 layers and FA connection in the middle |
|---|---|---|---|---|---|---|---|
| HV RRMS | 197 | 154 | 10 | 531 | 364 | 334 | 385 |
| HV SPMS | 14 | 12 | 4 | 379 | 217 | 178 | 246 |
| HV PPMS | 1 | 4 | 19 | 302 | 15 | 16 | 14 |
| RRMS PPMS | 16 | 0 | 15 | 16 | 19 | 16 | 19 |
| RRMS SPMS | 0 | 9 | 6 | 10 | 9 | 9 | 8 |
| SPMS PPMS | 9 | 0 | 11 | 10 | 13 | 15 | 13 |
| Total | 237 | 179 | 65 | 1,245 | 637 | 568 | 685 |

TABLE 2: STUDENT'S T-TEST PASSED OF ALL 7 MULTILAYER ARCHITECTURES PER PAIR OF GROUPS AND TOTAL

a graph. Measures the farness, or inverse distance, to all the nodes, using the shortest paths.

4. **Betweenness Centrality**: Is a way of detecting the amount of influence a node has over the flow of information in a graph. Measures the number of times the node is used in the shortest path between two nodes.

5. **Clustering**: Takes into account the adjacent nodes that are adjacent between them, too. With this cases is calculated following this formula:

$$\frac{1}{deg(u)(deg(u)-1)} \sum_{vw} (\check{w_{uv}} \check{w_{uw}} \check{w_{vw}})^{\frac{1}{3}} \quad (1)$$

$u$ is the node being measured. $v$ and $w$ are the adjacent nodes. $\check{w}$ symbolizes the weights of the edges.

It is only measured if $deg(u) > 2$.

6. **Local Efficiency**: The local efficiency of a particular vertex is the inverse of the average shortest path connecting all neighbors of that vertex.

## 8.3 Dijkstra Algorithm

Dijkstra is an iterative algorithm capable to find all the shortest path between a given node, the source, and all other nodes in the graph.

The algorithm follows these steps to find the paths:

1. Mark all nodes as unvisited, and with infinite distance, except the source node that would have 0 distance and would be the current node at that point.

2. For the current node, analyse all unvisited nodes by adding the current weight to the edge weight. In case that new weight is lower than the actual shortest path weight in the node, update the shortest path, and the shortest path weight of the node.

3. For all unvisited nodes, select the one with the lower shortest path distance as the current node, and delete the node from unvisited nodes.

4. If there are no more nodes on unvisited list, the algorithm has ended.

5. If there are still nodes on unvisited list, repeat the steps from step 2.

Executing the algorithm for all the nodes, and for all the patients, gives all the shortest paths of all the graphs, necessaries for the metric execution.

# 9 MACHINE LEARNING PROCESS

For this section, the PPMS and SPMS groups are being labeled as the same group because the number of patients in this groups is low, specially on PPMS. Additionally, the differences between this two groups are lower than all the other groups and because the two groups are advanced stages of the disease.

## 9.1 Feature Selection

The data used to train the classifier were the metrics obtained of all the nodes in the three-layer multilayer architecture, all three connected to each other.

There were results of node metrics that stood out more than others, but none that were able to differentiate all control groups, and few that differentiated more than half of group pairs. Consequently, we believed that better results would be obtained if no metrics were excluded.

## 9.2 Normalization

The range of values between features was quite different, because the features came from different graph metrics. To equate the range of all columns and ensure that all features have the same chance of being important for the classifier, the data has been normalized.

Therefore, the ranges for all features have been changed to be in range [0,1]. The same normalization has been applied to both train and test data.

## 9.3 Dimensionality Reduction

There were 1,368 node metric results for each patient. It is a very large dataset to train a machine learning classifier, especially with so few patients.

We wanted to a dimensionality reduction method in order to reduce the number of features per patient to train the classifier. The method used for reduce the dimensolaity was the Principal Component Analysis (PCA). Specifically, the dimensionality has been reduced to eighty features because it has been the number of features with the best results on training data.

