UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

GRAPH NEURAL NETWORKS FOR SEIZURE DISCRIMINATION BASED ON ELECTROENCEPHALOGRAM ANALYSIS

PABLO JOSÉ GALVÁN CALDERÓN

Thesis supervisor: SERGI ABADAL CAVALLÉ (Department of Computer Architecture)

Degree: Bachelor Degree in Informatics Engineering (Computing)

Thesis report

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

27/06/2023

Index

List of figures

List of tables

I want to thank Sergi Abadal for directing this thesis and making it possible, as well as the other members of BNN and other students I collaborated with during the study, such as Dimas Ávila Martínez, for providing the base code for the project, and Alberto Mármol. This would not have been possible either without the researchers at Mediterranea University of Reggio Calabria, who provided the dataset used for the project and gave additional guidance. Finally, I want to thank those who have always supported me and always motivate me to keep going: my parents, who have always guided me through the right path and raised me with endless care, and An, my partner in life and childhood friend, who will never fail to put a smile on my face.

Abstract

This study presents a research investigation on the classification of Psychogenic Non-Epileptic Seizures (PNES) and Epileptic Seizures (ES) using EEG data and Graph Neural Networks (GNN). The proposed model demonstrates outstanding performance, surpassing previous state-of-the-art results and achieving remarkable accuracy in ternary classification.

By utilizing a GNN architecture, the model effectively distinguishes between PNES and ES with an accuracy of 92.9%. Moreover, when employing Leave One Group Out cross-validation, the model achieves an even higher accuracy of 97.58%, outperforming the highest reported state-of-the-art accuracy of 94.4%. Furthermore, by extending the classification to include healthy patients, the model achieves an accuracy of 91.12%, surpassing the best-known state-of-the-art accuracy of 85.7%.

These findings highlight the potential of the model in accurately classifying and differentiating these medical conditions using EEG data. Future work includes the exploration of biomarkers for binary classification using the model's explainability capabilities, contributing to the development of objective diagnostic tools and personalized treatment strategies. Additionally, this study compares the performance, methodologies, and datasets of similar studies from the state-of-the-art, providing a comprehensive overview of seizure classification research.

In conclusion, this study demonstrates the success of the proposed model in classifying PNES and ES, paving the way for further advancements in the field and benefiting patients and healthcare practitioners in diagnosis and treatment.

Resumen

Este estudio presenta una investigación sobre la clasificación de Convulsiones Psicógenas No Epilépticas (PNES) y Convulsiones Epilépticas (ES) utilizando datos de EEG y Redes Neuronales de Grafos (GNN). El modelo propuesto muestra un rendimiento destacable, superando los resultados previos del estado del arte y logrando una precisión notable en la clasificación ternaria.

Mediante el uso de una arquitectura GNN, el modelo distingue de manera efectiva entre PNES y ES con una precisión del 92.9%. Además, al emplear la validación cruzada "Leave One Group Out", el modelo logra una precisión aún mayor del 97.58%, superando la precisión más alta reportada en el estado del arte de 94.4%. Asimismo, al ampliar la clasificación para incluir a pacientes sanos, el modelo alcanza una precisión del 91.12%, superando la mejor precisión conocida del estado del arte de 85.7%.

Estos hallazgos resaltan el potencial del modelo para clasificar y diferenciar de manera precisa estas condiciones médicas utilizando datos de EEG. El trabajo futuro incluye la exploración de biomarcadores para la clasificación binaria utilizando las capacidades de explicabilidad del modelo, contribuyendo al desarrollo de herramientas de diagnóstico objetivas y estrategias de tratamiento personalizadas. Además, este estudio compara el rendimiento, las metodologías y los conjuntos de datos de estudios similares del estado del arte, proporcionando una visión general completa de la investigación en clasificación de convulsiones.

En conclusión, este estudio demuestra el éxito del modelo propuesto en la clasificación de PNES y ES, allanando el camino para futuros avances en el campo y beneficiando a pacientes y profesionales de la salud en el diagnóstico y tratamiento.

Resum

Aquest estudi presenta una investigació sobre la classificació de Convulsions Psicogèniques No Epilèptiques (PNES) i Convulsions Epilèptiques (ES) utilitzant dades d'EEG i Xarxes Neuronals de Grafs (GNN). El model proposat demostra un rendiment destacable, superant resultats anteriors de l'estat de l'art i aconseguint una precisió remarcable en la classificació ternària.

