Review of Spectral Graph Convolutions for Population-based Disease Prediction

Manal Akhannous
ENS Paris-Saclay
Saclay, France
manal.akhannous@eleves.enpc.fr

Ines Vati
ENS Paris-Saclay
Saclay, France
ines.vati@eleves.enpc.fr

Balthazar Neveu
ENS Paris-Saclay
Saclay, France
balthazar.neveu@ens-paris-saclay.fr

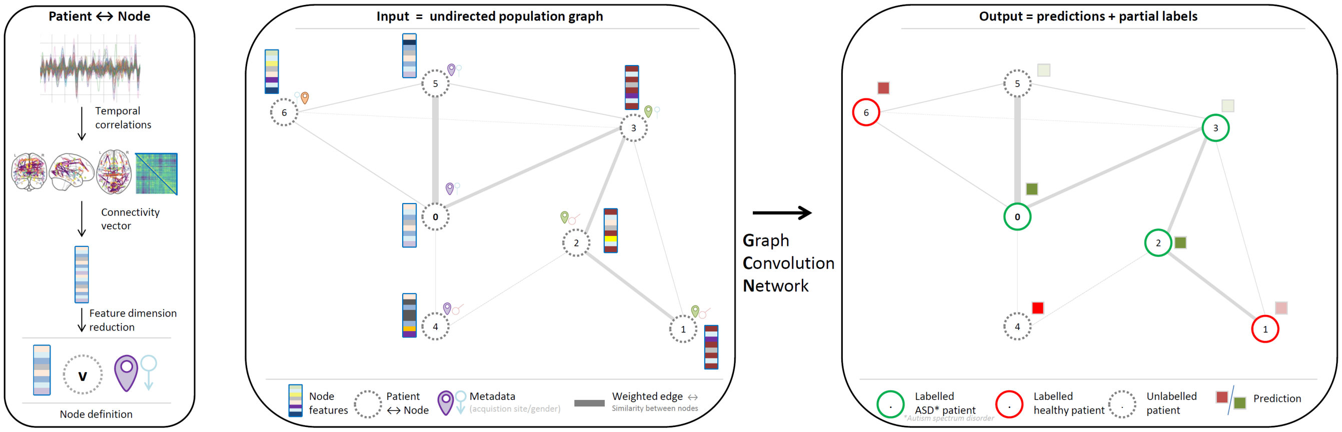

Patient ↔ Node
Temporal correlations
Connectivity vector
Feature dimension reduction
V
Node definition
Input = undirected population graph
6
5
3
0
2
4
1
Node features
Patient ↔ Node
Metadata (acquistion site/gender)
Weighted edge ↔ Similarity between nodes
Graph Convolution Network
Output = predictions + partial labels
6
5
3
0
2
4
1
Labelled ASD* patient
*Autism spectrum disorder
Labelled healthy patient
Unlabelled patient
Prediction

Figure 1: Overview of the use of graph convolutional networks to predict ASD (autism spectrum disorder). The ABIDE dataset was created to study autism and contains a set of functional MRI from 871 patients of different genders and captured over 17 different sites with different f-MRI devices. Input data is scarce and not totally homogeneous. On the left, the creation of the content of a single node is shown. 111 temporal series are extracted from the f-MRI and correlation allows creating a connectivity 111×111 symmetric matrix of the brain of each patient. A population graph is created by connecting each patient (node) to the other patients, with an edge weighted by the similarity between the patients. It is processed by a graph convolutional network to predict the ASD status of each patient. Nodes are partially labelled to split the dataset between training, validation and test set.

ABSTRACT

We propose a review of *Spectral Graph Convolutions for Population-based Disease Prediction* [11], a research paper published in 2017 which introduced the use of Graph Convolutional Networks (GCN) to the domain of brain-related disorders classification in populations. We conducted experiments on the ABIDE dataset to reproduce and challenge the results of the paper. We paid attention to make our experiments reproducible and made our code available on github.

https://github.com/balthazarneveu/spectral_graph_convolutions

KEYWORDS

Graph, Spectral, Convolution, Medical, Disease

1 INTRODUCTION

In recent years, the intersection of machine learning and medical research has seen an increase in innovative approaches to analyse complex patterns in population health data. One such approach is presented in the paper Spectral Graph Convolutions for Population-based Disease Prediction [11] which introduces the use of Graph Convolutional Networks (GCN) for analysing brain-related disorders in populations. By utilizing a graph structure that combines imaging and non-imaging data, this paper makes significant advancements in disease prediction.

The paper addresses the challenges posed by the growing dimensionality of health datasets and the limited availability of labelled instances. By representing populations as graphs, with each subject as a vertex and edges encoding phenotypic information, the authors propose a methodology that leverages auxiliary information to improve disease prediction. The graph structure captures the similarities between subjects, providing a nuanced representation of the population.

The Graph Convolutional Network (GCN) model is trained on partially labelled graphs, enabling it to predict classes for unlabelled nodes based on node features and pairwise associations between subjects. This novel use of graph structures demonstrates improved accuracy in disease prediction compared to traditional linear classifiers. The application of GCN is tested on different databases, showing promising results in brain-related disorders.

