MULTI MODAL APPROACH FOR MULTIPLE SCLEROSIS CLASSIFICATION USING GRAPH NEURAL NETWORKS

Joan Samper Vila

June 24, 2025

Abstract

The study of Multiple Sclerosis (MS) using graph-based representations of the brain presents a promising opportunity for advancing its diagnosis, treatment, and prevention. In this paper, we propose a graph analysis approach for classifying people with MS (pwMS) and healthy volunteers(HV) based on brain imaging data, results with the potential of diagnosing and preventing MS in its early stages. Our method uses an integration of three different brain imaging modalities: Diffusion Tensor Imaging (DTI),Resting State Functional Magnetic Resonance Imaging (rs-fMRI), and Gray Matter (GM) data. To address challenges such as class imbalance and limited data availability, we employ advanced data augmentation strategies including graph mixing and SMOTE. We evaluate traditional machine learning baselines alongside GCN and GAT architectures, demonstrating that the multi-modal GCN significantly outperforms other models, particularly in binary classification between MS patients and healthy controls. Additionally, we provide interpretability analyses to identify key brain regions and topological features relevant to disease progression. Our findings highlight the potential of GNN-based multi-modal approaches for improving MS diagnosis, contributing to the development of explainable AI in clinical neuroscience.

Keywords: Multi-modal, Multiple Sclerosis, Graph Neural Network, Graph Convolutional Network, Graph Attention Network

1 INTRODUCTION

The brain is an intricate and highly interconnected system, making its study and interpretation a challenging task. This complexity has led to the development of multiple brain imaging techniques. The three most commonly used methods, which also serve as the data sources for this study, are Diffusion Tensor Imaging (DTI), Resting State Functional Magnetic Resonance Imaging (rs-fMRI), and Gray Matter (GM) analysis.

Each imaging technique is associated with a specific atlas that segments the brain into Regions of Interest (ROIs), facilitating its interpretation. In our study the brain atlas used divides the brain data into 76 ROIs [1] . A widely used technique, also used in this study, is to preprocess the data to leverage these ROIs, treating them as nodes in a graph representation of the brain. The connections between different regions are represented by weighted edges, where the weights correspond to the correlation between ROIs, reflecting their functional or structural connectivity, getting as a result a dense fully connected weighted graph, represented as a 76x76 matrix, as in [2].

• Contact E-mail: Joan.Samper@autonoma.cat
• Supervised by: Jordi Casas Roma (Computer Science)
• Academic Year 2024/25

To enhance the effectiveness of graph processing, we apply edge pruning, removing weak connections that contribute minimal information. This step reduces noise, improves computational efficiency, and ensures that significant local connections are preserved, rather than being vanished by an average over all values. By refining the graph representation in this way, we aim to improve the accuracy and interpretability of brain connectivity.

Medical data is often protected due to privacy regulations, making it challenging to compile large datasets that combine individuals from multiple institutions. Our study is no exception, relying on a dataset that includes 71 healthy volunteers as controls and 270 subjects with MS, these last group divided into different stages of the disease. A relatively large dataset in the medical world, but limited when it comes to develop deep learning techniques.

To overcome the problem of a small dataset, data augmentation helps to generate synthetic samples, inspired in the method proposed in [3], together with a graph mixing approach explained in detail in further sections of this work.

We apply graph metrics computation to have a richer and purely numerical representation with statistical values of every graph [2] [4]. So, every datapoint can be represented with its raw graph representation, graph metrics, or a combination of both.

This study aims to classify brain graphs into healthy volunteers(HV) and people with MS(pwMS) and further cate-

gorize this last group according to 3 different disease stages.

Our approach employs various machine learning approaches, including Support Vector Machines (SVM), and Decision Trees. These models use graph adjacency matrices as input and serve as relatively simple yet well-established baselines for comparison against more advanced deep learning techniques. Although previous studies could serve as references, many do not provide open datasets, making direct evaluation of our data a more reliable approach.

Our primary focus is the development of a Graph Convolutional Network(GCN)[5] for classification. This model processes raw brain graphs, or brain graphs together with its metrics as input and generates graph embeddings, which are then used for classification. The main advantage of GCNs is that they leverage message passing, where information is propagated across connected nodes at each layer, capturing both local and global graph structures. This process results in dense graph embeddings, which can be utilized in multiple ways.

The most straightforward approach and the one implemented in this work is to pass these embeddings through a final MLP layer for classification.

2 Objectives

1. **Integrate DTI, rs-fMRI and GM into a unique Graph Representation:** Find an efficient, and convenient way to merge the three representations of the brain into a single graph in order to make an efficient training of the neural networks.
2. **Develop a GNN for classification:** Perform multiple experiments with GNN's with end to end classification to test their effectiveness in this task.
3. **Improve classification of the patient disease stage:** Achieve good performance in the multiple-class classification task, distinguishing different stages of MS, not only focusing on the binary classification to detect whether or not a patient has MS.
4. **Document all the process in a scientific paper:** Provide a complete and reproducible account of the methodology and findings in the form of a scientific paper.

3 Methodology and Work Plan

3.1 Methodology

The methodology for this study follows the CRISP-DM (Cross Industry Standard Process for Data Mining) framework described in [6], a structured approach for data analysis and machine learning projects. The pipeline consists of the following stages:

1. **Problem Definition**
   - **Read state of the art literature:** Read the main literature of this topic in order to know the hot topics and the research gaps to possibly fill.
   - **Define problem and objectives:** Clearly define a problem, and all the objectives required to solve it.
2. **Data exploration:** Explore the data, mainly by computing metrics, in order to know which type of research and techniques can be more useful.
3. **Data pre-processing**
   - **Graph preprocessing:** Generate graphs with the given data. Also compute the corresponding features of each node of the graphs in order to train GNNs.
   - **Data augmentation:** As the dataset is unbalanced and has few datapoints a data augmentation approach will be followed. It will generate new data using generative techniques and data mixing.
4. **Modeling**
   - **Machine learning baselines:** Train SVM and decision tree models in order to have a baseline of metrics with which we can compare posterior results with more advanced techniques like GNN's.
   - **Graph Neural Network construction:** Build the GNN models, by trying multiple configurations and methods. The main ones will be GCN and GAT.
5. **Model Evaluation**
   - **Metrics Extraction:** Compute the metrics for all the models and perform an ablation study.
   - **Interpretability analysis:** Interpret the results and find patterns in the classification that can lead to explainable results.
6. **Documentation of the research:** Provide a complete and reproducible account of the methodology and findings in the form of a scientific paper.

3.2 Work Plan

The work plan proposed for the project is summarized in the following graph, containing all the tasks and the time frames dedicated to each of them in the form of a gantt chart in 1.

4 State of the art

The study of neurological disorders, particularly Multiple Sclerosis (MS), is a widely explored topic in the field of data science and medicine, driven by advancements in neuroimaging techniques and sophisticated computational methods [7][8][9]. Magnetic Resonance Imaging (MRI) [10], together with data analysis [11][12], has emerged as a fundamental clinical tool for the diagnosis of MS.

A central aspect of neuroimaging data analysis is the construction of brain networks based on graph theory [13], which has seen significant advancements in recent years. This approach has proven to be a powerful framework for understanding the complex organization of the brain and its alterations in neurological conditions [14][12]. In this approach, brain regions are represented as nodes and the

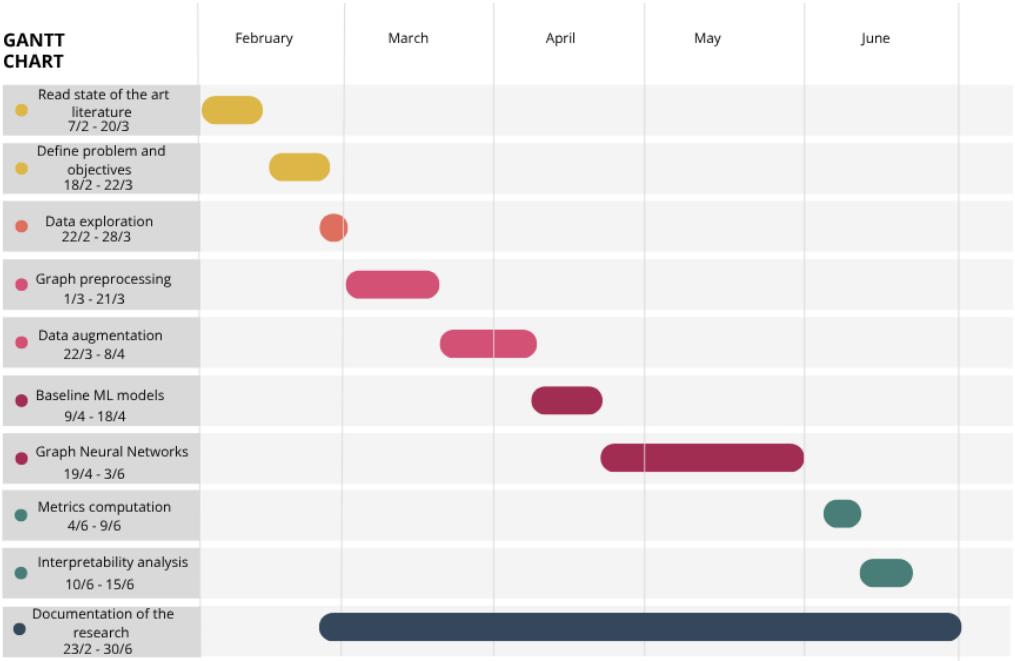

Fig. 1: Gantt chart describing the work plan.

connections between them as edges. These networks can be derived from different modalities like functional MRI (rs-fMRI), diffusion MRI (DTI), and structural MRI (GM), capturing different aspects of brain connectivity.

The integration of computational methods, graph processing techniques and deep learning models [15][7] allowed the development of graph neural networks (GNNs) [5]. These models greatly improved the analysis of neuroimaging data in the context of MS and other brain disorders. Additionally, widely known deep learning models like Convolutional Neural Networks (CNNs) have demonstrated superiority over traditional artificial intelligence methods for MS classification based on MRI data, particularly when combined with techniques like batch normalization and dropout to overcome issues like overfitting [8].

As deep learning models complexity grow, data augmentation techniques have become essential for addressing the limited data availability of clinical datasets. These techniques help enhance model robustness by generating additional training samples from existing data. Common strategies are to sample subsets of a subject data to get multiple datapoints from the same subject, or applying transformations to a datapoint, such as noise injection, rotations, and cropping [8][15].

In recent years, the field has been shifting towards multimodal and multilayer network analysis [2][13], which integrates information from different neuroimaging modalities to provide a more comprehensive understanding of brain changes due to MS. This approach has shown potential in identifying brain regions with synchronized connectivity deterioration in MS [12].

There is a growing focus on interpreting deep learning results in neuroimaging, particularly in identifying brain regions that contribute to specific predictive outcomes [2][12]. This is especially relevant in multiple sclerosis (MS), where deep learning techniques can help in highlighting the areas most affected by the disease. The goal is to enhance diagnostic accuracy by identifying reliable biomarkers that indicate disease progression, facilitating a deeper understanding of MS through network analysis [11].