Mitjançant l'ús d'una arquitectura GNN, el model distingeix de manera efectiva entre PNES i ES amb una precisió del 92.9%. A més, en emprar la validació creuada "Leave One Group Out", el model aconsegueix una precisió encara més alta del 97.58%, superant la precisió més alta reportada en l'estat de l'art del 94.4%. A més, en ampliar la classificació per incloure pacients sans, el model arriba a una precisió del 91.12%, superant la millor precisió coneguda de l'estat de l'art del 85.7%.

Aquests resultats posen de manifest el potencial del model per classificar i diferenciar de manera precisa aquestes condicions mèdiques utilitzant dades d'EEG. El treball futur inclou l'exploració de biomarcadors per a la classificació binària utilitzant les capacitats d'explicabilitat del model, contribuint al desenvolupament d'eines de diagnòstic objectives i estratègies de tractament personalitzades. A més, aquest estudi compara el rendiment, les metodologies i els conjunts de dades d'estudis similars de l'estat de l'art, proporcionant una visió general completa de la investigació en classificació de crisis.

En conclusió, aquest estudi demostra l'èxit del model proposat en la classificació de PNES i ES, obrint camí per a futurs avenços en el camp i beneficiant pacients i professionals de la salut en el diagnòstic i tractament.

1. Context and scope

1.1. Introduction

1.1.1 Context

According to the National Institute of Neurological Disorders and Stroke (NINDS), an estimated **5 to 20 %** of people diagnosed with epilepsy actually have non-epileptic seizures (NES), which can outwardly resemble epileptic seizures, but are not associated with seizure-like electrical discharge in the brain. Non-epileptic events may be referred to as psychogenic non-epileptic seizures (PNES) [1].

The issue with this similarity in terms of symptoms is that these types of seizures require radically different treatments, as epileptic patients respond well to medical drug treatment, whereas subjects with PNES do not, since this illness is of psychiatric nature and benefits from therapy instead [1]. In addition, it is estimated that around **50 million people** have epilepsy in the world, and it is one of the most common neurological diseases globally. In addition, approximately **80%** of people with epilepsy live in less developed countries, and up to **70%** of people with epilepsy could be cured from it if properly diagnosed and treated [2].

That is why this project aims to improve the accuracy when diagnosing these illnesses and to make it faster and easier to prevent seizures. This project belongs to a Bachelor Thesis of the Degree in Informatics Engineering from the Barcelona School of Informatics at the Universitat Politècnica de Catalunya, of its Computer Science specialization, and it is directed by Sergi Abadal Cavallé, doctorate in Computer Architecture.

Nowadays, the main methods used to identify cognitive diseases such as epilepsy use electroencephalograms (EEGs), since it is the safest and cheapest procedure to analyze brain signals properly. However, these diseases usually need to be identified by human experts, which may not only be specific and difficult to find, but are also subject to human error, which might be problematic due to the delicate nature of these problems and how difficult it can be to differentiate epileptic seizures from psychogenic non-epileptic seizures.

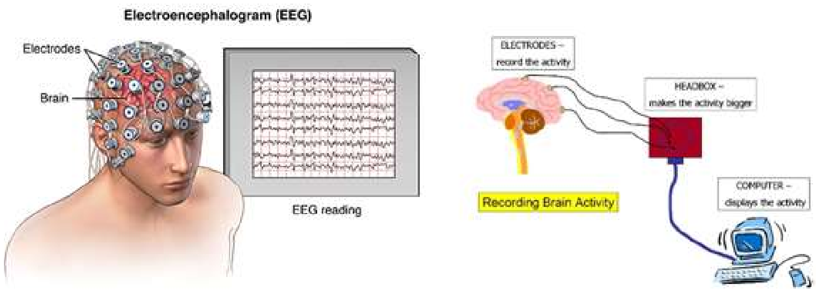


Figure 1: procedure of EEG recording. [3]

In addition, despite neural networks being used sometimes to identify these diseases, CNNs are the most common type of model used for this, but that may not take advantage of the fact that the different parts of the brain are not necessarily correlated by distance, but instead have arbitrary relations which are not properly captured by CNNs.

For those reasons, this project proposes the use of GNNs to identify these types of cognitive diseases by analyzing EEGs from patients as graphs and classifying them according to their illness.

1.1.2 Stakeholders

On the one hand, the stakeholders who will directly influence the project are the tutor, Sergi Abadal Cavallé, who will guide and lead the project, and the researcher Pablo José Galván Calderón.