The proposed method has applications in various domains of medical research, with a primary focus on disease prediction. Its effectiveness is demonstrated by predicting disease conversion, particularly the transition from Mild Cognitive Impairment (MCI) to Alzheimer's Disease (AD) in the ADNI database. The GCN model exhibits promising accuracy in disease prediction compared to linear classifiers, highlighting its potential clinical relevance.

In addition to disease prediction, the paper addresses a critical issue in medical diagnosis: the reliance on behavioural tests for identifying conditions such as Autism Spectrum Disorder (ASD) or Alzheimer's. Many related diseases are often diagnosed only through subjective observations and behavioural assessments, leading to delayed identification and intervention.

The impact of introducing a new diagnostic tool, such as the GCN-based methodology proposed in the paper, is profound. Early diagnosis enabled by this model offers a transformative opportunity for patients. It allows for timely access to medical treatments and interventions, potentially slowing down disease progression and improving overall outcomes. The ability to identify and predict diseases at earlier stages not only enhances patient care but also opens avenues for targeted therapies and interventions tailored to individual needs.

Furthermore, the model's capacity to integrate various data types into a unified graph structure enables a more comprehensive understanding of the diseases under consideration. Moreover, graph-based methods have an interpretability advantage over traditional deep learning Multi-Layer Perceptrons (MLP). The graph structure inherently captures relationships between subjects, providing a transparent and interpretable framework for understanding the model's decision-making process. This interpretability is crucial in the medical domain, where clinicians and researchers need to trust and comprehend the models' predictions for effective integration into clinical practice.

In summary, the paper introduces an advanced methodology for population-based disease prediction, offering a promising solution to the challenges posed by traditional diagnostic methods. The incorporation of graph-based models not only enhances prediction accuracy but also holds the potential to revolutionize the landscape of early disease diagnosis and intervention, ultimately improving patient outcomes.

2 RELATED WORK

In recent years, several works have been published exploring different approaches for disease prediction.

A prevalent approach in the last few years involved using convolutional neural networks, based solely on the imaging data. Dolz et al. [4] investigated a 3D CNN for subcortical brain structure segmentation in MRI. The proposed model addresses computational challenges using small kernels and embeds intermediate-layer outputs in the final prediction. Khosla et al. [8] evaluated the impact of the chosen brain parcellations on the performance of the 3D CNN model. They proposed an ensemble learning strategy that combines predictions from models trained on connectivity data with different parcellations, improving the classification accuracy. More recently, Sherkatghanad et al. [12] developed a CNN model that uses fewer parameters compared to state-of-the-art models, making it less computationally expensive but showing competitive results in autism disorder detection.

Other researchers, for instance, Kong at al. [10], investigated the use of autoencoders as a way of extracting a few features before using neural networks on these features to perform the classification task for autism disorder detection. ASD-DiagNet, proposed by Eslami et al. [5], integrated a data augmentation strategy to further improve the robustness of this model.

After Parisot et al. [11] introduced graph convolutional networks as an approach for disease prediction, further works proposed some improvements to this approach. The same authors proposed an improved loss function for better generalization, especially for heterogeneous data using metric learning. More recently, Jiang et al. [7] proposed a Hierarchical GCN (hi-GCN) model for learning deep representations from fMRI brain connectivity networks. The model has a two-level GCN architecture, capturing region-to-region brain activity correlations and subject-to-subject relationships simultaneously.

These diverse approaches contribute to the evolving landscape of disease prediction using neural networks, offering insights into novel architectures, optimization strategies, and the integration of multi-modal data for improved accuracy and clinical relevance.

3 PROPOSED METHOD

3.1 Graph construction

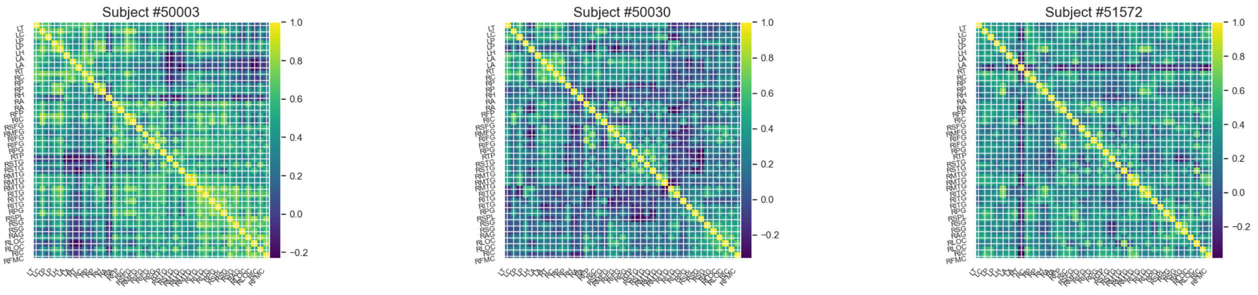
Autism spectrum disorder (ASD) is now recognized to occur in more than 1% of children. Despite continuing research advances, their pace and clinical impact have not kept up with the urgency to identify ways of determining the diagnosis at earlier ages, selecting optimal treatments, and predicting outcomes. The researchers [11] proposed to tackle those challenges and applied their method on the Autism Brain Imaging Data Exchange (ABIDE) dataset, which consists of highly heterogeneous functional MRI data acquired at multiple sites. Resting-state functional magnetic resonance imaging (rs-fMRI) is a neuroimaging technique that measures and analyses brain activity in a resting state, i.e., when a person is not engaged in any specific task. It is based on the principle that different regions of the brain exhibit correlated spontaneous fluctuations in blood oxygen level-dependent (BOLD) signals even in the absence of an explicit stimulus or task. These fluctuations are thought to reflect the intrinsic functional connectivity of the brain.