5 Dataset

The dataset used in this paper is obtained by merging data from the Hospital Clínic de Barcelona and an italian hospital. The dataset comprises a total of 270 subjects: 165 subjects from the Hospital Clínic (dataset1) and 105 subjects from the italian hospital (dataset2). The dataset contains healthy volunteers (HV) and people with multiple sclerosis (pwMS). The pwMS group is further categorized into three clinical subtypes, describing different stages of the disease. The first subset of samples is composed of people with Relapsing-Remitting Multiple Sclerosis (RRMS), characterized by clearly defined attacks of worsening neurological function followed by periods of partial or complete recovery. The second subset includes people with Secondary Progressive Multiple Sclerosis (SPMS), which initially begins as RRMS and later transitions into a phase of steady disease progression. The third subset contains subjects with Primary Progressive Multiple Sclerosis (PPMS), marked by a gradual worsening of symptoms without early relapses or remissions. The dataset source and the amount of samples of each group is depicted in 1

Category	dataset1	dataset2	TOTAL
HV	18	53	71
RRMS	125	30	155
SPMS	16	15	31
PPMS	6	7	13
pwMS	147	52	199
TOTAL	165	105	**270**

TABLE 1: Distribution of subjects by source and MS phenotype. HV: Healthy Volunteers; RRMS: Relapsing-Remitting MS; SPMS: Secondary Progressive MS; PPMS: Primary Progressive MS; pwMS: People with MS.

For each subject in the dataset, three different brain scans

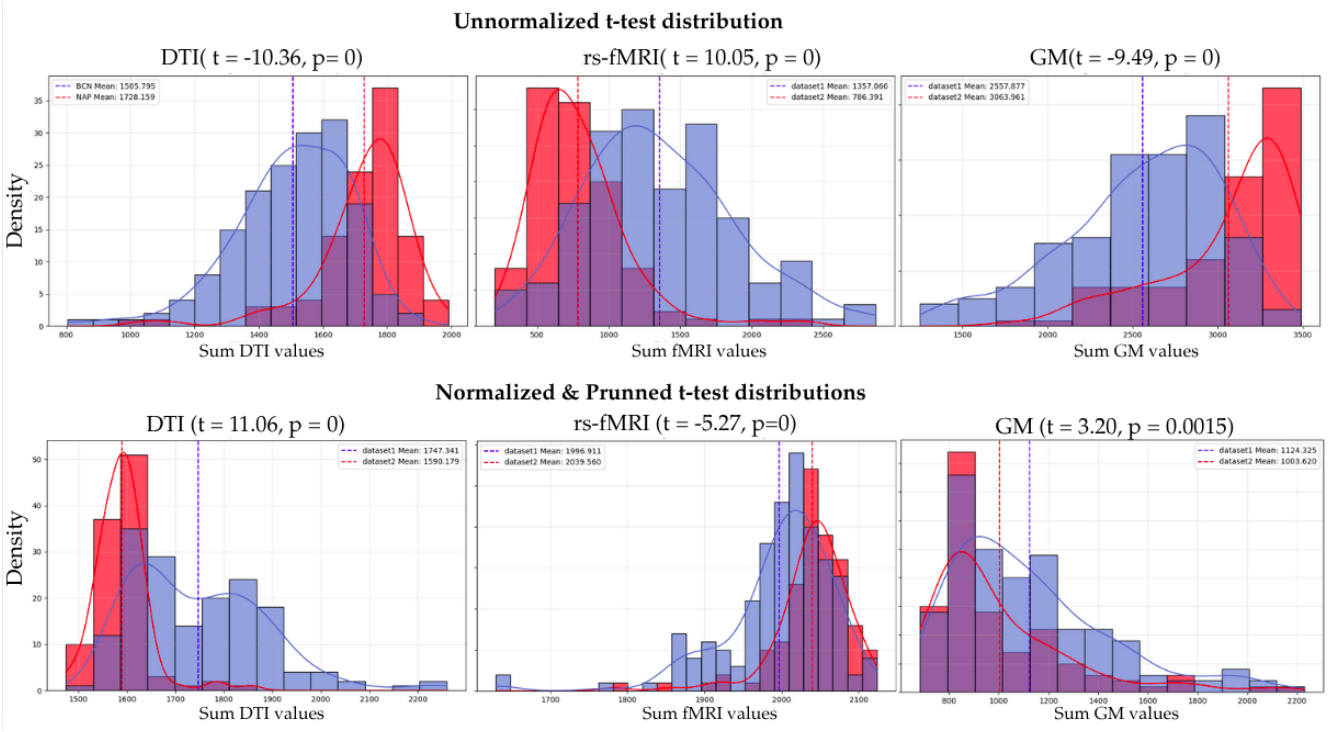

Fig. 2: t-testing is performed to compare the data distributions of the different datasets. Blue lines represent dataset 1 and red lines represent dataset 2. The first row of the figure shows the distributions of the raw (unnormalized) data, while the second row displays the same distributions after normalization and thresholding. For visualization purposes, each data matrix was reduced to a single scalar value by summing all its elements. These scalar values are then used to construct the distributions shown in the plots and perform t-testing.

are provided. This 3 scans are the Diffusion Tensor Imaging (DTI) of the brain, a resting state functional magnetic resonance imaging(rs-fMRI) sample, and a gray matter(GM) sample. Each of the 3 different scans are already processed as regions of the brain, following the atlas described in [1] . The atlas divide the brain in 76 different regions, which are then processed to find an adjacency matrix of dimension 76x76, which represent the connection between regions in the brain by showing correlation of those regions [2]. Additional to the adjacency matrices, we have the volume of each region of the brain for each subject in the dataset, which is the same through the 3 different scans.

5.1 Dataset t-test

A statistical t-test was conducted to compare the distributions of dataset 1 and dataset 2, as illustrated in Figure 2. The results clearly indicate that the two datasets originate from significantly different distributions. This conclusion is supported by the dominance of low p-values < 0.05, confirming that the observed differences are statistically significant. The statistical difference can also be assessed by visually looking at the displacement between distributions. Notably, this distinction remains even after normalization and thresholding, indicating a fundamental divergence in the underlying data distributions. Such distributional differences pose a significant challenge for the design and implementation of a classification model, as the model is likely to perform well on data from one distribution while generalizing poorly to the other. This difference in data distribution indicates that models trained on data from a specific source may not generalize well to data from other sources, even when representing the same type of brain scans, due to substantial variations in underlying distributions.

6 Pipeline

The pipeline depicted in figure 3, explain the whole processing to do the experiments presented in this work. The source code implementing the proposed methodology is publicly available at `https://github.com/Kerasam12/TFG_MS_classification`. However, data used in the study is not publicly available due to sensitivity concerns, thought could be available from the corresponding author upon reasonable request.

Pipeline:

1. **Load brain graph matrices** derived from the three types of neuroimaging data: DTI, rs-fMRI and GM.
2. **Graph matrices are normalized and pruned** to remove noise and redundant connections.
3. **Data augmentation** using either SMOTE or graph mixing.
4. The upper branch of the pipeline outlines the procedure for training the baseline models. First, a **joint multi-matrix representation** is constructed by combining the individual connectivity matrices from the three modalities. This joint representation is then flattened into a feature vector, which serves as input for the baseline models.
5. The lower branch details the pipeline for training GNN models. Initially, each **adjacency matrix is converted into a graph**.
6. **Graphs are interconnected** using DTI and rs-fMRI data to define the inter-graph links.
7. **Features are extracted** for single-graph data and for multi-graph data. This generated data is used separately to train different models.

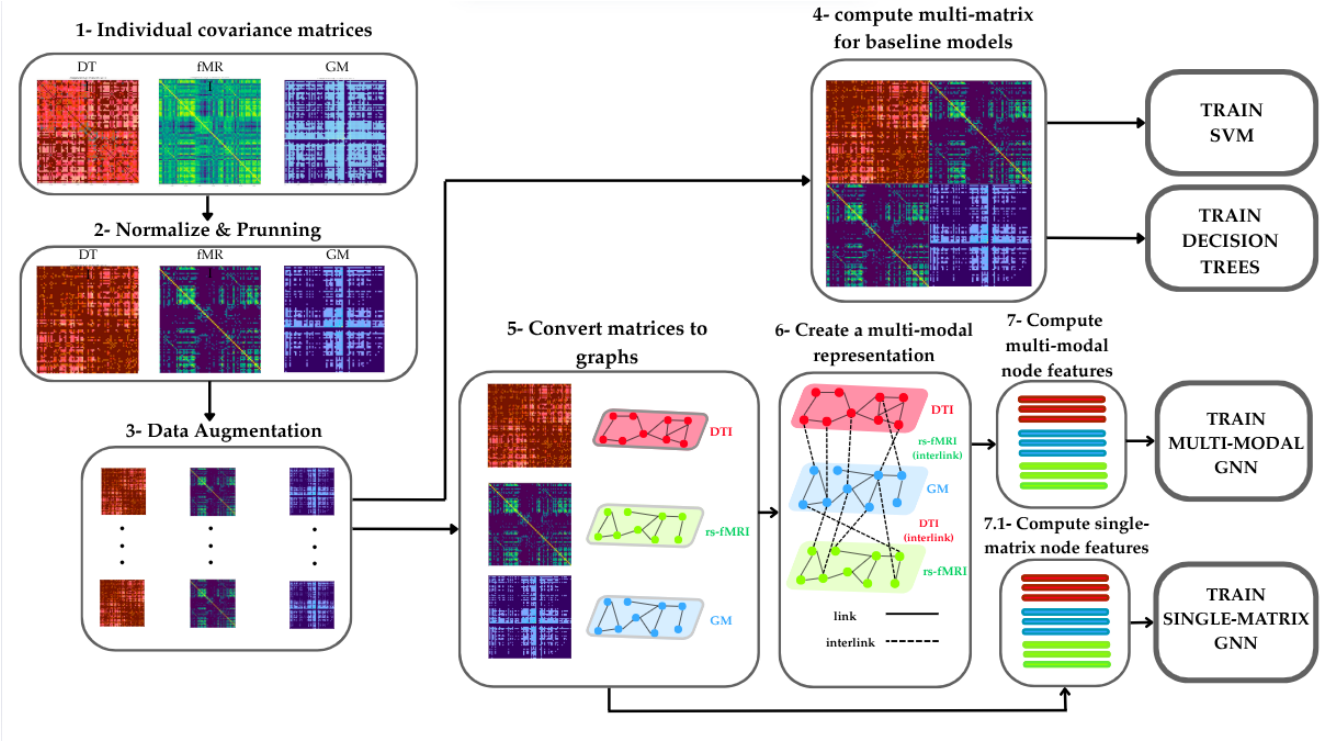

Fig. 3: Schematic illustration of the pipeline used to train the Baseline and GCN models.

7 Data Preprocessing

7.1 Graph Pruning

In our dataset the adjacency matrices are densely connected, so each region of the brain is connected with all the others, disregarding it's physical distance. This creates a problem when graphs are processed by GNN models, because adjacent node features are combined. As a result, all the node features across the graph tend to be very similar, not getting specific information of each node. This is specially critical in our case, as the brain is almost the same for all subjects, but we are searching not general patterns on it, but individual or local affectations in some parts of the brain created by the Multiple Sclerosis. The solution is to prune the connection between regions with low connection values, setting them to 0. In this way when generating the graphs, those regions won't be connected and the aggregation of features will be performed in a more local way that allows the GNN to model more specific and local features of the graph. The effect of pruning on a brain scan is depicted in figure 4.