On the other hand, there are some groups who will not have any direct interactions with the project, but they might be socially affected by it. These are people who suffer (or believe they might suffer) from the diseases covered by the research (since the final model might help them identify their illness), medical centers and the research community around this topic (these two groups might be able to use this model for their work).

In addition, something that will directly affect any applications of this project is the legislation of each country. Many countries do not allow software to be used for medical purposes without human supervision, and most feature certain regulations the software must fulfill to be used in such fields, such as EU countries [4].

Lastly, some parties could be economically benefited from the results of this thesis, such as insurance companies (who would reduce the cost of diagnosis), companies or start-ups dedicated to machine learning or data science (who could provide their services to medical centers by diagnosing the researched diseases with the resulting model), companies selling equipment for recording EEGs, etc.

1.2. Motivation

1.2.1 Previous studies

Several studies regarding the identification of ES or PNES via machine learning exist,with methods like CNNs. That is the case of the study by [5], for example which also uses EEGs to classify epileptic seizures, PNES and healthy control subjects, and it will be one of the main references for the performance of the proposed model.

In addition, some studies also research the use of GNNs to improve the accuracy of predictions on EEGs, like the work by [6], which studies the diagnosis of Alzheimer's disease, MCI and healthy control subjects with GNNs. This study will be the starting point of the project, as its resulting model and its code will be used as a reference for the thesis' model.

1.2.2 Justification

Firstly, as mentioned before, CNNs do not take advantage of the arbitrary relations between the different parts of the brain, which leaves significant potential unused, since the use of CNNs does not only cause a decreased accuracy, but they also allow for more explainability, since it is easier to understand the underlying functioning of a model if graphs are used to model it (they are able to not only give an answer, but also to justify it), making them more suitable for this problem.

Secondly, the work by [6] does take advantage of the benefits of GNNs, but it is a different use case, and it is believed that this study and its functionalities could be applied to the differentiation of seizures, as the EEG analysis procedure would be the same for the larger part.

Thirdly, EEGs have proven to be a reliable method to analyze cerebral activity, since they do not require surgery, which makes them a lot less invasive than other methods who do [3][7]. In addition to that, the existing literature on PNES and ES diagnosis (seen in Section 3) appears to prove that a proper analysis of these can yield great results. For this reason, EEGs were taken as input in this project.

Finally, this project aims to use GNNs to provide more explainability to the solutions provided by the model than those delivered by other architectures, such as CNNs, which allow for a more limited expliainability due to their nature.

1.3. Scope

1.3.1 Objectives and sub-objectives

As specified before, the main goal this project aims for is to create a GNN model which properly discriminates ES and PNES, as well as healthy control subjects.

This objective is greatly simplified once it is broken down into several sub-objectives, which we will divide into two categories: theoretical objectives and practical objectives.

Firstly, the theoretical objectives will be:

- **Project management**: this objective involves organizing and coordinating the various tasks and resources required to complete the project. It includes setting timelines and goals, allocating resources, managing budgets, and communicating with stakeholders.
- **State-of-the-art research:** this step is crucial, since understanding how GNNs and other alternatives (such as CNNs and SVMs) work and how they can be optimized is important to be able to create a proper model for this task, especially since this can help identify gaps in the existing literature and learn from techniques used by other work.

Secondly, the practical goals will be:

- **Replicating results by existing model:** a GNN project which diagnoses mental illnesses through EEGs will be used as a basis for the creation of our model. This step will not only help grasping how GNNs work and the way they are implemented, but it will also make it possible to find models which achieve the main goal to an extent so that this project can modify them, improve their accuracy, and adapt them to the objectives of this thesis.
- **Expanding work for epilepsy and PNES:** once the former sub-objectives are accomplished, the model will be adapted to identify patients with epilepsy and PNES by making the required changes and finding proper datasets. This objective will be considered to be complete if the results found in the state of the art are surpassed.
- **Improving model:** to improve the accuracy of the already existing models, datasets will be required to train the model so that it can learn to properly identify each illness. Different models will be acquired through training until one with particularly good accuracy is found. In addition, the explainability potential of GNNs will be exploited to gain greater insight into how the resulting model predicts its results.
- **Documentation and defense**: this objective involves documenting the work done throughout the project and preparing a defense of the results. This will include writing the memory for the thesis, creating visualizations or other explanatory materials, and presenting the work to others in the field.