The authors [11] proposed to construct a population graph integrating imaging and non-imaging data. They defined the feature vector $x(v)$ of each patient v as the vectorized form X_v of its functional connectivity matrix. Connectivity matrices represent the strength of functional connections between different brain regions for a given subject. They are derived from rs-fMRI data by computing the Pearson's correlation coefficient between brain regions' activity patterns, offering a concise representation of interregional communication within brain regions. The figure 2 display the connectivity matrix obtained for three patients in the ABIDE dataset[1].

The connectome encapsulates the intricate web of connections between different brain regions. In figure 3, we depict the brain graph associated with the connectivity matrix computed in figure 2. The coordinate of brain regions are derived from the Harvard-Oxford atlas[2] which defines a mask containing 111 regions of interest (ROIs).

[1] Note that the authors also applied the Fisher transform of the Pearson correlation but we did not do it.

[2] http://preprocessed-connectomes-project.org/abide/Pipelines.html



(a) Patient with ASD in PITT acquisition site

(b) Patient without ASD in PITT acquisition site

(c) Patient with ASD in SBL acquisition site

Figure 2: Connectivity matrix of given subjects. Each brain region is indicated in the x-axis and y-axis. The colour of the pixel indicates the strength of the functional connection between the two regions. We did not plot all 111 regions for the sake of readability.

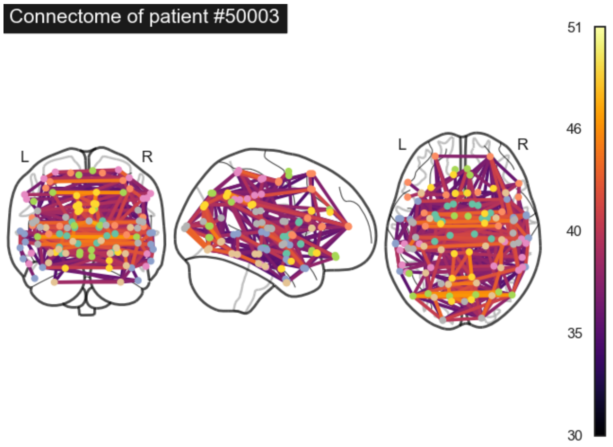


Figure 3: Connectome graph of a given patient. Each node represents a specific brain region and edges correspond to the connectivity between two regions as plotted in figure 2. This representation offers a unique insight into the patient's neural architecture, highlighting the potential correlates of cognitive processes, as well as providing valuable information for understanding the impact of neurological conditions. Note that the exact values of the edges weights were multiplied by 50 to improve the readability of the figure.

Given the matrix's high dimensionality (number of patients ×6217), the authors employed a ridge classifier to identify the most discriminative features within the training set.

They modelled the interactions between individuals via the definition of the graph edges, $\mathcal{E}$ which writes as follows for two nodes v and v' :

$$\mathbf{W}(v,v') = Sim(X_v, X_{v'}) \left[\delta_{M_{\text{SEX}}(v)}(M_{\text{SEX}}(v')) + \delta_{M_{\text{SITE}}(v)}(M_{\text{SITE}}(v'))\right]$$

δ is the Kronecker delta function. M_{SITE} and M_{SEX} are contextual information. *Sim* is a similarity measure between the connectivity matrices of the subjects v and v'. In [11], the authors used the correlation distance

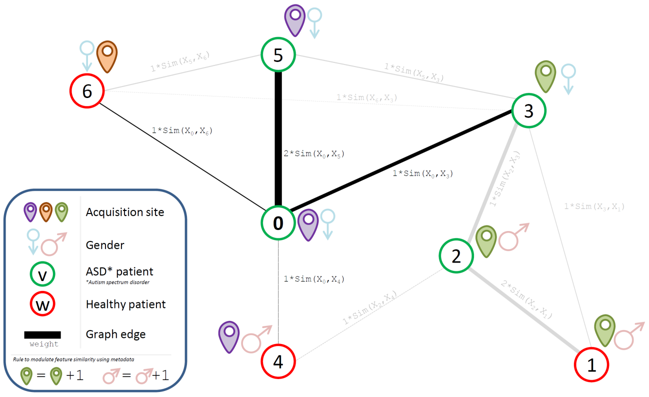


Figure 4: Graph edge construction process by modulating similarity between features by phenotypic information (gender and acquisition site).

$$Sim(X_v, X_{v'}) = 1 - \frac{(X_v - \bar{X}_v).(X_{v'} - \bar{X}_{v'})}{\left\|X_v - \bar{X}_v\right\|_2 \left\|X_{v'} - \bar{X}_{v'}\right\|_2}$$

where $\bar{X}_v$ is the mean of the elements of the vectorized connectivity matrix of the patient v.