The PCA has been trained with train data, and has been applied to both train and test data.

### 9.4 Data Balancing

The data was extremely unbalanced. There was much more data from the RRMS control group than from the other groups. In order to try to reduce the class imbalance in training, in addition to taking advantage of the technique to increase the data, the SMOTE oversampling technique was used.

SMOTE oversampling is a technique that uses k-nearest neighbour algorithm to create synthetic data by choosing random data from the minority class (at the moment). It stops when all the classes have the same number of cases, and at least one class have 100% real data.

Finally, we were left with a total of 279 patients within the train set (train + validation set), divided into three groups, with 93 patients per group.

The oversampling technique was not applied to the test set.

### 9.5 Machine Learning Algorithms

Many machine learning algorithms have been evaluated to classify the data:

1. Support Vector Machine: This algorithm find hyperplanes that separates the classes. Then it finds the nearest points to the hyperplane, that are called support vectors, and it calculates the distance between the hyperplane and the support vectors, that is called margin. The algorithm chose the hyperparameter that maximizes margin.

2. Logistic Regression Classifier: It the most popular machine learning classifiers for binary data classification. It adapt a logistic function to the train data, and classify the data according to a threshold in the function.

3. K Neighbors Classifier: It Classifies the targets according to the voting of the nearest ”k” points in the training data.

4. Decision Tree Classifier: It is an algorithm that classifies data based on how a previous set of questions were answered. The boundaries of the algorithm are decided based on the minimization of entropy.

5. Gradient Boosting Classifier: GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage ”n“ classes regression trees are fit on the negative gradient of the loss function, e.g. binary or multi-class log loss.

6. Extreme Gradient Boosting Classifier (XGB Classifier): Uses the same algorithm as the Gradient Boosting Classifier, but it is implemented in an efficient and highly effective way than the others Gradient Boosting Classifiers.

The best classifiers were Support Vector Machine and Logistic Regression Classifier. Specifically, the best results were with Logistic Regression Classifier.

The other classifiers achieve lower accuracy values. In some executions, these models have achieved good accuracy values with train and validation data, but not with training data. We realised that the models were suffering from a lot of overfitting, caused in part from the oversampling technique that created very similar data than the existence data

Precisely the only two algorithms that have worked a little well of all the trained ones, Supported Vector Machine and Logistic Regressor Classifier as mentioned before, have been two algorithms that are not characterized by being too capable of suffering a lot of overfitting with the data with which the model is trained.

All the results shown and discussed in the subsection (9.6) and sections (10) and (11), have been made through a Logistic Regressor Classifier.

### 9.6 Training and Evaluation

The train and evaluation set consisted of 75% of the total of the real data. The cross validation technique was used to train the best model, with 4 folds. The train data then consisted of 75% of the set, and the validation of 25%. It is not controlled how much real and synthetic data (oversampling data) was used in the folds.

To measure the results of the models on the validation fold, it has been considered convenient to use the Precision, Recall and F1-Score measures both by class and in the total set, as well as the accuracy as a total data set. For the global Precision, Recall and F1-score, is used the Micro Average Method and Macro Average method. Micro Average Metric calculates the metric using the total number of true positives, false positives and/or false negatives, while Macro Average Method makes the mean of the metric in all classes, regardless the class weight.

As it was possible that the model suffered from overfitting, and that we found that the data of the validation set were very similar to some of the data of the training, due to the oversampling, a single model of cross-validation was not chosen. We chose all models that exceeded 0.85 in precision, recall, F1-score and accuracy. There were three models that exceeded that threshold. The result are shown in tables 3, 4, 5 and 6.

These three models were then scored using the test data, and the one that best ranked the independent data set was chosen. The test results are shown in section 10.

Precisely the model that best ranked the test set was the model chosen in the previous step with the worst metric results. It has probably been the model most able to generalize the cases, while the other two models have ”memorized” too much the data they have used to train.