1.3.2 Requirements

For the thesis to be fully successful, the accuracy of the model should surpass that of studies using other machine learning models for the same goal, which will be identified during the state-of-the-art research.

In addition, the network should be able to provide a justification for the given result thanks to the GNN architecture.

1.3.3 Potential obstacles and risks

Besides the objectives and requirements of this project, it is also very relevant to keep into account the various risks that can be found during the process so that these do not come as a surprise and do not slow down the thesis.

Some of the risks that can be run upon are:

- **Deadlines:** since the thesis must be finished in a limited time span, a difficulty that can be found is not finishing the objectives in time and lacking features in the final model, such as the inclusion of more identifiable diseases. For this reason, a proper organization of the project will be crucial.
- **Insufficient data:** new datasets will have to be found, but lacking these would be harmful for the accuracy. A solution would be to make the model learn faster by modifying parameters like the learning rate, but this can still decrease the performance.
- **Unbalanced data:** even if sufficient data is acquired, something very likely is that the number of entries for each category is significantly different, which would yield worse results for the population with worse data. In that case, the architecture and hyper-parameters will have to be adjusted to make up for this imperfection.
- **Computational risk:** the training step can require a considerable amount of time and memory, which would increase the time required to obtain the model past the deadlines.
- **Lack of knowledge about signal processing:** signal processing is a field which is relevant to this study to an extent, since it is required to analyze the input. However, this is outside the contents learned in the degree of the thesis, so this might make it more difficult to learn about it due to a lack of background.

1.4. Methodology and rigor

1.4.1 Methodology

For this thesis, a mix of the agile and the Kanban methodologies will be used.

On the one hand, in order to follow the agile methodology, the project will be divided into several *sprints*: short phases of the project (in this case, ranging from 1 to 2 weeks) which will be focused on a particular set of tasks each. This style is thought to suit the project due to the limited time to fulfill the requirements, which makes it important to keep a constant pace of work, while also dividing problems in smaller objectives to simplify the tasks.

On the other hand, Kanban is based on a *Kanban board*, which allows a simple visualization of the current state of the project by displaying three categories of tasks: the ones to be done, those that are currently being worked on and the finished tasks. This permits significant flexibility and can improve organization due to the visual and intuitive nature of the method.

These methodologies will be followed using Trello, an application used to organize tasks in visible cards which can be arranged as the user desires.

1.4.2 Validation

The project will be monitored and validated thanks to periodical meetings with the tutor and version control through GitHub.

GitHub will allow the use of a repository in the cloud, which, besides facilitating backups for the project's workspace and being useful for organization, will make it possible for the tutor to follow the work and its results whenever changes are made.

For the practical part of this thesis, the results can be evaluated with the accuracy achieved by the model, which will directly determine the performance and how close the project is to the main goal.

2. Background

2.1. Seizures

Seizures are a phenomenon which involves the loss of someone's control over their body and their consciousness, often involving convulsions, although they can also be less obvious and simply make the victim stop acting and staring blankly into emptiness for a brief period of time [1]. Around 10% of people suffer from at least one seizure in their lifetime, but this becomes a dangerous condition if someone suffers two or more unprovoked seizures separated by at least one day [1][2]. These can be classified in two types, which are the focus of this thesis: epileptic seizures (ES) and Psychogenic Non-Epileptic Seizures (PNES).

On the one hand, ES are a neurological illness which causes a person's neurons to send an overload of electrical signals, leading to repeated seizures. Immediate treatment is key to be able to cure the disease properly, and it is done through antiseizure drugs [1].

On the other hand, PNES are an illness which also causes seizures, but they are a psychiatric illness instead of being epileptic, despite the symptoms being almost identical. As a consequence, they must not be treated with medication (which may even have negative effects), but through therapy [1].

2.2. Electroencephalograms

An electroencephalogram (EEG) is a recording of the electrical signals emitted by the neurons from the cerebral cortex after a synapsis process. These are measured in microvolts by a set of electrodes attached to the scalp which are connected to an electroencephalograph [7].

EEG signals are often distributed in several frequency bands, normally the Gamma band (30 Hz and up), the Beta band (12-30 Hz), the Alpha band (8-12 Hz), the Theta band (4-7 Hz) and the Delta band (0-4 Hz) [7][8]. These will be used in several data extractors when transforming the EEG data into input graphs. In addition, the Beta band will be replaced by the Lower Beta band (12-16 Hz) and the Higher Beta band (16-30 Hz).