On the figure 5, certain sets or squares stand out with significantly higher correlation along the diagonal, demonstrating the similarity between this metadata information.

To classify from this brain population graph, the authors [11] proposed to use a Graph Convolutional Neural Network (GCN) on graphs. We will now present the mathematical framework of these networks.

3.2 Mathematical framework on Graph Spectral Theory

3.2.1 The main challenges of signal processing on graphs. The generic data representation forms of graphs are useful for describing the

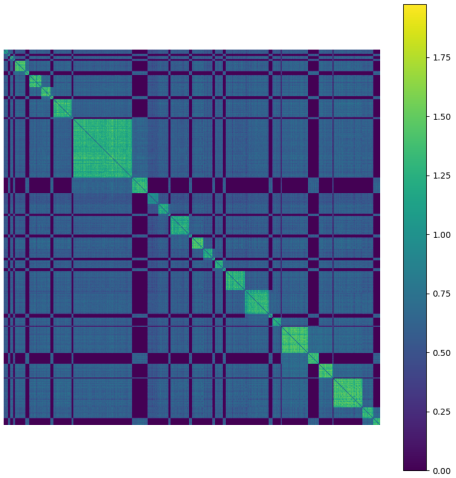


Figure 5: Edge weights sorted by *acquisition sites* and *gender*. This plot explains the process of using contextual and phenotypic information to construct connections in the graph. The fundamental concept underlying this graph structure is to exploit site information, anticipating greater comparability among subjects within the same site owing to diverse acquisition protocols.

geometric structures of data domains in very different application fields. However, classical tools and techniques from signal processing can not be used seamlessly on graphs, unlike audio signal and images [13].

In particular, graphs have no inherent ordering of the vertices, unlike images where each pixel is uniquely identified by its position within the image. We therefore need algorithms that are node-order equivariant: they should not depend on the ordering of the nodes [2]. We may also need localized transforms that compute information about the data at each vertex by using data from a small neighbourhood of vertices close to it in the graph.

Graphs can also be very large but, in our study, we consider sparse population graphs, i.e. individuals or nodes are connected to a limited number of nodes [3]. Spectral graph theory has enabled constructing, analysing, and manipulating graphs.

In this section, we present some basic definitions from spectral graph theory that will be needed to apply neural networks on graphs. As stated previously, we consider an undirected, connected, weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$.

3.2.2 The non-normalized and normalized Graph Laplacian. The non-normalized graph Laplacian, also called the combinatorial graph Laplacian, is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where the degree matrix $\mathbf{D}$ is a diagonal matrix whose ith diagonal element, d_i, is equal to the sum of the weights of all edges incident to the vertex i (e.g. the sum over the rows of $\mathbf{W}$).

The graph Laplacian $\mathbf{L}$ is a real symmetric matrix, it has therefore real, non-negative eigenvalues $\{\lambda_l\}_{l=0,\ldots,N-1}$. We denote their associated orthonormal eigenvectors by $\{u_l\}_{l=0,\ldots,N-1}$. In the following, Λ and U denote the diagonal matrix of eigenvalues and the matrix of eigenvectors, respectively.

Since we consider connected graphs, the eigenvalue $\lambda_0 = 0$ has multiplicity 1 [13].

A popular practice is to normalize each weight $W_{i,j}$ by a factor of $\frac{1}{\sqrt{d_i d_j}}$. Doing so leads to the normalized graph Laplacian, which is defined as $\widetilde{\mathbf{L}} = D^{-1/2}\mathbf{L}D^{-1/2} = I_N - D^{-1/2}\mathbf{W}D^{-1/2}$. This strategy is used to improve the stability of gradient flow during backpropagation in neural networks. This can be valuable in mitigating the vanishing gradient problem and facilitating the training of deeper and more expressive models on graph data.