## 10 Results

### 10.1 Classification Model Results on Test Data

Various measures have been used to measure the results of the classifier in test data, basically because it is unbalanced data, as it did not make sense to apply oversampling in the test set, nor to delete data, as there was little, and when it

| Classes | Precision | Recall | F1-Score | nº Cases |
|---|---|---|---|---|
| HV | 0.92 | 1.00 | 0.96 | 23 |
| RRMS | 0.67 | 0.80 | 1.00 | 24 |
| PPMS + SPMS | 0.79 | 1.00 | 0.88 | 23 |
| Accuracy | | | 0.89 | 70 |
| Micro Average | 0.90 | 0.89 | 0.88 | 70 |
| Macro Average | 0.91 | 0.89 | 0.89 | 70 |

TABLE 3: CLASSIFICATION REPORT ON TRAIN DATA, FIRST FOLD DIVISION

| Classes | Precision | Recall | F1-Score | nº Cases |
|---|---|---|---|---|
| HV | 0.81 | 1.00 | 0.90 | 22 |
| RRMS | 1.00 | 0.63 | 0.78 | 30 |
| PPMS + SPMS | 0.75 | 1.00 | 0.86 | 18 |
| Accuracy | | | 0.84 | 70 |
| Micro Average | 0.85 | 0.85 | 0.84 | 70 |
| Macro Average | 0.84 | 0.90 | 0.85 | 70 |

TABLE 4: CLASSIFICATION REPORT ON TRAIN DATA, SECOND FOLD DIVISION

| Classes | Precision | Recall | F1-Score | nº Cases |
|---|---|---|---|---|
| HV | 0.93 | 1.00 | 0.96 | 25 |
| RRMS | 1.00 | 0.88 | 0.93 | 16 |
| PPMS + SPMS | 1.00 | 1.00 | 1.00 | 29 |
| Accuracy | | | 0.97 | 70 |
| Micro Average | 0.98 | 0.96 | 0.96 | 70 |
| Macro Average | 0.97 | 0.97 | 0.97 | 70 |

TABLE 5: CLASSIFICATION REPORT ON TRAIN DATA, THIRD FOLD DIVISION

| Classes | Precision | Recall | F1-Score | nº Cases |
|---|---|---|---|---|
| HV | 0.92 | 1.00 | 0.96 | 23 |
| RRMS | 1.00 | 0.78 | 0.88 | 23 |
| PPMS + SPMS | 0.88 | 1.00 | 0.94 | 23 |
| Accuracy | | | 0.93 | 69 |
| Micro Average | 0.93 | 0.93 | 0.93 | 69 |
| Macro Average | 0.93 | 0.94 | 0.93 | 69 |

TABLE 6: CLASSIFICATION REPORT ON TRAIN DATA, FORTH FOLD DIVISION

comes to unbalanced data it is more complicated to measure the performance of a classifier model with just a single metric.

Precision, Recall, and F1-Score measurements were used for both class and total set, as well as accuracy as total data set, as with validation data. The results are in table 7.

This time we also considered convenient to attach the results in the form of a confusion matrix (see figure 8), as the test set does not consist of too much data or too many classes, and so we can clearly see in which cases the model fails more, less, or never.

| Classes | Precision | Recall | F1-Score | nº Cases |
|---|---|---|---|---|
| HV | 0.75 | 0.75 | 0.75 | 4 |
| RRMS | 0.91 | 0.88 | 0.89 | 33 |
| PPMS + SPMS | 0.5 | 0.6 | 0.55 | 5 |
| Accuracy | | | 0.83 | 42 |
| Micro Average | 0.72 | 0.74 | 0.73 | 42 |
| Macro Average | 0.84 | 0.83 | 0.84 | 42 |

TABLE 7: CLASSIFICATION REPORT ON TEST DATA


Fig. 8: Confusion Matrix of the test set of the Logistic Regressor Classifier trained model.