2.2.1 Power Spectral Density

One of the features that will be extracted from the available EEGs is the Power Spectral Density (PSD). This represents the average power of a signal in a determined frequency band, and in this case it will be used for the aforementioned bands used to analyze EEGs. PSD is calculated using equation (1), where f is a given frequency, S is the PSD, τ is time and X is the signal [9].

$$S(f) = \lim_{\tau \to \infty} \frac{|X(f)|^2}{\tau} \qquad (1)$$

2.2.2 Spectral Coherence

Another feature which can be extracted is the Spectral Coherence (SC). This coefficient measures the relation between two signals using the covariance between them at a given frequency band. It is calculated with equation (2), where f is a given frequency, C is the Spectral Coherence, S_{xy} is the cross PSD of the two signals and S_{xx} and S_{yy} are the PSDs of signals x and y, respectively [10].

$$C(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)} \qquad (2)$$

2.2.3 Pearson Correlation

The Pearson correlation coefficient is a widely used statistic used to measure the correlation between two variables, and it has a value between -1 and 1, where 0 indicates no correlation and -1 and 1 show a fully linear relationship (negative and positive, respectively). The equation used to calculate said value is (3), where r is the correlation, σ_{xy} is the covariance of the two signals and σ_x and σ_y are the standard deviations of said signals [11].

$$r = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \qquad (3)$$

2.2.4 Phase Lag Index

Another approach which can be taken to compare two signals is the Phase Lag Index (PLI), which measures the difference between the phases of said signals. This is calculated using equation (4), where $\Delta\Phi_{xy}$ represents the phase difference between the two signals and *sign* is the signum function, which removes any phase difference equivalent to 0 mod π [12]. This statistic was proposed by [13] as a way to specifically analyze EEGs, making it an interesting feature to use for this project.

$$PLI = |\ \langle sign[\Delta\Phi_{xy}(t_k)] \rangle\ | \qquad (4)$$

2.3. Neural networks

A key part of this project is the use of neural networks, as these are the basis of the goal model.

An artificial neural network is a computational model designed to imitate biological brains by connecting several layers of neurons (or perceptrons) [14]. These neurons receive a set of numerical values as their input and output another based on these values, an activation function and various coefficients associated to one input each known as **weights**. These weights determine the output of the **activation function**, and their value is optimized so that the output calculated by the network using its layers of neurons, their activation functions and a given input is as close as possible to the expected output.

In order to achieve this, the model must be trained using gradient descent and backwards propagation. The gradient descent method uses a loss function (representing how far the model's prediction is from the true output) to calculate its gradient over the weights, which is used to update said weights so that the loss is minimized gradually according to a factor called **learning rate**. This gradient is calculated using the back propagation algorithm, which iteratively calculates the derivative of each perceptron for every weight according to its activation function, starting from the output layer backwards.

The activation functions used for this project will be the linear function, where every weight is the slope of its corresponding input value, and the **Leaky ReLU** function, which takes the output of a linear function and, if the output is negative, it is multiplied by a small factor called the leak. This is done so that the negative output is almost zero, but at the same time the gradient for that function does not become zero in that case.

As for the loss function, the experiments will use **Cross-Entropy loss**, which is a loss function designed for classification models, and it is meant to fit the model so that the predicted values are as close to the actual label as possible. As a result, Cross-Entropy Loss is calculated as shown in (5), where t_i is the true output, p_i is the predicted output and n is the number of classes [15].

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i) \qquad (5)$$

Lastly, an **Adam optimizer** will be employed to train the model. Unlike other optimization algorithms like Stochastic Gradient Descent (which Adam builds upon), Adam changes the learning rate according to the mean and the variance of the gradients, combining the advantages of other algorithms like Adaptive Gradient Algorithm and Root Mean Square Propagation [16].

2.3.1 Convolutional Neural Networks

A crucial component of modern advancements in deep learning is the utilization of Convolutional Neural Networks (CNNs), which play a prominent role in the state-of-the-art models that we are comparing our own model against. Convolutional Neural Networks are a specialized type of artificial neural network designed specifically for processing grid-like data, such as images or signals [17].

Similar to traditional neural networks, CNNs consist of interconnected layers of neurons. However, what sets CNNs apart is their ability to effectively exploit the spatial structure present in the input data. This is accomplished through the application of convolutional layers, which convolve learnable filters over the input data, extracting local patterns and features. These filters, often referred to as kernels or feature detectors, capture specific characteristics within the input data, such as edges, textures, or shapes, by aggregating neighboring values into a new one using learnable weights, which are calculated through gradient descent and back propagation.