3.2.3 A Graph Fourier Transform and Notion of Frequency. The Fourier transform for an analogous function f

$$\hat{f}(\xi) = \left\langle f \middle| e^{2\pi i \xi t} \right\rangle = \int_{\mathbb{R}} f(t) e^{-2\pi i \xi t} dt$$

is the expansion of a function f in terms of the complex exponentials, which are the eigenfunctions of the one-dimensional Laplace operator Δ :

$$-\Delta(e^{2\pi i \xi t}) = -\frac{\partial^2}{\partial t^2} e^{2\pi i \xi t} = (2\pi i \xi)^2 e^{2\pi i \xi t}$$

Similarly, we can define the *Graph Fourier Transform* $\hat{f}$ of any function on the vertices of $\mathcal{G}$ as the expansion of f in terms of the eigenvectors of the graph Laplacian $\hat{f}(\lambda_l) = \langle f | u_l \rangle = \sum_{i=1}^{N} f(i) u_l^*(i)$: where $u_l^*(i)$ is the conjugate of $u_l(i)$:

$$\hat{f} = U^T f \tag{1}$$

The *inverse graph Fourier transform* is then given by $f(i) = \sum_{i=1}^{N} \hat{f}(\lambda_i) u_l(i)$:

$$f = U\hat{f} \tag{2}$$

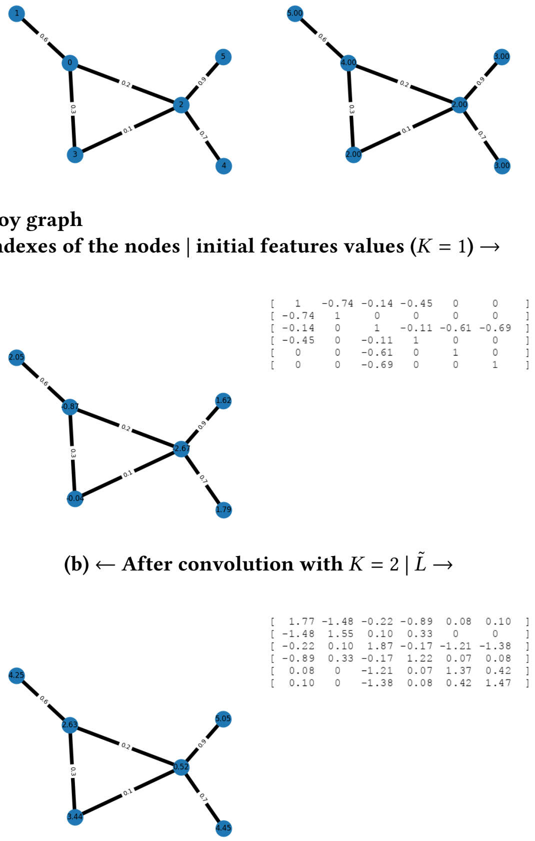
Note that, in our case, the signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$ associates a feature vector to each node of the graph.

3.2.4 Spectral Graph Convolutions. Spectral convolution of the signal f with a filter g_θ is defined as $g_\theta * f = g_\theta(\widetilde{\mathbf{L}}) f = U g_\theta(\Lambda) U^T f = U g_\theta(\Lambda) \hat{f}$. Indeed, we recover the property that the convolution in the vertex domain is equivalent to a multiplication in the graph spectral domain.

Polynomial filters, defined as $g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k$, are commonly used in localized convolutions. Those filters are localized thanks to the lemma 5.2 in [6]. We can show that for some integer s and for any vertices m and n in the graph $\mathcal{G}$, if $d_{\mathcal{G}}(m,n) > s$ then $(L^s)_{m,n} = 0$. $d_{\mathcal{G}}(m,n)$ denotes the number of edges in the shortest path connecting the nodes m and n. It follows that a K-order polynomial filter is strictly K-localized, i.e. it only depends on the K-hop neighbourhood of each vertex. In the studied paper [11], the authors chose $K = 3$.

Furthermore, according to the lemma 1, such filters can be well approximated by a truncated Chebyshev expansion of the form

[3] We also say that the number of edges is linear in the number of nodes.



(a) Toy graph
← indexes of the nodes | initial features values ($K = 1$) →

```
[   1    -0.74 -0.14 -0.45   0     0    ]
[ -0.74   1     0     0      0     0    ]
[ -0.14   0     1    -0.11 -0.61 -0.69  ]
[ -0.45   0    -0.11  1      0     0    ]
[   0     0    -0.61  0      1     0    ]
[   0     0    -0.69  0      0     1    ]
```

(b) ← After convolution with $K = 2 \mid \tilde{L}$ →

```
[  1.77 -1.48 -0.22 -0.89  0.08  0.10  ]
[ -1.48  1.55  0.10  0.33   0     0    ]
[ -0.22  0.10  1.87 -0.17 -1.21 -1.38  ]
[ -0.89  0.33 -0.17  1.22  0.07  0.08  ]
[  0.08   0   -1.21  0.07  1.37  0.42  ]
[  0.10   0   -1.38  0.08  0.42  1.47  ]
```

(c) Convolution with $K = 3 \mid \tilde{L}^2$ →

Figure 6: Convolution of a toy graph and the normalized Laplacian matrix $\widetilde{\mathbf{L}}^K$ for $K = 1, 2, 3$.

$g_\theta(\widetilde{\mathbf{L}}) = \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{\mathbf{L}})$ where T_k is the kth Chebyshev polynomial of the first kind.