## 10.2 Key Nodes for MS Distinction

During the analysis of graph multilayer architectures, it has been observed that there are certain nodes that their metric results are more likely to pass the student's T-tests than others. Similarly, there are nodes that in no architecture tend to pass student's T-test. The following nodes would be highlighted:

1. Right lateralorbitofrontal. Results in table 8. Moreover, have the top 1 node per metric in a concrete layer with more passed student's T-tests, as shown in tables 9, 10 and 11 and figures 9 and 10.

| Architecture | Passed student's T-test | Ranking |
|---|---|---|
| 2 Layers, FA and GM | 11 | 1 |
| 2 Layers, FA and RS | 2 | 37 |
| 2 Layers, GM and RS | 8 | 1 |
| 3 Layers, 3 Connections | 29 | 1 |
| 3 Layers, FA in the middle | 16 | 1 |
| 3 Layers, GM in the middle | 16 | 1 |
| 3 Layers, RS in the middle | 14 | 1 |

TABLE 8: RIGHT LATERALORBITOFRONTAL NODE. PASSED STUDENT'S T-TEST PER ARCHITECTURE OUT OF 108 (P-VALUE=0.001) AND RANKING COMPARED THE OTHER 76 NODES

2. Right amygdala. Results in table 12.
3. Left amygdala. Results in table 13.
4. Right putamen. Results in table14.

| Layer | Metric | Passed student's T-test |
|---|---|---|
| GM | Local Efficiency | 5 |
| GM | Closeness Centrality | 5 |

TABLE 9: RIGHT LATERALORBITOFRONTAL NODE. MAXIMUM PASSED STUDENT'S T-TEST PER METRIC IN A CERTAIN LAYER (OUT OF 6). ARCHITECTURE OF 3 LAYERS AND 3 INTER CONNECTIONS.

| Group | P-value |
|---|---|
| RRMS SPMS | 0.599303 |

TABLE 10: RIGHT LATERALORBITOFRONTAL NODE. P-VALUE OF NO PASSED STUDENT'S T-TESTS ON GM LAYER AND LOCAL EFFICIENCY METRIC.

| Group | P-value |
|---|---|
| RRMS SPMS | 0.4506 |

TABLE 11: RIGHT LATERALORBITOFRONTAL NODE. P-VALUE OF NO PASSED STUDENT'S T-TESTS ON GM LAYER AND CLOSENESS CENTRALITY METRIC.

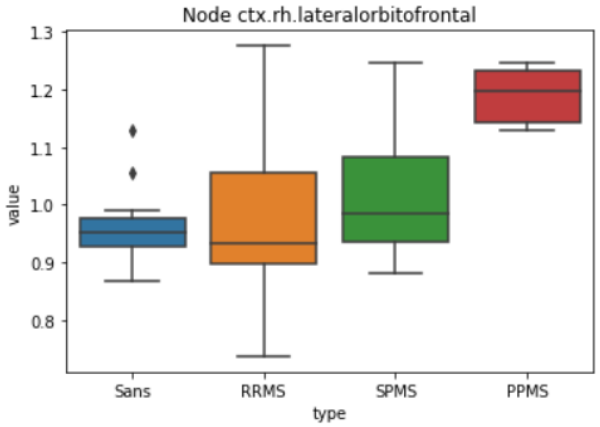


Fig. 9: Right lateralorbitofrontal node. Boxplots comparison of metric results on layer GM using Local efficiency.

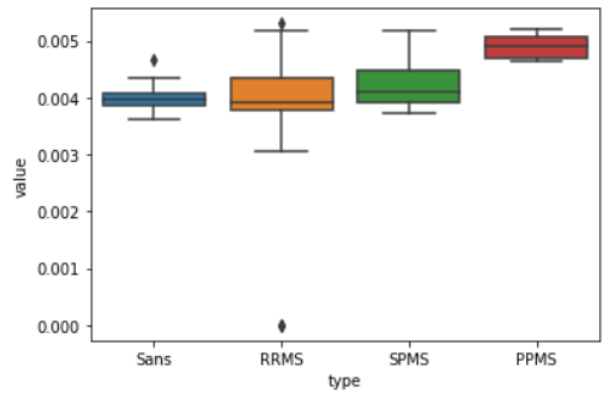


Fig. 10: Right lateralorbitofrontal node. Boxplots comparison of metric results on layer GM using Closeness centrality.