All the following experiments presented in the paper were conducted with a pruning threshold of 0.7, meaning that the 70% of edges are deleted from every graph. The threshold value is chosen because it demonstrated the best results in cross validation using SVMs and different thresholds. All the experiments and results regarding threshold the threshold value are in the the appendix A.1

7.2 Data Augmentation

One of the main problems in the fields of medical data analysis is the lack of data and our case is no exception. Multiple data augmentation techniques are used to address the problem, and compared against the same techniques without the augmentation to see the real effect of data augmentation in the results. This paper uses the Synthetic Minority Over-sampling Technique (SMOTE) [3] data augmentation for the Baseline models. A graph mixing approach explained below 7.2.2 is applied to both baseline and GNN models.

All experiments involving data augmentation were conducted using the number of data points specified in table 2. To ensure a balanced dataset across all classes, we standardized the number of samples per class to 113. This value was chosen to avoid excessive oversampling, particularly for classes with fewer original data points, while also preserving all available data. Notably, the largest class in the original dataset contains 109 samples for training. By selecting 113 as the target count minimizes the need for augmentation in that class while ensuring that no original data points are discarded.

Category	NDA			DA		
	Train	Test	Val	Train	Test	Val
PRMS	109	33	14	113	6	4
SPMS	21	7	3	113	7	3
PPMS	9	3	1	113	3	1
pwMS	49	15	7	113	15	7
Total	188	58	25	452	58	25

TABLE 2: Table showing the number of datapoints before and after applying data augmentation. Notice that in the case of Test and Validation(Val), the number of samples is the same as the data augmentation is not applied to that part of the dataset to have a fair evaluation. Abbreviations: NDA = No data augmentation applied , DA = Data augmentation applied.

7.2.1 SMOTE

To address class imbalance and limited data availability, we employed the Synthetic Minority Over-sampling Technique (SMOTE) [3] for data augmentation. SMOTE generates synthetic samples for the minority class by interpolating between existing samples and their nearest neighbors in feature space. This method enhances the representation of under-represented classes without simply replicating existing data, thereby reducing the risk of overfitting.

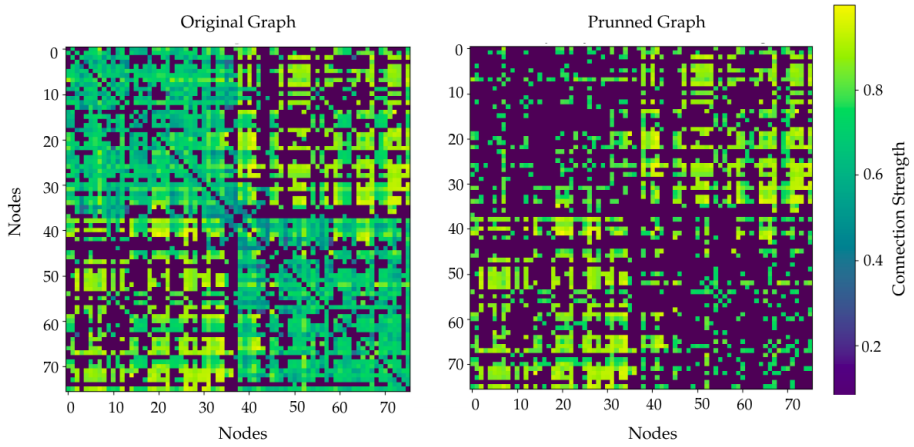

Fig. 4: Original Graph: figure showing the connections of the graphs without any pruning. Pruned Graph: figure showing the adjacency matrix with pruning.

7.2.2 Graph Mixing

Graph Mixing is a data augmentation technique that involves combining information from multiple graphs of the same type. Specifically, this is achieved by replacing half of the columns in the adjacency matrix of one graph with the corresponding half from another graph of the same class. This operation preserves the original values while altering their spatial configuration, thereby creating a new sample that remains within the original data distribution. Since the process is combinatorial, it enables the generation of a large number of diverse samples without introducing artificial noise. The key intuition behind this approach is that the augmented graphs retain the inherent characteristics of the original class, as no new values are introduced, only their arrangement is modified. With this technique we can combine the same graphs multiple times by selecting a subset of different columns to combine.

8 Baseline Models

Given the nature of the dataset and the integration of different data modalities, direct benchmarking against standard techniques becomes challenging. To address this, we employ a series of baseline models, to have some reference values for simpler model performance on the dataset. At the same time these models are aimed at exploring the effects of various data configurations, hyperparameter settings, and data augmentation strategies. Baseline experiments serve as an initial step, providing empirical insights that inform the design and optimization of more complex and computationally intensive models. Extensive testing of baseline models is presented in appendix A.1A.2

While there is no strict guarantee that the optimal hyperparameters identified in simpler models will transfer directly to more advanced architectures, the baselines nonetheless offer valuable guidance. They help delineate promising directions for model refinement and act as reference points to quantify the performance gains attributable to increased model complexity.

The baseline models investigated in this study include Support Vector Machines (SVMs) and Decision Trees. For each model type, we perform cross-validation across multiple data configurations to systematically evaluate their robustness and generalization capabilities. Results of cross validation are presented in appendix A.1

8.1 Baseline Models Data

In these experiments, we directly utilized the values of the adjacency matrices as input features by flattening them to fit the model requirements, with this flattened adjacency matrices the different data configurations described below are tested.

1. **Voting classifiers:** Three classifiers are trained, each one with a different brain scan type. On evaluation each classifier assign a class to each sample and the final classification type is assigned to the class that have more votes. If 3 different classes are assigned by the 3 classifiers, as a tie breaking rule we aggregate the probabilities of the 3 classifiers for each class, and select the class with higher value in the sum of the probabilities of all the classifiers.
2. **Separate classifiers:** 3 classifiers for the 3 different types of brain scans are trained separately .
3. **Joint adjacency matrix:** A matrix combining the 3 brain scans is created following the paper [2].

8.2 Baseline SVM Results

The SVM baseline models in table 3 demonstrate satisfactory performance in distinguishing between healthy volunteers (HV) and people with MS (pwMS). However, they exhibit considerable limitations in accurately classifying the different MS subtypes. Specifically, the SVM models perform poorly in differentiating among MS types, with minor improvements observed only when models are trained on individual scan modalities. Nevertheless, these improvements in MS subtype classification come at the expense of reduced performance in distinguishing pwMS from HV. For example, the SVM model trained on gray matter (GM) scans achieves a 20% increase in overall accuracy for MS subtype classification. However, this is accompanied by a 36.4% decrease in HV classification accuracy and a 14.9% decrease in relapsing-remitting MS (RR-MS) classification accuracy compared to the SVM Voting classifier. A similar trade-off is observed in terms of precision, with a 33.3% improvement in MS subtype classification precision, offset by a 25.6% reduction in HV precision and a 7.2% reduction in RR-MS precision. This pattern is also evident in recall metrics, with a 27.3% decline in HV recall and a 14.9% decline in RR-MS recall. These results highlight a critical trade-off between improving MS subtype classification and maintaining robust differentiation between pwMS and HV. This phenomenon is mostly because of the few data-points of the SP-MS and PP-MS class, showing that the data augmentation applied is not having the desired effect on the data. Finally, better general results are achieved by using graph mixing, rather than using SMOTE approach.

8.3 Baseline Decision Tree Results

For the Decision Tree baseline models in table 3, the best performance is observed with the Voting classifier, which achieves reasonable accuracy in distinguishing between healthy volunteers (HV) and people with MS (pwMS). However, similar to the SVM models, it fails to effectively differentiate among the MS subtypes. Notably, the Decision Tree model trained with rs-fMRI data demonstrates some ability to identify primary progressive MS (PP-MS) cases, which is particularly challenging due to the limited number of samples in this class. Other configurations exhibit lower overall classification performance. These findings also highlight a trade-off between accurately distinguishing HV from pwMS and distinguishing among MS subtypes. Specifically, improvements in MS subtype classification are consistently accompanied by a decline in classification performance for HV across all evaluated metrics. In the decision trees approach we can also generally observe better results using graph mixing than the SMOTE data augmentation approach.

9 Graph Neural Networks

9.1 Graph creation

The creation of the graphs and it's configuration is crucial in the performance of the GNN models. The data is configured in such a way that a single graph joining the three different types of scans of the brain is created. In a first step the adjacency matrices representing the brain graphs are created, so returning a graph with 76 nodes representing the 76 brain regions of the used brain atlas. We also add a weighted connection between brain regions based on the value of the adjacency matrix, connecting those regions if the value connecting them is bigger than 0. In a second step, the DTI matrix nodes are joint with the Gray Matter matrix nodes using the values of the rs-fMRI adjacency matrix between those brain regions. In a third step the rs-fMRI nodes and the Gray Matter nodes are joint together using the DTI connection values between those brain regions. This process is schematically depicted in steps 5 and 6 of figure 3.

Node features: Node features are generated from the graph using normalized graph metrics, together with the volume of each node normalized. The metrics computed to create the features are:

1. **Degree**: The number of edges connected to a node. It quantifies the direct connectivity of a node.

$$k_i = \sum_j a_{ij}$$

where a_{ij} is 1 if there is an edge between node i and node j, and 0 otherwise.

2. **Strength**: In a weighted graph, the strength is the sum of the weights of the edges connected to a node.

$$s_i = \sum_j w_{ij}$$

where w_{ij} is the weight of the edge between node i and node j.

3. **Triangles**: The number of triangles (i.e., sets of three mutually connected nodes) that include a given node.

$$T_i = \frac{1}{2} \sum_{j,k} a_{ij} a_{jk} a_{ki}$$

4. **Closeness Centrality**: Measures how close a node is to all other nodes in the graph. It is the reciprocal of the sum of the shortest path distances from node i to all other nodes.

$$C_i = \frac{N-1}{\sum_{j \neq i} d_{ij}}$$

where d_{ij} is the shortest path distance from node i to node j, and N is the total number of nodes.

5. **Betweenness Centrality**: Measures the extent to which a node lies on paths between other nodes.

$$B_i = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

where σ_{st} is the number of shortest paths from node s to node t, and $\sigma_{st}(i)$ is the number of those paths that pass through node i.

6. **Clustering Coefficient**: Measures the tendency of a node's neighbors to also be connected to each other.

$$C_i = \frac{2T_i}{k_i(k_i - 1)}$$

where T_i is the number of triangles through node i, and k_i is its degree.

7. **local efficiency**: Measures how efficiently information is exchanged in its immediate neighborhood when the node itself is removed.

$$E_{\text{local}}(i) = \frac{1}{k_i(k_i-1)} \sum_{\substack{j,k \in \mathcal{N}(i) \\ j \neq k}} \frac{1}{d_{jk}}$$

where k_i is the degree of node i, $\mathcal{N}(i)$ is the set of neighbors of node i, and d_{jk} is the shortest path length between neighbors j and k within the subgraph induced by $\mathcal{N}(i)$.

8. **pagerank**: Measures node influence using the probability of arriving at a given node by following random paths through the network.

$$PR(i) = \alpha \sum_{j \in \mathcal{N}(i)} \frac{PR(j)}{k_j^{\text{out}}} + (1-\alpha)\frac{1}{N}$$

where $PR(i)$ is the PageRank score of node i, $\mathcal{N}(i)$ represents the set of nodes linking to i, k_j^{out} is the out-degree of node j, N is the total number of nodes, and $\alpha \in [0, 1]$ is a damping factor.