One of the primary advantages of CNNs is their ability to automatically learn and discover relevant features from raw data. This is achieved by employing a hierarchical structure of multiple convolutional layers, where each layer learns increasingly complex and abstract representations of the input. The initial layers capture low-level features, such as edges or corners, while deeper layers learn high-level representations, combining these low-level features to form more meaningful and discriminative patterns.

Convolutional Neural Networks have demonstrated remarkable success in various computer vision tasks, such as image classification, object detection, and image segmentation. Their ability to automatically learn relevant features from raw data and their hierarchical structure make them powerful tools for extracting meaningful information from complex visual inputs.

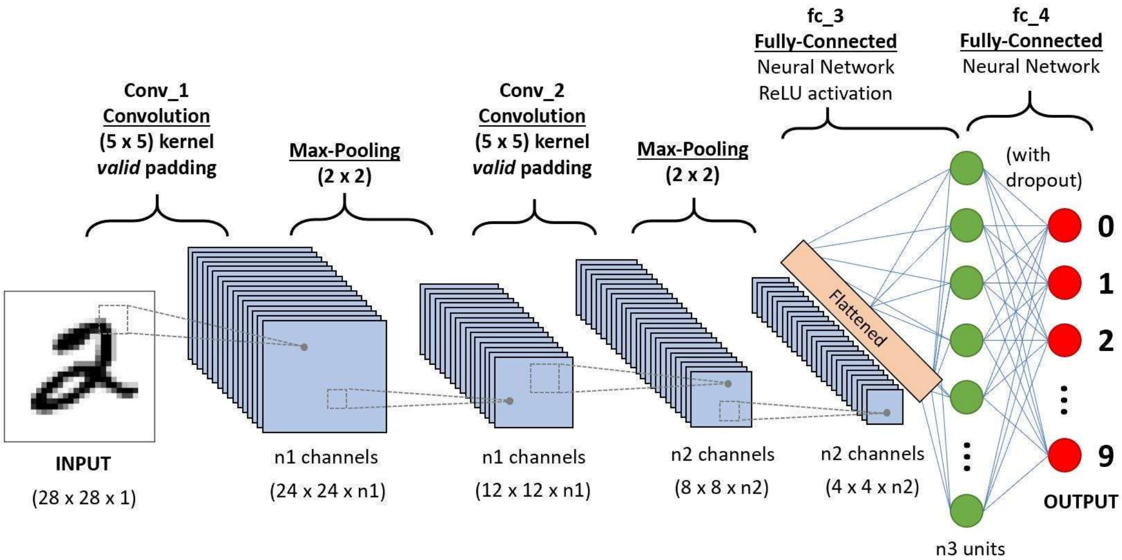


Figure 2: graphical example of a CNN. [17]

By comparing our model against these state-of-the-art Convolutional Neural Networks, we can gain valuable insights into its performance and assess its potential for advancements in the field.

2.3.2 Graph Neural Networks

One of the main issues with Convolutional Neural Networks is how much they depend on spatial relationships. This makes it difficult and often impractical to apply CNNs to certain problems where the nodes have non-Euclidean relations (e.g., social networks).

This is where **Graph Neural Networks (GNNs)** excel at. These neural networks can be seen as a generalization of CNNs, where the relationships between nodes can be chosen arbitrarily. Instead of representing the input as a 2-dimensional array of nodes, the network is fed a graph, composed by nodes connected by edges. Each node and edge can have several features, which are used to predict the correct output through a procedure known as message passing, which is done through an update function that modifies the features of a node (or an edge, depending on the case) according to the values of its neighbors. This procedure is similar to the filters used in CNNs[18][19].

GNNs can execute 3 main types of predictions:

- Graph-level: the prediction is associated with the input graph itself, and it is often used for graph classification.
- Node-level: the prediction is performed on nodes or on their features.
- Edge-level: the model predicts proprieties of edges, and often the edges themselves. This can be done, for example, if the graph is a social network, where the nodes represent users and the edges show an existing friendship between users, and the model's goal is to predict said friendships given user data.

In this project's case, the model will execute a graph-level task, since the goal is to classify the input graph.