Lemma 1. *In the appropriate Sobolev space, the set of Chebyshev polynomials form an orthonormal basis, so that a function in the same space can, on $-1 \leq x \leq 1$, be expressed via the expansion :*

$$g(x) = \sum_{n=0}^{\infty} a_n T_n(x)$$

.

This decomposition significantly reduces the computational complexity of the convolution operator [11].

3.2.5 A simple illustration. We have implemented a simple application of the spectral graph convolution on a toy graph. On the illustration 6, we notice that the greater the order of the convolution, the more the signal is smoothed, and the less the graph Laplacian have null coefficients.

For the example displayed in figure 6, we initialized the weight of the convolution to one.

4 EXPERIMENTS

4.1 Proposed Modifications

We studied various aspects of the methodology, including:

(1) The relevance of dimensionality reduction applied to features before training the neural networks.
(2) The approach of choosing a graph-based classification approach rather than a simple dense network operating on the vectorized feature vectors.

As a result, we explored different techniques for dimensionality reduction and various classification models. We run the following experiments to address these questions.

Initially, we trained the models using the original features without any dimensionality reduction (1). Subsequently, we trained them with a reduced number of features achieved through recursive feature elimination (2), mirroring the approach detailed in the referenced paper. Finally, we applied an autoencoder to reduce feature dimensionality (3).

We implemented three distinct models:

(1) A simple Dense Neural Network consisting of linear layers and ReLU activations.
(2) A Graph Convolutional Network (GCN) as outlined in [9], incorporating the *renormalization trick*.
(3) A neural network utilizing Chebyshev polynomials with an order of $K = 3$, in accordance with the methodology presented in the referenced paper [11].

4.2 Dimensionality reduction by Recursive Feature Elimination with a Ridge classifier

A Ridge classifier is a standard linear classifier with an added regularization term known as the Ridge (L2) penalty.

The optimization problem of the ridge classifier is to find the coefficients β that minimize the following objective function:

$$\min_{\beta} \|y - X\beta\|_2^2 + \alpha \|\beta\|_2^2$$

where X is the design matrix, here the vectorized connectivity matrix, y is the corresponding target value, α is the regularization parameter, and $\|.\|_2$ denotes the Euclidean norm. The ridge classifier converts the targets values into $\{-1, 1\}$ and then treats the problem as a regression task. The predicted class corresponds to the sign of the regressor's prediction, which is

$$\hat{y} = sign(X\hat{\beta})$$

where *sign* is a function that return 1 if the input is positive, -1 if the input is negative, and 0 if the input is zero.

The recursive feature elimination (RFE) is employed to iteratively select features by progressively considering smaller sets of features. Initially, the estimator is trained on the complete set of features, and the importance of each feature i is determined by evaluating the weights assigned to the Ridge estimator $\hat{\beta}_i$. Subsequently, the least important features are pruned from the current set and this process is repeated recursively on the pruned set until the desired number of features to select is reached.

4.3 Dimensionality reduction by Autoencoder

Autoencoders are a type of neural network architecture widely employed for dimensionality reduction in machine learning applications. When applied to feature selection, autoencoders aim to learn a compressed representation of the input data, effectively and automatically capturing its essential features.

We tested these autoencoders against Recursive Feature Elimination (RFE) and using all input features. We implemented simple autoencoders, the encoder and the decoder contain one linear layer each, and the bottleneck has the same dimension as the number of features to select used in the RFE, finally, we chose a tanh function as an activation function between these layers.

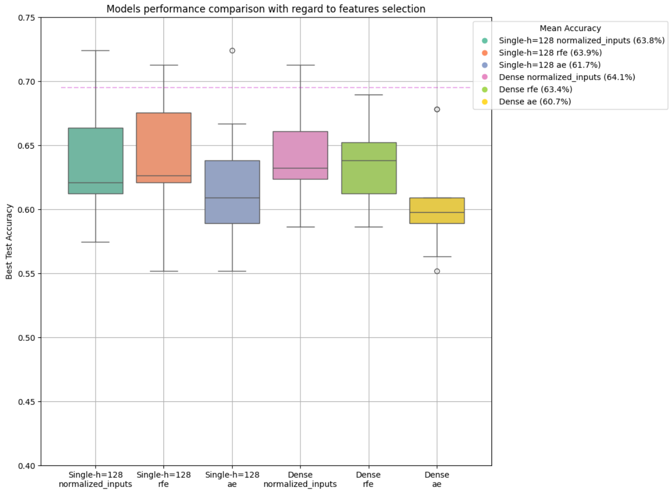


Figure 7: Comparison of dimensionality reduction methods - including using autoencoders.

The results obtained are reported in the figure 7. While autoencoders provide close accuracies, they fall slightly short of the performance achieved by the other methods. This outcome suggests that, in the context of disease prediction classifier, RFE and utilizing the complete set of input features may yield slightly better results. Understanding the trade-offs between these techniques is crucial in selecting an appropriate dimensionality reduction strategy that aligns with the specific characteristics and requirements of your dataset and prediction task.