9.2 GNN Models

This study evaluates multiple Graph Neural Network (GNN) architectures and conducts ablation studies to determine which models best adapt to the characteristics of the data. The models considered include Graph Attention Networks(GAT), and Graph Convolutional Networks(GCN).

1. **Graph Attention Networks (GATs)**
Graph Attention Networks [16] introduce a mechanism of self-attention at the graph level, allowing the model to assign different levels of importance to neighboring nodes during message passing. This architecture enhances the model's capacity to learn more nuanced relationships in the graph structure.

2. **Graph Convolutional Networks (GCNs)**
Graph Convolutional Networks [5] generalize the concept of convolution from grid-structured data to graphs. GCNs aggregate information from a node's neighbors to update its representation, enabling the extraction of informative patterns from the graph topology.

			HV			RR-MS			SP-MS			PP-MS		
Arch	**Mod**	**Aug**	**Acc**	**Prec**	**Rec**	**Acc**	**Prec**	**Rec**	**Acc**	**Prec**	**Rec**	**Acc**	**Prec**	**Rec**
VC	**DT**	**GrM**	0.682	0.517	0.682	0.745	**0.745**	0.745	**0.200**	**0.500**	**0.200**	0.000	0.000	0.000
VC	DT	SMT	0.658	0.488	0.658	0.712	0.712	0.712	0.150	0.400	0.150	0.000	0.000	0.000
SM-DTI	DT	GrM	0.273	0.462	0.273	0.745	0.593	0.745	0.100	0.111	0.100	0.000	0.000	0.000
SM-DTI	DT	SMT	0.250	0.420	0.250	0.712	0.562	0.712	0.050	0.090	0.050	0.000	0.000	0.000
SM-fMRI	DT	GrM	0.409	0.450	0.409	0.766	0.706	0.766	0.200	0.286	0.200	**0.250**	**0.200**	**0.250**
SM-fMRI	DT	SMT	0.386	0.430	0.386	0.733	0.670	0.733	0.150	0.250	0.150	0.200	0.150	0.200
SM-GM	DT	GrM	0.091	0.105	0.091	0.532	0.521	0.532	0.100	0.091	0.100	0.000	0.000	0.000
SM-GM	DT	SMT	0.073	0.090	0.073	0.500	0.490	0.500	0.050	0.080	0.050	0.000	0.000	0.000
VC	SVM	GrM	0.636	0.700	0.636	0.914	0.682	0.914	0.000	0.000	0.000	0.000	0.000	0.000
VC	SVM	SMT	0.605	0.670	0.605	0.880	0.645	0.880	0.000	0.000	0.000	0.000	0.000	0.000
MM	SVM	GrM	0.636	0.700	0.636	0.893	0.688	0.893	0.000	0.000	0.000	0.000	0.000	0.000
MM	SVM	SMT	0.614	0.668	0.614	0.860	0.660	0.860	0.000	0.000	0.000	0.000	0.000	0.000
SM-DTI	**SVM**	**GrM**	**0.727**	**0.761**	**0.727**	**0.914**	0.704	**0.914**	0.000	0.000	0.000	0.000	0.000	0.000
SM-DTI	SVM	SMT	0.693	0.730	0.693	0.882	0.670	0.882	0.000	0.000	0.000	0.000	0.000	0.000
SM-fMRI	SVM	GrM	0.681	0.535	0.681	0.808	0.717	0.808	0.000	0.000	0.000	0.000	0.000	0.000
SM-fMRI	SVM	SMT	0.652	0.500	0.652	0.778	0.685	0.778	0.000	0.000	0.000	0.000	0.000	0.000
SM-GM	SVM	GrM	0.363	0.444	0.363	0.766	0.610	0.766	0.200	0.333	0.200	0.000	0.000	0.000
SM-GM	SVM	SMT	0.330	0.400	0.330	0.733	0.575	0.733	0.150	0.280	0.150	0.000	0.000	0.000

TABLE 3: The table show the computed metrics for the presented machine learning models. Here different architechtures for Decison Tree and SVM models are shown. All this models have been trained with data augmentation, using Graph Mixing(GrM) and SMOTE(SMT) approaches. Models of DT and SVM with best overall performance are marked in **bold**. Abbreviations: Arch = Architecture of the model, SVM = Support Vector Machines, DT = Decision Trees, VC = Voting Classifier, MM = Multi-Matrix, SM = SingleMatrix, Aug= Data augmentation, GrM = Graph Mixing, SMT= SMOTE.

9.3 Model Configuration

All the models use an Adam optimizer with a learning rate of $1e^{-4}$ and a weight decay of $1e^{-3}$. The shceduler used is CosineAnnealingWarmRestart with $T_0 = 10$, $T_{mult} = 2$ and $eta_{min} = 1e^{-5}$. The models are trained for 1000 epochs using crossentropy loss for multy class classification. In the following section the specific model configurations are shown.

1. **GCN**: The model consists of two stacked GCNConv layers, with hidden dimension of 128 and 64 respectively. GraphNorm and layer dropout of 0.5 is applied after each GCN layer to normalize node embeddings. On top of that, edge dropout of 0.2 and node dropout of 0.2 is applied to every graph before training to add regularization. The output from the final GCN layer is flattened, resulting in a feature vector of size 4,864, which is passed through a fully connected layer with 128 units and batch normalization. A final linear layer maps the 128-dimensional representation to the 4 output classes. All activations use the ELU function.
2. **GAT**: The model consists of two stacked GATConv layers, with hidden dimension of 32 and 8 respectively, with 4 heads in each layer. GraphNorm and dropout of 0.5 is applied after each GAT layer to normalize node embeddings. On top of that, edge dropout of 0.2 and node dropout of 0.2 is applied to every graph before training to add regularization. The output from the final GAT layer is flattened, resulting in a feature vector of size 4,864, which is passed through a fully connected layer with 32 units and batch normalization. A final linear layer maps the 32-dimensional representation to the 4 output classes. All activations use the ELU function.

9.4 Results

As shown in table 4, both multi-modal GCN and GAT has been trained with data augmentation and without it. Notice that only precision, recall and F1 scores are provided because the accuracy metric in a context of unbalanced dataset is a misleading metric that does not present relevant information.

In table 4, for the baseline GAT model, best performance was achieved for the RR-MS class, with an F1 score of 67.6%, followed by the HV class with an F1 score of 58.1%. The model performed poorly in identifying SP-MS with an F1 of 18.2% and completely failed to classify any PP-MS samples with an F1 of 0%. After applying data augmentation, the GAT model improved in distinguishing RR-MS achieving an F1 of 72.9% and slightly improved HV performance with an F1 of 59.3%, but showed no improvement in SP-MS or PP-MS classification, which remained at 0%.

In table4, the GCN model showed better general performance. Without data augmentation, it achieved high performance for RR-MS with an F1 of 79.5% and HV with F1 of 64.5%, with limited success in identifying PP-MS with an F1 of 40.0%, and failed entirely on SP-MS with an F1 of 0.00%. With data augmentation, the GCN model further improved on both RR-MS with an F1 of 80.6% and HV with F1 of 66.7%. Notably, performance on PP-MS improved significantly to an F1 score of 57.1%, although SP-MS remained unclassified with F1 of 0%.

Results in table 4 highlight that data augmentation moderately improves the classification performance, especially in the dominant classes of HV and RR-MS. Also GCN models outperform GAT models across all configurations. However, the challenge of classifying SP-MS and PP-MS remains, likely due to class imbalance and limited sample availability.

9.4.1 Individual Matrices

The following models are trained by each types of analysis individually, not following a multi-matrix approach. This analysis is performed in order to asses the difference in prediction by following a multi-matrix approach against a single-matrix one.

As shown in 4, the GCN models trained on different brain scan modalities exhibit varying classification performance across the healthy volunteers (HV) and multiple sclerosis (MS) subtypes. The GCN-DTI model performs best overall, particularly in identifying RR-MS subjects, achieving a precision of 71.0%, recall of 81.8%, and an F1 score of 76.1%, indicating reliable sensitivity and specificity for this class.

			HV			RR-MS			SP-MS			PP-MS		
Model	**Data**	**DA**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
GCN	DTI	DA	0.588	0.666	0.625	0.711	0.818	0.761	0.000	0.000	0.000	0.333	0.333	0.333
GCN	fMRI	DA	0.500	0.533	0.516	0.724	0.636	0.677	0.400	**0.286**	**0.333**	0.000	0.000	0.000
GCN	GM	DA	0.289	**0.733**	0.415	0.579	0.333	0.423	0.000	0.000	0.000	0.000	0.000	0.000
GCN*	**MM**	**DA**	**0.692**	0.600	**0.645**	0.707	**0.879**	0.784	**0.500**	0.143	0.222	**0.500**	**0.333**	**0.400**
GCN	MM	NDA	0.625	0.666	0.645	**0.725**	0.878	**0.794**	0.000	0.000	0.000	**0.500**	**0.333**	**0.400**
GAT	MM	DA	0.667	0.533	0.593	0.658	0.818	0.729	0.000	0.000	0.000	0.000	0.000	0.000
GAT	MM	NDA	0.562	0.600	0.581	0.631	0.727	0.676	0.250	0.143	0.182	0.000	0.000	0.000

TABLE 4: Table showing the results of the metrics computed for both GCN and GAT models evaluated with different datasets. The **GCN*** present the best model in most of the metrics. Best results for each metric per class are highlighted in **bold**. Abbreviations: DA = Data augmentation, NDA = No data augmentation applied, MM = multi-modal (combination of the 3 matrices).

Performance on the HV class with GCN-DTI is moderate, with an F1 score of 62.5%, driven by a precision of 58.82% and recall of 66.6%. However, classification performance degrades significantly for SP-MS and PP-MS, both showing 0.00% precision, recall, and F1 score, suggesting that the model fails to identify any cases of these subtypes.

The GCN rs-fMRI model yields slightly more balanced, but generally lower score results, with RR-MS classification achieving an F1 score of 67.7%, but with reduced recall 63.6% compared to GCN trained with DTI. The model identifies some SP-MS cases with F1 of 33.3% but fails entirely on PP-MS with a 0.00% across all metrics.

GCN-GM demonstrates the weakest overall performance. While it shows relatively high recall for the HV class with a 73.3%, its precision is poor with a 28.9%, resulting in an F1 score of only 41.5%. The model also fails to identify any SP-MS or PP-MS cases, with all metrics at 0.00%, and shows limited performance on RR-MS with an F1 of 42.3%.

Non of the individual models presented in table 4 overcome the best multi-modal approach, experimentally showing that joining multiple representations of the brain is beneficial for classification, rather than individually train for each brain scan type.

9.4.2 Binary classification

In the following section we analyze the binary results, obtained by keeping the HV as an untouched class, but merging all the MS types in a single class called MS. All the binary results are computed with the structure and data augmentation of the best GCN model in the last section, but changing the number of output neurons, from four to two.

Model	Class	Acc	Prec	Rec	F1
GCN	HV	-	0.727	0.533	0.615
GCN	MS	-	0.851	0.930	0.889
GCN	Overall	0.828	0.819	0.827	0.818

TABLE 5: GCN binary classification results between HV and MS patients

As shown in Table 5, the GCN model achieved an overall accuracy of 82.8%, with a balanced F1 score of 81.8%, indicating strong general performance in distinguishing between HV and MS patient groups. When analyzed by class, the model demonstrates a clear performance asymmetry. The classification of pwMS yielded high precision of 85.1%, recall of 93.0%, and F1 score of 88.9%, suggesting that the model is highly effective at correctly identifying MS cases with minimal false positives and false negatives. In contrast, the performance on the HV class was notably lower, with an F1 score of 61.5%, driven by modest precision of 72.7% and particularly low recall of 53.3%. This indicates a tendency of the model to misclassify HV subjects as MS, due to the imbalance of subjects in the dataset.