There are several types of GNN layers which will be used in this project:

Graph Convolutional Networks

This network type, proposed by [19], uses equation (6) to calculate the layer's new features for every node, where H_t is the matrix with said features, H_{t-1} is the matrix with the input features of the graph's nodes, $\tilde{A}$ is the adjacency matrix of the graph with added self-loops, D is the degree matrix (also with self-loops), W is a matrix with the trainable weights of the layer and σ is an activation function.

$$H_t = \sigma\left(D^{-\frac{1}{2}}\tilde{A}^T D^{-\frac{1}{2}} H_{t-1} W\right) \quad (6)$$

The adjacency matrix's values also depend on the non-trainable weights of the edges, which represent how related two nodes are. The issue with this approach is that these weights will remain the same throughout the training and must be computed *a priori*, which requires additional pre-processing.

Graph Attention Networks

This alternative to Graph Convolutional Networks, proposed by [20], addresses the problem of said networks by assigning **attention coefficients** to every pair of neighbor nodes depending on their features and an additional weight matrix, as shown on equation (7), where e_{ij} is the attention coefficient between nodes i and j, h_i is node i's horizontal feature vector, W is the aforementioned additional weight matrix and *att* is the attention mechanism. This attention coefficient is then normalized using a softmax function, and then they are used as edge weights in the adjacency matrix like in GCNs [21].

$$e_{ij} = att(Wh_i, Wh_j) \quad (7)$$

The attention mechanism is computed as shown in equation (8), where a is a trainable weight vector.

$$att(v_i, v_j) = LeakyReLU(a^T[v_i||v_j]) \quad (8)$$

Graph Attention Layers and their variants also use **multi-head attention**, which allows layers to perform this process repeatedly with different parameters and then aggregate these to acquire the final attention coefficients. Each calculated attention map is called a head.

Self-supervised Graph Attention Networks

This is an ampliation on GATs proposed by [22], which intends to improve their performance on noisy graphs. It is also more focused on self-supervision tasks, which is not the goal of this project, so an in-depth study of this type of layer will not be done, but they will still be tested to explore more alternatives for the model's architecture.

In summary, this type of network includes two variants of itself: scale dot-product and a mixture of dot-product and standard GATs. Both of these are trained to predict the existence or not of every edge in the node by using the attention coefficients. A diagram of this architecture is shown on Figure 3, where GO is a traditional GAT operator, DP is the dot-product, MX is the mixture of GO and DP and SD is the scaled dot-product.

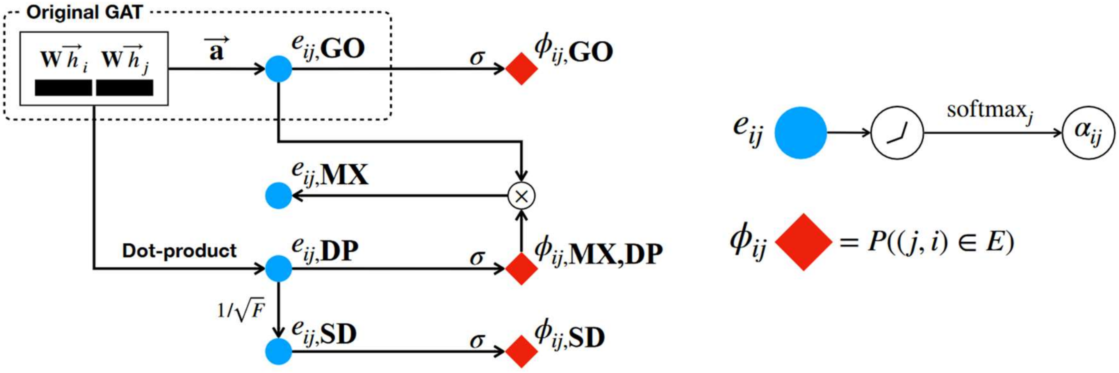


Figure 3: diagram of the SuperGAT architecture. DP is the dot-product operator, GO is the traditional GAT operator, SD is the scaled dot-product and MX is a mixture of DP and GO. [22]

Graph Transformer Networks

Finally, Graph Transformer Networks are an operator proposed by [23] which apply the Transformer architecture by [24], most commonly known for its various Natural Language Processing uses, into GNNs.

Like GATs, Graph Transformers use a multi-head attention mechanism. These two differ in the way the attention score is calculated. In a Transformer head, the attention coefficients are calculated from 3 values: Query (Q), Key (K) and Value (V) (the acquisition of these values may vary depending on the application), which are fed to a trainable linear layer each. Q and K are multiplied, and a normalization and a softmax function are applied to them. This value is finally multiplied by V, and the result is a matrix with the attention coefficients. This process is exemplified in Figure 4, which shows a detailed diagram with an example of an NLP Transformer.