5 RESULTS AND DISCUSSION

5.1 Experimental setting

In order to compare several methods (architecture, feature selection), we train on the ABIDE dataset with the systematic method of cross-validation proposed below.

- We create 10 splits of the dataset with fixed seeds. Seeds are the same across all experiments.
- Each split is made of a training (80%), validation (10%) and test set (10%).
- At training time, the whole population is present in the graph but only a fraction of the nodes are labeled (the 80% of the training set).
- For each experiment configuration, we train 10 times the model with each split. We pick the epoch which corresponds to the minimum validation loss and take the corresponding test accuracy at this specific epoch. Mean and standard deviation of the accuracy across all splits.

We abbreviate the tested configurations as follows:

- Single-h=h refers to a single hidden layer with around 6000 features or 2000 when reduced, h hidden neurons and a single output for binary classification.
- Dense refers to a fully connected neural network with 2 hidden layers and a final linear classifier.
- Cheb-dr=dr refers to the Chebyshev spectral graph convolutional [3] architecture with a dropout rate of dr.

We systematically used the Adam optimizer with a learning rate of 10^{-4} and a weight decay of 0.1 and train for 1000 epochs. The majority of models overfit the training dataset. In all following boxplots, we report the original paper's accuracy (69.5%) with a dashed purple line.

5.2 Results on using reduced features

In line with the methodology outlined in the original paper, we systematically selected a subset of 2000 features with the RFE algorithm trained on a subset of 300 samples.

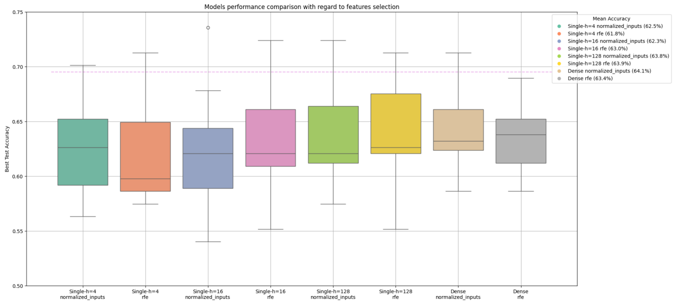


Figure 8: No significant differences in performance between using all features and using a subset of 2000 features selected with RFE, no matter the number of neurons in the hidden layer.

Despite the varied configurations tested, it is notable that these setups did not result in significant differences in performance on the test data. Furthermore, the reduction of features through Recursive Feature Elimination (RFE) did not demonstrate any discernible improvement in performance. Table 1 summarizes the results of our experiments on feature dimensionality reduction.

5.3 Effect of using graph convolution

The boxplot 9 displays the performance of the various configurations in our study. We observe that the use of graph convolutional layers improves the performance of the model as claimed by the authors of the original paper. All average performances are reported in 2.

Model	Feature	Test accuracy
Single-h=4	normalized inputs	62.5 +/- 4.2%
Single-h=4	rfe	61.8 +/- 4.4%
Single-h=4	ae	59.8 +/- 8.6%
Single-h=16	normalized inputs	62.3 +/- 5.4%
Single-h=16	rfe	63.0 +/- 5.1%
Single-h=16	ae	60.3 +/- 3.9%
Single-h=128	normalized inputs	63.8 +/- 4.3%
Single-h=128	rfe	63.9 +/- 4.6%
Single-h=128	ae	61.7 +/- 4.8%
Dense	normalized inputs	64.1 +/- 3.4%
Dense	rfe	63.4 +/- 3.3%
Dense	ae	60.7 +/- 4.0%

Table 1: Models performances with regard to feature reduction method. There is no obvious advantage to using a feature reduction method.

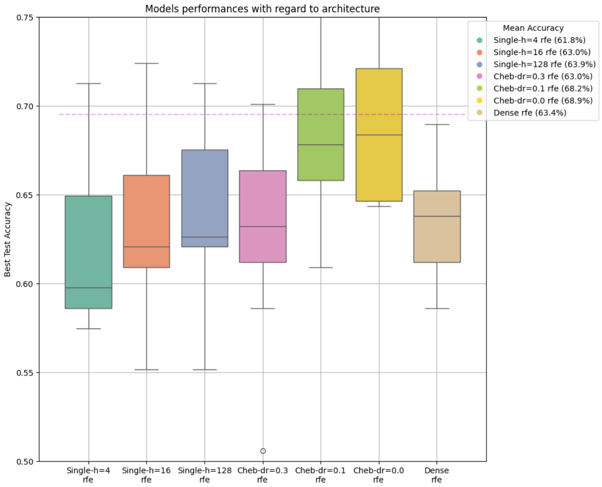


Figure 9: Classification accuracy: comparison of different architectures. Single architectures are composed of a single hidden layer with a given number of neurons h. *Dense* architecture refer to a fully connected neural network with a single hidden layer. *Cheb* architectures refer to the graph convolution architecture, we vary the dropout rate dr during training.