10 INTERPRETABILITY

In order to better interpret the results we can get the feature relevance for explaining the results of each fitted model following the approach presented in [17]

10.1 Multi-Matrix

10.1.1 Multi-class classification

The attribution matrix corresponding to the multi-class classification task in figure 5 show how node degree and node strength exhibit consistently high attribution scores across a broad range of brain regions. This pattern reflects the biological reality that MS leads to progressive degradation of brain connections and lower interaction intensity between affected regions. As the disease progresses, the extent of these disruptions varies across MS subtypes, leading to measurable differences in nodal degree and strength. These variations provide a discriminative signal that enables the model to differentiate between disease stages and from healthy controls. A result in line with other scientific studies [2][11].

Betweenness Centrality feature also displays consistently high importance across nodes, suggesting that long-range information transfer within the brain network is significantly altered in MS. As demyelination disrupts critical communication pathways, the brain may attempt to reroute information through alternative paths, thereby changing centrality roles to different nodes. This reconfiguration is effectively captured by the GCN model and highlights the global impact of MS on network topology.

In addition, brain density feature gain prominence in later stages of the disease. This is consistent with clinical observations of regional atrophy and tissue loss, particularly in gray matter, which results in reduced nodal density in affected regions. The model sensitivity to these changes show the utility of density-based features in identifying more severe stages of MS, such as SP-MS and PP-MS.

Together, these attribution patterns offer biologically plausible interpretation for how MS pathology manifests in brain network structure. The presented results go in line with state of the art investigation [13][11][2], reinforcing the explainablility and clinical relevance of the proposed GCN-based approach.

10.1.2 Binary classification

The attribution matrix for binary classification shown in figure 5 reveal how some features in some regions of the brain have a heightened importance, suggesting regionally localized network alterations associated with MS. Betweenness Centrality and Local Efficiency consistently show high attribution values across multiple nodes. These metrics reflect global and local integration properties respectively, indicating that MS pathology may affect both long-range communication pathways and local circuit efficiency.

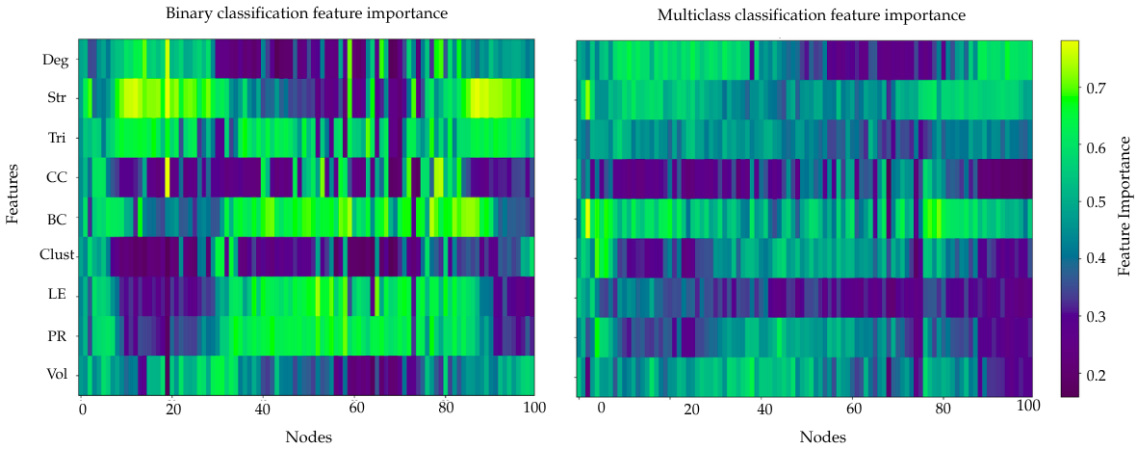

Fig. 5: Figure showing the importance of each feature for the models trained for binary classification and multi-class classification using the multi-matrix approach. In the figure the vertical axis represents the features associated to each node. On the horizontal axis we find the enumeration of only the first 100 nodes of the graph for the sake of easier interpretability. Abbreviations: Deg = Degree, Str = Strength, Tri = Triangles, CC = Closeness centrality, 5- BC = Betweenness centrality, Clust = Clustering coefficient, LE = Local efficiency, PR = Pagerank, Vol = Graph Volumes

The prominence of betweenness centrality in discriminative features underscores the role of topological disconnection in MS. As betweenness centrality reflects control over information flow, its high importance implies disrupted hub function of certain nodes for pwMS. This reflect how the degradation of brain connections makes the brain find different paths than in a usual brain for the information to flow, changing the betweenness centrality of many nodes. These results go in line with state of the art investigation [13][11]

Similar to multi-class classification, node strength and degree also play a crucial role in detecting MS brains because the degradation of brain connections due to MS reduce the degree and strength of some nodes, in line with the results presented in [2].

10.2 Single-Matrix

Figure 6 show 3 matrices, each one for a different model trained with one type of data, DTI, rs-fMRI or GM. Each matrix quantifies the relative importance of each graph-theoretical feature across all 76 brain regions (nodes), providing insight into how structural characteristics influence each different type of data on classification.

10.2.1 DTI

The explainer for the GCN trained with DTI samples in table 6, reveals broad and high attribution for node degree and node strength, consistent across a wide distribution of brain regions. These features quantify direct connectivity and the cumulative strength of edges, directly tied to white matter clusters integrity, which is often compromised in MS due to demyelination and axonal loss. The prominence of betweenness centrality and local efficiency further supports the notion of global network disruption in MS patients, particularly in SP-MS and PP-MS, where compensatory re-routing of information flow becomes necessary. Notably, triangle count and clustering coefficient also exhibit focal importance, reflecting disrupted local circuit integrity. Overall, DTI features emphasize both global disintegration and regional structural loss, aligning with white matter degradation in progressive MS subtypes [11].

10.2.2 rs-fMRI

In the attribution map for GCN model trained with rs-fMRI samples in table 6, node strength emerges as the most discriminative feature, with high attribution scores across multiple regions. This suggests that the model heavily relies on the strength of functional connectivity between brain regions to differentiate among MS types and HV. This result go in line with MS literature, where in early stages like RR-MS, functional reorganization or hyper-connectivity may temporarily compensate for structural loss [11][9]. Related to node strength, clustering coefficient is also discriminative, indicating that some brain clusters are disrupted due to MS degradation. The distributed importance of betweenness centrality and PageRank can be indicative of altered global communication efficiency, suggesting reconfiguration of brain hubs.

10.2.3 Gray Matter

Figure 6 presents the feature attribution matrix obtained from the GCN model trained exclusively on Gray Matter (GM) data.

Node Degree and Node Strength exhibit the highest and most consistent attribution across nodes. These features capture local connectivity and edge weight accumulation respectively, and their dominance aligns with known MS-related structural degradation [13]. MS is characterized by demyelination and cortical atrophy, particularly in later stages, which leads to a reduction in both the number and intensity of inter-regional connections [11]. The model leverages these alterations to effectively differentiate between healthy controls and various MS types. The heterogeneity in the attribution distribution across nodes suggests region-specific vulnerability, possibly reflecting specific atrophy patterns in GM.

Pagerank and node density also show moderate to high attribution values, particularly in localized clusters of nodes. Pagerank, which quantifies a node's influence in the network, becomes increasingly important as MS progression alters the brain's hierarchical connectivity structure. Reduced PageRank in key hub regions may signify a loss of nodal importance due to neurodegeneration. Meanwhile, the elevated importance of density-related features points to global reductions in nodal compactness and tissue volume, clear marks of GM atrophy in progressive MS subtypes.

In contrast, features related to high-order topology, such as triangles and closeness centrality contribute minimally. This indicates that the GM model captures degradation in nodal properties rather than complex topological structures.

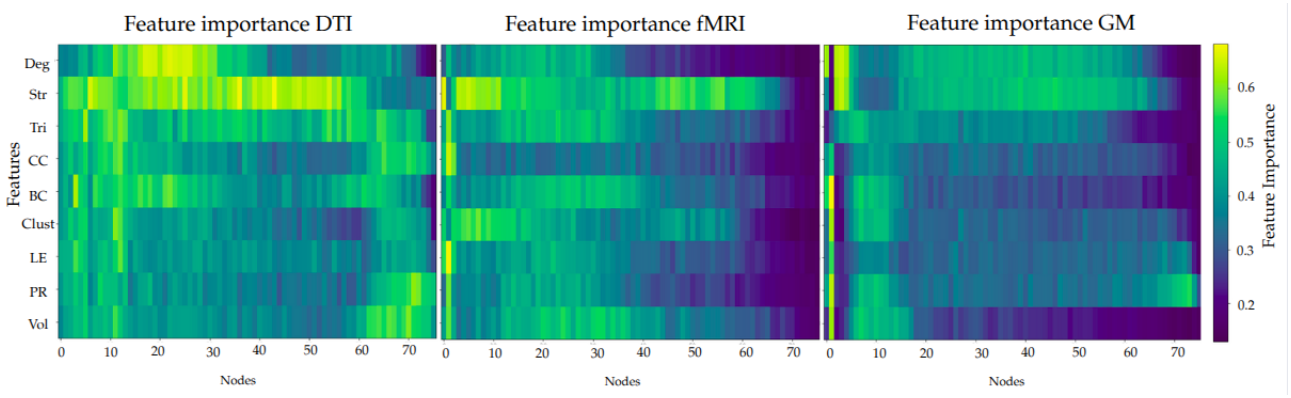

Fig. 6: Figure showing the importance of each feature for the final model trained only with the DTI scans. In the figure each number in features axis is associated with a feature, computed for each node of the graph. 1- Degree, 2- Strength, 3- Triangles, 4-Closeness centrality, 5- Betweenness centrality, 6- Clustering coefficient, 7-Local efficiency, 8-Pagerank

11 Conclusions

This work presents a multilayer, **machine learning** and **graph neural network based** approach for the classification of Multiple Sclerosis (MS), integrating three different neuroimaging modalities: Diffusion Tensor Imaging (DTI), Resting State Functional Magnetic Resonance Imaging (rs-fMRI), and Gray Matter (GM) volumes. The proposed method demonstrates consistent improvements over single-layer approaches across all the performance metrics presented for both binary and multiclass classification tasks. This results comply with the objective 3 of the project.

The key contribution of this study lies in the **multi-matrix approach**, which allows the aggregation of complementary information from distinct brain scans into a unified graph representation. **This fusion leads to a more robust modeling of brain connectivity**, ultimately improving the generalization capability of the GNN models. The main idea behind the multi-matrix is reducing the bias of the samples by incorporating more information in each datapoint, a positive thing in the scenario in hand where the given data samples are very limited. On top of that, due to the intrinsic complexity of the brain, by combining different data modalities in a unique representation allow the models to capture more complex relationships between brain regions and discover patterns to characterize MS.