## 11 DISCUSSION AND CONCLUSIONS

### 11.1 Classification Model Discussion and Conclusions

We can see in the results table 7 that the class with the most problems with classification was the class of groups in the advanced stage of the disease, followed by the HV group. The group with the best results was the control group of the initial phase of the disease. One of the main reasons for this is that the train set contained more real data from the patients control group in the early stages of the disease than from the other groups. We can see that the errors come from both false positives and false negatives in a similar proportion in each class.

A fact that can be observed in the confusion matrix (see figure 8) is that in no case has a case of a healthy person been classified as a patient in an advanced stage of the disease, or vice versa. This must be due, apart from the unbalance of data in favor of the initial cases of the disease, to the fact that these two classes are much more differentiable between them than between the group of the initial state of the disease.

The results of the classification model have not been as expected. They are around values similar to or lower than

| Architecture | Passed T-test | Ranking |
|---|---|---|
| 2 Layers, FA and GM | 7 | 3 |
| 2 Layers, FA and RS | 3 | 24 |
| 2 Layers, GM and RS | 5 | 5 |
| 3 Layers, 3 Connections | 17 | 28 |
| 3 Layers, FA in the middle | 14 | 3 |
| 3 Layers, GM in the middle | 15 | 1 |
| 3 Layers, RS in the middle | 14 | 1 |

TABLE 12: RIGHT AMYGDALA NODE. PASSED STUDENT'S T-TEST PER ARCHITECTURE OUT OF 108 (P-VALUE=0.001) AND RANKING COMPARED THE OTHER 76 NODES

| Architecture | Passed T-test | Ranking |
|---|---|---|
| 2 Layers, FA and GM | 6 | 4 |
| 2 Layers, FA and RS | 4 | 13 |
| 2 Layers, GM and RS | 7 | 3 |
| 3 Layers, 3 Connections | 18 | 10 |
| 3 Layers, FA in the middle | 15 | 2 |
| 3 Layers, GM in the middle | 15 | 3 |
| 3 Layers, RS in the middle | 13 | 3 |

TABLE 13: LEFT AMYGDALA NODE. PASSED STUDENT'S T-TEST PER ARCHITECTURE OUT OF 108 (P-VALUE=0.001) AND RANKING COMPARED THE OTHER 76 NODES

| Architecture | Passed T-test | Ranking |
|---|---|---|
| 2 Layers, FA and GM | 4 | 9 |
| 2 Layers, FA and RS | 7 | 1 |
| 2 Layers, GM and RS | 2 | 9 |
| 3 Layers, 3 Connections | 20 | 4 |
| 3 Layers, FA in the middle | 14 | 3 |
| 3 Layers, GM in the middle | 13 | 4 |
| 3 Layers, RS in the middle | 11 | 6 |

TABLE 14: RIGHT PUTAMEN NODE. PASSED STUDENT'S T-TEST PER ARCHITECTURE OUT OF 108 (P-VALUE=0.001) AND RANKING COMPARED THE OTHER 76 NODES

other MS control group classifier models, which are based on other features. Outstanding results were expected compared to existing classifiers due to the use of multilayer graph analysis for feature construction. However, the results are not considered a failure either.

## 11.2 Key Nodes Conclusion

1. Right lateralorbitofrontal: It is the node that has been most able to create differentiable populations among patients in control groups by applying node metrics to it. The node is in the top 1 of nodes with more student's T-tests passed in 6 of the 7 multilayer architectures created (in some cases tied with others). In addition, with the architecture of 3 layers and 3 inter connections, the architecture that has been chosen to extract the metrics by the classifier model, has been able to differentiate 5 of 6 pairs of populations with 2 metrics in a specific layer. Exactly in the GM layer and with the metrics of local efficiency and closeness centrality. In both cases, the pair that could not be considered differentiable was RRMS with SPMS. We can hypothesize that the changes produced in this part of the brain from the point of view of multilayer graph analysis, of a patient who has evolved into SPMS and one who is in RRMS are not very different, but it does change a lot with respect to the other couples.