Model	Feature	Test accuracy
Single-h=4	rfe	61.8 +/- 4.4%
Single-h=16	rfe	63.0 +/- 5.1%
Single-h=128	rfe	63.9 +/- 4.6%
Cheb-dr=0.3	rfe	63.0 +/- 5.4%
Cheb-dr=0.1	rfe	68.2 +/- 4.0%
Cheb-dr=0.0	rfe	68.9 +/- 4.0%
Dense	rfe	63.4 +/- 3.3%

Table 2: Models performances with regard to architecture

6 MAIN LIMITATIONS AND CONCLUSION

One prominent challenge that emerged from our experiments was the significant discrepancy between the number of subjects (871) and the number of features (higher than 6100), even after dimension reduction techniques were applied. This scenario places us squarely in the realm of the "curse of dimensionality," introducing complexities that can adversely impact the training of neural network models. With a limited number of subjects, the data points within the high-dimensional feature space become sparsely distributed. This sparsity poses challenges for neural networks, as they may struggle to capture meaningful patterns and relationships in the data, leading to overfitting or suboptimal generalization.

In particular, all the models were overfitting the training data, as evidenced by the large gap between the training (100% accuracy) and validation accuracies (65%). Despite our attempts to mitigate this issue (by applying dropout and L_2 regularization), neural network models always end up overfitting the training dataset. No matter how high the training accuracy gets, validation accuracy keeps increasing and we observe no degradation in the performances. This could be an instance of benign overfitting [1]. We note that the ChebGCN trained without dropout yields the best performance on the validation data 68.9 +/- 4.0%, still not reaching the 69.5% accuracy mentioned in the original paper.

In conclusion, our experiments encompassed a range of configurations, each designed to explore different aspects of the model. Despite the diversity in hyperparameters and settings, the performance across these configurations on the test data exhibited minimal variations.

REFERENCES

[1] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. 2020. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences* 117, 48 (April 2020), 30063–30070. https://doi.org/10.1073/pnas.1907378117

[2] Ameya Daigavane, Balaraman Ravindran, and Gaurav Aggarwal. [n. d.]. Understanding Convolutions on Graphs. 6, 9 ([n. d.]), e32. https://doi.org/10.23915/distill.00032

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. (6 2016). http://arxiv.org/abs/1606.09375

[4] Jose Dolz, Christian Desrosiers, and Ismail Ben Ayed. 2018. 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage* 170 (4 2018), 456–470. https://doi.org/10.1016/j.neuroimage.2017.04.039

[5] Taban Eslami, Vahid Mirjalili, Alvis Fong, Angela R. Laird, and Fahad Saeed. 2019. ASD-DiagNet: A Hybrid Learning Approach for Detection of Autism Spectrum Disorder Using fMRI Data. *Frontiers in Neuroinformatics* 13 (11 2019). https://doi.org/10.3389/fninf.2019.00070

[6] David K. Hammond, Pierre Vandergheynst, and RÃ©mi Gribonval. [n. d.]. Wavelets on graphs via spectral graph theory. 30, 2 ([n. d.]), 129–150. https://doi.org/10.1016/j.acha.2010.04.005

[7] Hao Jiang, Peng Cao, Ming Yi Xu, Jinzhu Yang, and Osmar Zaiane. 2020. Hi-GCN: A hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction. *Computers in Biology and Medicine* 127 (12 2020). https://doi.org/10.1016/j.compbiomed.2020.104096

[8] Meenakshi Khosla, Keith Jamison, Amy Kuceyeski, and Mert R. Sabuncu. 2019. Ensemble learning with 3D convolutional neural networks for functional connectome-based prediction. *NeuroImage* 199 (10 2019), 651–662. https://doi.org/10.1016/j.neuroimage.2019.06.012

[9] Thomas N. Kipf and Max Welling. [n. d.]. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs, stat] http://arxiv.org/abs/1609.02907

[10] Yazhou Kong, Jianliang Gao, Yunpei Xu, Yi Pan, Jianxin Wang, and Jin Liu. 2019. Classification of autism spectrum disorder by combining brain connectivity and deep neural network classifier. *Neurocomputing* 324 (1 2019), 63–68. https://doi.org/10.1016/j.neucom.2018.04.080

[11] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrerro Moreno, Ben Glocker, and Daniel Rueckert. 2017. *"Spectral Graph Convolutions for Population-Based Disease Prediction"*.

[12] Zeinab Sherkatghanad, Mohammadsadegh Akhondzadeh, Soorena Salari, Mariam Zomorodi-Moghadam, Moloud Abdar, U. Rajendra Acharya, Reza Khosrowabadi, and Vahid Salari. 2020. Automated Detection of Autism Spectrum Disorder Using a Convolutional Neural Network. *Frontiers in Neuroscience* 13 (1 2020). https://doi.org/10.3389/fnins.2019.01325

[13] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. [n. d.]. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. 30, 3 ([n. d.]), 83–98. https://doi.org/10.1109/MSP.2012.2235192 arXiv:1211.0053 [cs]

Received 10 December 2023; revised 10 December 2023; accepted 10 December 2023