The **GNN multi-matrix models outperform all single-layer models**, validating the hypothesis that joint analysis of multi-matrix brain data mitigates bias and overfitting, a critical advantage given the relatively limited sample size. The robustness achieved through this integration is particularly important in medical applications where high sensitivity and specificity are vital. The multi-matrix approach presented comply with objective 1 of combining the tree different types of data for classification.

While **data augmentation graph mixing** technique show a positive impact on performance, particularly in underrepresented MS subtypes, the **observed improvements are moderate**, and in some evaluations the gain in performance is not clear. These results suggest that while augmentation helps in predicting under represented classes, it cannot fully compensate for the lack of diverse clinical samples, especially in rare MS phenotypes such as SP-MS and PP-MS. This reinforces the need for future studies to explore more effective augmentation techniques or to prioritize data collection efforts in underrepresented clinical subgroups.

The **GCN model consistently outperforms GAT architectures**, leading to the conclusion that message-passing mechanisms based on convolutional operators are particularly well-suited for integrating spatial and topological information in brain graphs. In the case of GAT, the lack of data might have specially affected the model performance. However, both models consistently fail in correctly predicting under represented classes in the original dataset. By the design, training and evaluation of these modes we comply with objective 2

Furthermore, **interpretability analysis** using graph feature attribution maps reveals that topological features such as betweenness centrality and local efficiency are particularly influential in the classification task. These metrics are specially relevant because they reflect how brain regions lose local connectivity while compensating these lose with relocation of these paths due to MS. These findings align with clinical and scientific state of the art results [13][11] showing that MS disproportionately affects brain network hubs and disrupts efficient communication pathways.

Despite the models presented in this work are trained for MS classification, due to their capability of modeling complex graph structures they have the potential to be applied for the classification of other brain related diseases. This opens the window for a broader study with scans from multiple diseases, giving insight and further interpretation to them.

Finally, the proper documentation and analysis of the results using a written report following the proposed methodology and work plan makes us to comply with objective 4

Acknowledgments

Special acknowledgment to Jodi Casas, who has been my tutor during all the process of elaboration of this paper and provided vital insights and corrections to the proposed work.

References

[1] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, M. S. Albert, and R. J. Killiany, "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest," *NeuroImage*, vol. 31, no. 3, pp. 968–980, Jul. 2006.

[2] J. Casas-Roma, E. Martinez-Heras, A. Solé-Ribalta, E. Solana, E. Lopez-Soley, F. Vivó, M. Diaz-Hurtado, S. Alba-Arbalat, M. Sepulveda, Y. Blanco, A. Saiz, J. Borge-Holthoefer, S. Llufriu, and F. Prados, "Applying multilayer analysis to morphological, structural, and functional brain networks to identify relevant dysfunction patterns," *Network Neuroscience*

(Cambridge, Mass.), vol. 6, no. 3, pp. 916–933, Jul. 2022.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Oversampling Technique," Jun. 2011, arXiv:1106.1813. [Online]. Available: http://arxiv.org/abs/1106.1813

[4] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: Uses and interpretations," *NeuroImage*, vol. 52, no. 3, pp. 1059–1069, Sep. 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S105381190901074X

[5] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," Feb. 2017, arXiv:1609.02907. [Online]. Available: http://arxiv.org/abs/1609.02907

[6] P. Chapman, *CRISP-DM 1.0: Step-by-step Data Mining Guide*. SPSS, 2000. [Online]. Available: https://books.google.es/books?id=po7FtgAACAAJ

[7] A. Marzullo, G. Kocevar, C. Stamile, F. Durand-Dubief, G. Terracina, F. Calimeri, and D. Sappey-Marinier, "Classification of Multiple Sclerosis Clinical Profiles via Graph Convolutional Neural Networks," *Frontiers in Neuroscience*, vol. 13, Jun. 2019. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2019.00594/full

[8] S.-H. Wang, C. Tang, J. Sun, J. Yang, C. Huang, P. Phillips, and Y.-D. Zhang, "Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling," *Frontiers in Neuroscience*, vol. 12, Nov. 2018. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2018.00818/full

[9] G. Martí-Juan, J. Sastre-Garriga, E. Martinez-Heras, A. Vidal-Jordana, S. Llufriu, S. Groppa, G. Gonzalez-Escamilla, M. A. Rocca, M. Filippi, E. A. Høgestøl, H. F. Harbo, M. A. Foster, A. T. Toosy, M. M. Schoonheim, P. Tewarie, G. Pontillo, M. Petracca, Rovira, G. Deco, and D. Pareto, "Using The Virtual Brain to study the relationship between structural and functional connectivity in patients with multiple sclerosis: a multicenter study," *Cerebral Cortex (New York, N.Y.: 1991)*, vol. 33, no. 12, pp. 7322–7334, Jun. 2023.

[10] M. A. Rocca, P. Preziosa, F. Barkhof, W. Brownlee, M. Calabrese, N. De Stefano, C. Granziera, S. Ropele, A. T. Toosy, Vidal-Jordana, M. Di Filippo, and M. Filippi, "Current and future role of MRI in the diagnosis and prognosis of multiple sclerosis," *The Lancet Regional Health. Europe*, vol. 44, p. 100978, Sep. 2024.

[11] 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, and S. Llufriu, "Modified connectivity of vulnerable brain nodes in multiple sclerosis, their impact on cognition and their discriminative value," *Scientific Reports*, vol. 9, no. 1, p. 20172, Dec. 2019. [Online]. Available: https://www.nature.com/articles/s41598-019-56806-z

[12] M. Mijalkov, E. Kakaei, J. B. Pereira, E. Westman, G. Volpe, and Alzheimer's Disease Neuroimaging Initiative, "BRAPH: A graph theory software for the analysis of brain connectivity," *PloS One*, vol. 12, no. 8, p. e0178798, 2017.

[13] G. Pontillo, F. Prados, A. M. Wink, B. Kanber, A. Bisecco, T. A. A. Broeders, A. Brunetti, A. Cagol, M. Calabrese, M. Castellaro, S. Cocozza, E. Colato, S. Collorone, R. Cortese, N. De Stefano, L. Douw, C. Enzinger, M. Filippi, M. A. Foster, A. Gallo, G. Gonzalez-Escamilla, C. Granziera, S. Groppa, H. F. Harbo, E. A. Høgestøl, S. Llufriu, L. Lorenzini, E. Martinez-Heras, S. Messina, M. Moccia, G. O. Nygaard, J. Palace, M. Petracca, D. Pinter, M. A. Rocca, E. Strijbis, A. Toosy, P. Valsasina, H. Vrenken, O. Ciccarelli, J. H. Cole, M. M. Schoonheim, F. Barkhof, and MAGNIMS study group, "More Than the Sum of Its Parts: Disrupted Core Periphery of Multiplex Brain Networks in Multiple Sclerosis," *Human Brain Mapping*, vol. 46, no. 1, p. e70107, Jan. 2025.

[14] H. Cui, W. Dai, Y. Zhu, X. Kan, A. A. C. Gu, J. Lukemire, L. Zhan, L. He, Y. Guo, and C. Yang, "BrainGB: A Benchmark for Brain Network Analysis with Graph Neural Networks," Nov. 2022, arXiv:2204.07054. [Online]. Available: http://arxiv.org/abs/2204.07054

[15] X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang, D. Scheinost, L. H. Staib, P. Ventola, and J. S. Duncan, "BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis," *Medical Image Analysis*, vol. 74, p. 102233, Dec. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841521002784

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018. [Online]. Available: https://arxiv.org/abs/1710.10903

[17] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1903.03894

APPENDIX

A.1 SVM cross validation

In this appendix section a more extensive cross validation of the SVMs is presented. We evaluated different kernel types, pruning thresholds, and mixing levels to determine the optimal configuration.

A.1.1 Per-Class Performance (RBF Kernel)

The detailed accuracy metrics for each class are provided in Table 7.

Tables 8 and 9 show the precision and recall per class, respectively.

A.1.2 Linear Kernel Results

Best results depicting the accuracy,precision and recall of linear kernel SVMs are in tables 11 12 13. The main results can be found in table 10.

A.1.3 Per-Class Performance (Linear Kernel)

The detailed accuracy metrics for each class are provided in Table 11.

Tables 12 and 13 show the precision and recall per class, respectively.

A.2 Decision Trees cross validation

In this appendix section a more extensive cross validation for the Decision Trees models is presented. We evaluate different data mixing levels and pruning thresholds.

A.2.1 Results Overview

The classification performance was measured using accuracy, precision, recall, and F1-score.

Figure 11 shows how accuracy varies with different threshold and mixing level combinations.

The detailed accuracy metrics for each class are provided in Table 15.

Tables 16 and 17 show the precision and recall per class, respectively.

TABLE 6: Main Results for SVM with RBF kernel

Threshold	Mixing Level	Mean Accuracy	Mean Precision	Mean Recall	Mean F1 Score
0.500000	0	0.5904	0.4388	0.5904	0.4700
0.500000	1	0.5904	0.4388	0.5904	0.4700
0.500000	2	0.5904	0.4388	0.5904	0.4700
0.500000	3	0.5904	0.4388	0.5904	0.4700
0.500000	4	0.5904	0.4388	0.5904	0.4700
0.700000	0	**0.6022**	**0.4653**	**0.6022**	**0.4789**
0.700000	1	**0.6022**	**0.4653**	**0.6022**	**0.4789**
0.700000	2	**0.6022**	**0.4653**	**0.6022**	**0.4789**
0.700000	3	**0.6022**	**0.4653**	**0.6022**	**0.4789**
0.700000	4	**0.6022**	**0.4653**	**0.6022**	**0.4789**
0.800000	0	0.5904	0.4310	0.5904	0.4585
0.800000	1	0.5904	0.4310	0.5904	0.4585
0.800000	2	0.5904	0.4310	0.5904	0.4585
0.800000	3	0.5904	0.4310	0.5904	0.4585
0.800000	4	0.5904	0.4310	0.5904	0.4585
0.900000	0	0.5662	0.3210	0.5662	0.4096
0.900000	1	0.5662	0.3210	0.5662	0.4096
0.900000	2	0.5662	0.3210	0.5662	0.4096
0.900000	3	0.5662	0.3210	0.5662	0.4096
0.900000	4	0.5662	0.3210	0.5662	0.4096