2. Right amygdala:.It is a node that is in the top 3 nodes that have passed more student's T-test in 4 of the 7 multilayer graph architectures. In the case of the architecture chosen for the construction of the classifier model, it is in the ranking 28.

3. Left amygdala:It is a node that is in the top 3 nodes that have passed more student's T-test in 4 of the 7 multilayer graph architectures. In the case of the architecture chosen for the construction of the classifier model, it is in the ranking 10.

4. Right putamen: It is a node that is in the top 4 nodes that have passed more student's T-test in 4 of the 7 multilayer graph architectures. In the case of the architecture chosen for the construction of the classifier model, it is in ranking 4.

## REFERENCES

[1] Eshaghi, A., Young, A.L., Wijeratne, P.A. et al. Identifying multiple sclerosis subtypes using unsupervised machine learning and MRI data. Nat Commun 12, 2078 (2021). https://doi.org/10.1038/s41467-021-22265-2

[2] E. Solana, E. Martinez-Heras, J. Casas-Roma, L. Calvet, E. Lopez-Soley, M. Sepulveda, N. Sola-Valls, C. Montejo, Y. Blanco, I. Pulido-Valdeolivas, M. Andorra, A. Saiz, F. Prados, S. Llufriu. (2019). Modified connectivity of vulnerable brain nodes in multiple sclerosis, their impact on cognition and their discriminative value. Scientific Reports 9, 20172. https://doi.org/10.1038/s41598-019-56806-z

[3] Comparing multilayer brain networks between groups: Introducing graph metrics and recommendations https://doi.org/10.1016/j.neuroimage.2017.11.016

[4] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A. Porter, Sergio Gómez, and Alex Arenas. Mathematical Formulation of Multilayer Networks. Phys. Rev. X 3, 041022 (2013). 10.1103/PhysRevX.3.041022

[5] Manlio De Domenico, Multilayer modeling and analysis of human brain networks, GigaScience, Volume 6, Issue 5, May 2017, gix004, https://doi.org/10.1093/gigascience/gix004

[6] De Domenico, M., Solé-Ribalta, A., Omodei, E. et al. Ranking in interconnected multilayer networks reveals versatile nodes. Nat Commun 6, 6868 (2015). https://doi.org/10.1038/ncomms7868

[7] Multilayer Networks Library for Python (Pymnet), http://www.mkivela.com/pymnet/

[8] Hu, W., Combden, O., Jiang, X. et al. Machine learning classification of multiple sclerosis patients based on raw data from an instrumented walkway. BioMed Eng OnLine 21, 21 (2022). https://doi.org/10.1186/s12938-022-00992-x

[9] Relapsing-remitting MS (RRMS) https://www.nationalmssociety.org/What-is-MS/Types-of-MS/Relapsing-remitting-MS

[10] Primary progressive MS (PPMS) https://www.nationalmssociety.org/What-is-MS/Types-of-MS/Primary-progressive-MS

[11] Secondary progressive MS (SPMS) https://www.nationalmssociety.org/What-is-MS/Types-of-MS/Secondary-progressive-MS

[12] De Domenico, Manlio and Solé-Ribalta, Albert and Cozzo, Emanuele and Kivelä, Mikko and Moreno, Yamir and Porter, Mason A. and Gómez, Sergio and Arenas, Alex. 10.1103/PhysRevX.3.041022. Mathematical Formulation of Multilayer Networks.

[13] De Domenico, M., Solé-Ribalta, A., Omodei, E. et al. Ranking in interconnected multilayer networks reveals versatile nodes. Nat Commun 6, 6868 (2015). https://doi.org/10.1038/ncomms7868