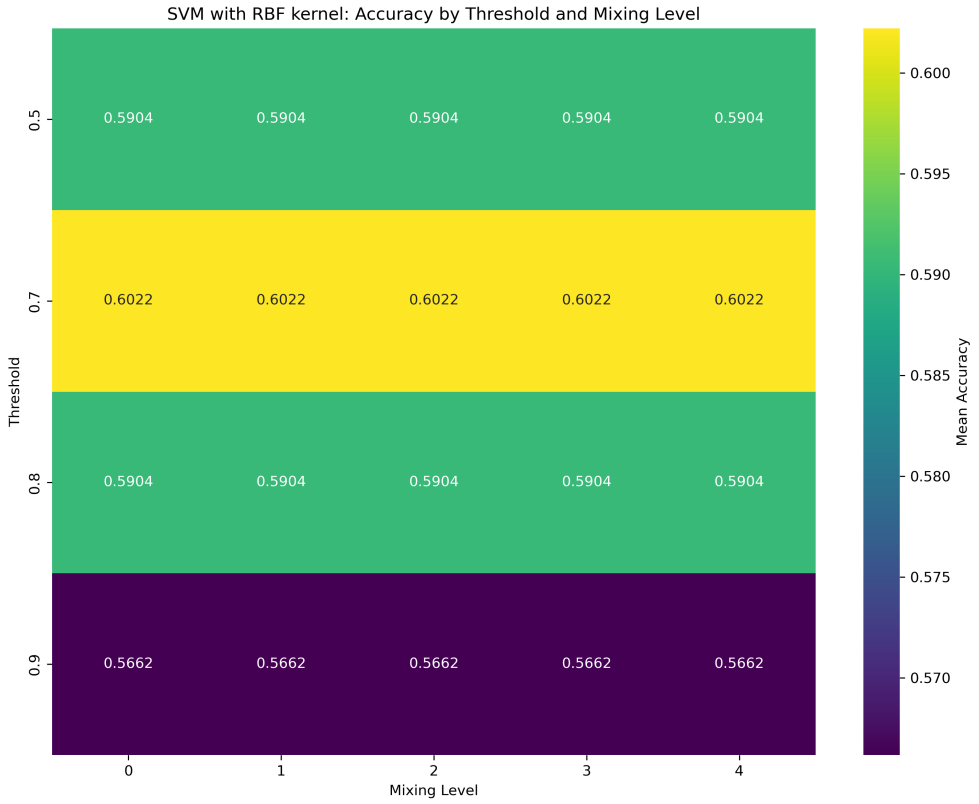

Fig. 7: Heatmap of Mean Accuracy for SVM with RBF kernel

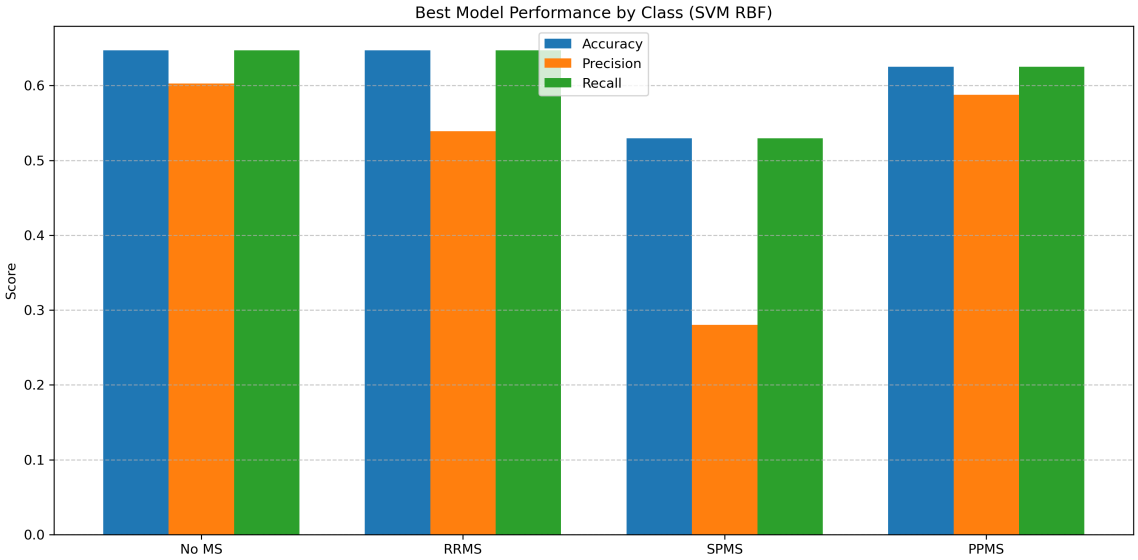

Fig. 8: Performance by Class for Best SVM Model with RBF kernel (Threshold: 0.7, Mixing Level: 0.0)

TABLE 7: Per-Class Accuracy for SVM with RBF kernel

Threshold	Mixing Level	No MS Accuracy	RRMS Accuracy	SPMS Accuracy	PPMS Accuracy
0.500000	0	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	1	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	2	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	3	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	4	0.5882	**0.6471**	**0.5294**	**0.6250**
0.700000	0	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	1	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	2	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	3	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	4	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.800000	0	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	1	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	2	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	3	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	4	**0.6471**	0.5882	**0.5294**	**0.6250**
0.900000	0	0.5882	0.5882	**0.5294**	0.5625
0.900000	1	0.5882	0.5882	**0.5294**	0.5625
0.900000	2	0.5882	0.5882	**0.5294**	0.5625
0.900000	3	0.5882	0.5882	**0.5294**	0.5625
0.900000	4	0.5882	0.5882	**0.5294**	0.5625

TABLE 8: Per-Class Precision for SVM with RBF kernel

Threshold	Mixing Level	No MS Precision	RRMS Precision	SPMS Precision	PPMS Precision
0.500000	0	0.4706	**0.5392**	**0.2803**	**0.5875**
0.500000	1	0.4706	**0.5392**	**0.2803**	**0.5875**
0.500000	2	0.4706	**0.5392**	**0.2803**	**0.5875**
0.500000	3	0.4706	**0.5392**	**0.2803**	**0.5875**
0.500000	4	0.4706	**0.5392**	**0.2803**	**0.5875**
0.700000	0	**0.6029**	**0.5392**	**0.2803**	**0.5875**
0.700000	1	**0.6029**	**0.5392**	**0.2803**	**0.5875**
0.700000	2	**0.6029**	**0.5392**	**0.2803**	**0.5875**
0.700000	3	**0.6029**	**0.5392**	**0.2803**	**0.5875**
0.700000	4	**0.6029**	**0.5392**	**0.2803**	**0.5875**
0.800000	0	**0.6029**	0.3676	**0.2803**	**0.5875**
0.800000	1	**0.6029**	0.3676	**0.2803**	**0.5875**
0.800000	2	**0.6029**	0.3676	**0.2803**	**0.5875**
0.800000	3	**0.6029**	0.3676	**0.2803**	**0.5875**
0.800000	4	**0.6029**	0.3676	**0.2803**	**0.5875**
0.900000	0	0.3460	0.3460	**0.2803**	0.3164
0.900000	1	0.3460	0.3460	**0.2803**	0.3164
0.900000	2	0.3460	0.3460	**0.2803**	0.3164
0.900000	3	0.3460	0.3460	**0.2803**	0.3164
0.900000	4	0.3460	0.3460	**0.2803**	0.3164

TABLE 9: Per-Class Recall for SVM with RBF kernel

Threshold	Mixing Level	No MS Recall	RRMS Recall	SPMS Recall	PPMS Recall
0.500000	0	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	1	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	2	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	3	0.5882	**0.6471**	**0.5294**	**0.6250**
0.500000	4	0.5882	**0.6471**	**0.5294**	**0.6250**
0.700000	0	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	1	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	2	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	3	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.700000	4	**0.6471**	**0.6471**	**0.5294**	**0.6250**
0.800000	0	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	1	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	2	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	3	**0.6471**	0.5882	**0.5294**	**0.6250**
0.800000	4	**0.6471**	0.5882	**0.5294**	**0.6250**
0.900000	0	0.5882	0.5882	**0.5294**	0.5625
0.900000	1	0.5882	0.5882	**0.5294**	0.5625
0.900000	2	0.5882	0.5882	**0.5294**	0.5625
0.900000	3	0.5882	0.5882	**0.5294**	0.5625
0.900000	4	0.5882	0.5882	**0.5294**	0.5625

TABLE 10: Main Results for SVM with LINEAR kernel

Threshold	Mixing Level	Mean Accuracy	Mean Precision	Mean Recall	Mean F1 Score
0.500000	0	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.500000	1	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.500000	2	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.500000	3	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.500000	4	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.700000	0	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.700000	1	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.700000	2	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.700000	3	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.700000	4	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.800000	0	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.800000	1	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.800000	2	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.800000	3	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.800000	4	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.900000	0	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.900000	1	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.900000	2	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.900000	3	**0.7235**	**0.5981**	**0.7235**	**0.6509**
0.900000	4	**0.7235**	**0.5981**	**0.7235**	**0.6509**

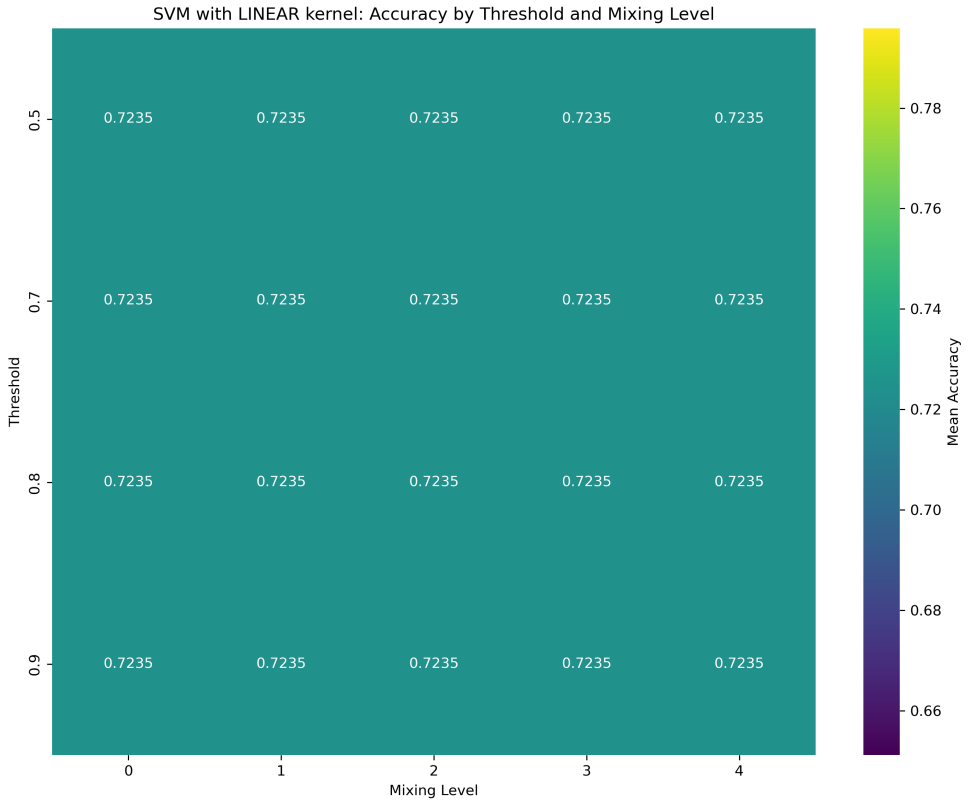

Fig. 9: Heatmap of Mean Accuracy for SVM with LINEAR kernel

TABLE 11: Per-Class Accuracy for SVM with LINEAR kernel

Threshold	Mixing Level	No MS Accuracy	RRMS Accuracy	SPMS Accuracy	PPMS Accuracy
0.500000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**

TABLE 12: Per-Class Precision for SVM with LINEAR kernel

Threshold	Mixing Level	No MS Precision	RRMS Precision	SPMS Precision	PPMS Precision
0.500000	0	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.500000	1	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.500000	2	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.500000	3	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.500000	4	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.700000	0	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.700000	1	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.700000	2	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.700000	3	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.700000	4	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.800000	0	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.800000	1	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.800000	2	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.800000	3	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.800000	4	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.900000	0	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.900000	1	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.900000	2	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.900000	3	**0.5350**	**0.5995**	**0.6301**	**0.6167**
0.900000	4	**0.5350**	**0.5995**	**0.6301**	**0.6167**

TABLE 13: Per-Class Recall for SVM with LINEAR kernel

Threshold	Mixing Level	No MS Recall	RRMS Recall	SPMS Recall	PPMS Recall
0.500000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.500000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.700000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.800000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	0	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	1	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	2	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	3	**0.6471**	**0.7059**	**0.7647**	**0.7500**
0.900000	4	**0.6471**	**0.7059**	**0.7647**	**0.7500**

TABLE 14: Main Results for Trees

Threshold	Mixing Level	Mean Accuracy	Mean Precision	Mean Recall	Mean F1 Score
0.500000	0	0.5426	0.5366	0.5426	0.5344
0.500000	1	0.5647	**0.5982**	0.5647	0.5688
0.500000	2	0.4949	0.5117	0.4949	0.4969
0.500000	3	0.5191	0.5133	0.5191	0.5099
0.500000	4	0.5426	0.5355	0.5426	0.5337
0.700000	0	0.4471	0.4329	0.4471	0.4362
0.700000	1	0.4699	0.4254	0.4699	0.4409
0.700000	2	0.4838	0.4822	0.4838	0.4744
0.700000	3	0.4588	0.4448	0.4588	0.4477
0.700000	4	0.4713	0.4662	0.4713	0.4620
0.800000	0	0.5162	0.5040	0.5162	0.5036
0.800000	1	0.4213	0.4682	0.4213	0.4299
0.800000	2	0.4581	0.4619	0.4581	0.4456
0.800000	3	0.4691	0.5116	0.4691	0.4714
0.800000	4	0.4809	0.4775	0.4809	0.4737
0.900000	0	0.5441	0.5280	0.5441	0.5286
0.900000	1	0.5441	0.5427	0.5441	0.5382
0.900000	2	0.5566	0.5251	0.5566	0.5347
0.900000	3	**0.6051**	0.5801	**0.6051**	**0.5853**
0.900000	4	0.5551	0.5444	0.5551	0.5450

TABLE 15: Per-Class Accuracy for Trees with

Threshold	Mixing Level	No MS Accuracy	RRMS Accuracy	SPMS Accuracy	PPMS Accuracy
0.500000	0	0.5882	0.6471	0.3529	**0.6875**
0.500000	1	**0.6471**	0.6471	0.5294	0.6250
0.500000	2	0.5294	0.4706	0.4118	0.6250
0.500000	3	0.5882	0.6471	0.2353	0.6250
0.500000	4	**0.6471**	0.5882	0.3529	0.6250
0.700000	0	0.3529	0.4706	0.4118	0.4375
0.700000	1	0.4706	0.5294	0.4118	0.3750
0.700000	2	0.4118	0.4706	0.4118	0.5000
0.700000	3	0.4118	0.4706	0.4118	0.4375
0.700000	4	0.5294	0.4706	0.2941	0.5000
0.800000	0	0.4706	**0.7059**	0.5294	0.3750
0.800000	1	0.4706	0.4118	0.4118	0.3750
0.800000	2	0.5294	0.4118	0.4118	0.4375
0.800000	3	0.5294	0.4118	0.5294	0.3750
0.800000	4	0.5882	0.5294	0.4118	0.4375
0.900000	0	0.3529	0.5294	0.5882	0.6250
0.900000	1	0.3529	0.5294	0.5882	0.6250
0.900000	2	0.3529	0.5294	0.5882	0.6250
0.900000	3	0.4118	0.5294	**0.6471**	**0.6875**
0.900000	4	0.5294	0.4706	0.5882	0.5625

TABLE 16: Per-Class Precision for Tress with

Threshold	Mixing Level	No MS Precision	RRMS Precision	SPMS Precision	PPMS Precision
0.500000	0	0.5098	0.6912	0.3863	0.6375
0.500000	1	0.5641	0.7108	**0.6392**	0.6250
0.500000	2	**0.6216**	0.5588	0.3908	0.5625
0.500000	3	0.5490	0.6043	0.2958	0.6172
0.500000	4	0.5980	0.6340	0.3908	0.5341
0.700000	0	0.3209	0.5392	0.4412	0.3381
0.700000	1	0.3620	0.5556	0.3627	0.3646
0.700000	2	0.4118	0.5574	0.4496	0.4735
0.700000	3	0.3431	0.5147	0.4779	0.3693
0.700000	4	0.4626	0.5147	0.3249	0.5286
0.800000	0	0.4332	**0.7353**	0.4652	0.3527
0.800000	1	0.4588	0.5686	0.4367	0.4048
0.800000	2	0.5059	0.4804	0.3627	0.4241
0.800000	3	0.4902	0.6373	0.4811	0.4152
0.800000	4	0.5098	0.6471	0.3627	0.3958
0.900000	0	0.4538	0.5294	0.5000	0.5758
0.900000	1	0.4538	0.5784	0.5000	0.6000
0.900000	2	0.4303	0.5098	0.5000	0.6250
0.900000	3	0.4930	0.5098	0.6387	**0.6500**
0.900000	4	0.5882	0.4902	0.5000	0.5625

TABLE 17: Per-Class Recall for trees with

Threshold	Mixing Level	No MS Recall	RRMS Recall	SPMS Recall	PPMS Recall
0.500000	0	0.5882	0.6471	0.3529	**0.6875**
0.500000	1	**0.6471**	0.6471	0.5294	0.6250
0.500000	2	0.5294	0.4706	0.4118	0.6250
0.500000	3	0.5882	0.6471	0.2353	0.6250
0.500000	4	**0.6471**	0.5882	0.3529	0.6250
0.700000	0	0.3529	0.4706	0.4118	0.4375
0.700000	1	0.4706	0.5294	0.4118	0.3750
0.700000	2	0.4118	0.4706	0.4118	0.5000
0.700000	3	0.4118	0.4706	0.4118	0.4375
0.700000	4	0.5294	0.4706	0.2941	0.5000
0.800000	0	0.4706	**0.7059**	0.5294	0.3750
0.800000	1	0.4706	0.4118	0.4118	0.3750
0.800000	2	0.5294	0.4118	0.4118	0.4375
0.800000	3	0.5294	0.4118	0.5294	0.3750
0.800000	4	0.5882	0.5294	0.4118	0.4375
0.900000	0	0.3529	0.5294	0.5882	0.6250
0.900000	1	0.3529	0.5294	0.5882	0.6250
0.900000	2	0.3529	0.5294	0.5882	0.6250
0.900000	3	0.4118	0.5294	**0.6471**	**0.6875**
0.900000	4	0.5294	0.4706	0.5882	0.5625

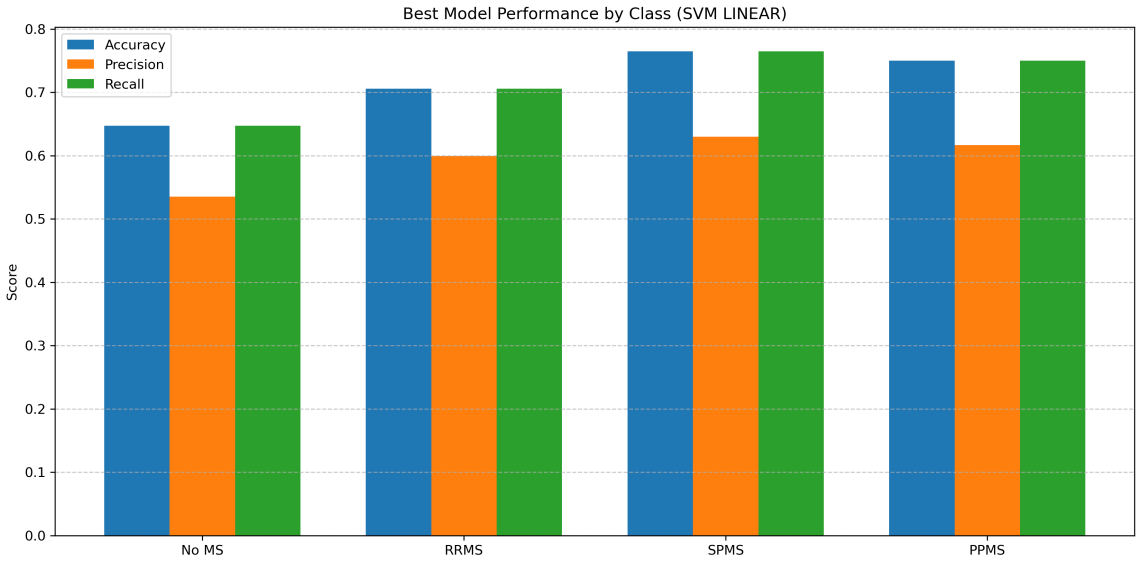

Fig. 10: Performance by Class for Best SVM Model with LINEAR kernel (Threshold: 0.5, Mixing Level: 0.0).

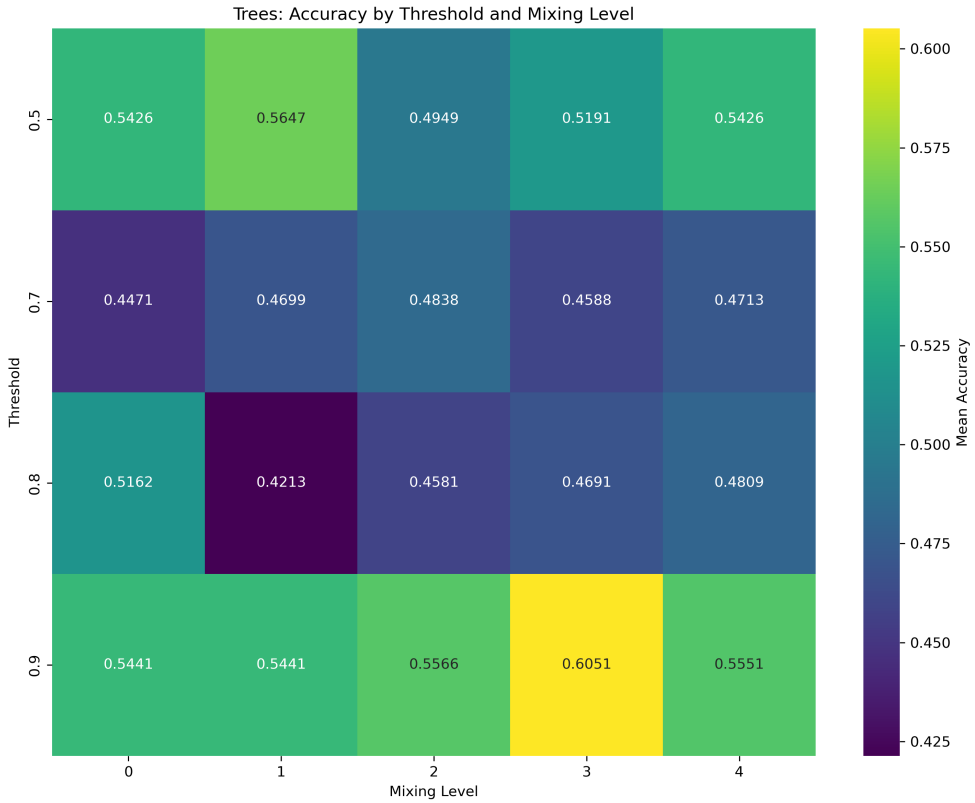

Fig. 11: Heatmap of Mean Accuracy for Trees with

Fig. 12: Performance by Class for Best tree Model (Threshold: 0.9, Mixing Level: 3.0)