This particular application of Transformers on GNNs performs this on every node by assigning the center node's features to the Q value and the neighboring nodes' features to the K and V values, as shown in Figure 5.

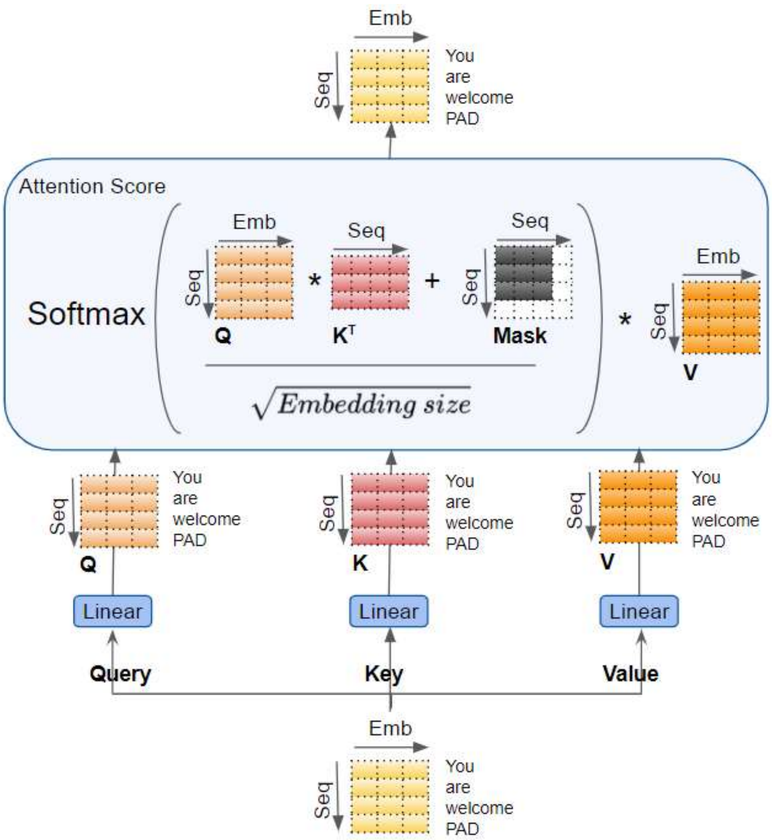


Figure 4: diagram of an example Transformer application on NLP. [25]

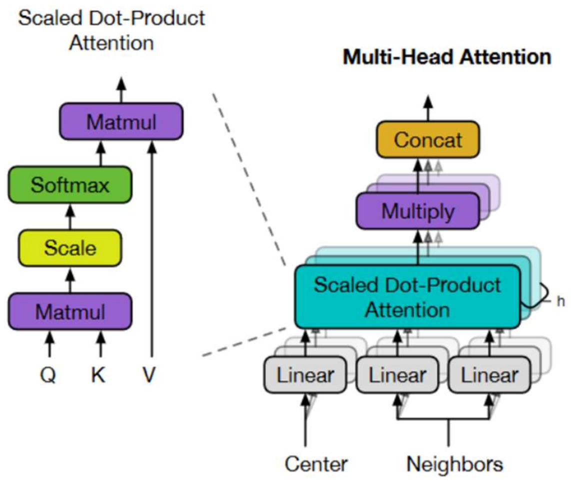


Figure 5: diagram of the calculations followed by GraphTransformer to acquire the attention coefficients. [23]

3. State-of-the-art

Initial research was conducted on the existing state of the art regarding epilepsy analysis through EEGs and machine learning. To achieve this, several articles using different methods and architectures were read and documented in Table 1. Since research on ES and PNES differentiation appears to be limited, other similar classifications were included, since even though they do not solve the exact same problem, they help provide insight on techniques used to analyze EEGs in the field of epilepsy. For example, in [28], it was discovered that information relative to the Beta band is significant for EEG classification, which highlights the importance of extracting features which separate EEG data in frequency bands.

We can observe that many approaches exist for diagnosing and classifying ES and PNES, such as Support Vector Machines (which appear to perform well), Convolutional Neural Networks, K Nearest Neighbors and Linear Discriminant Analysis. However, Graph Neural Networks do not seem to be used as often as the other approaches, as the only works using GNNs that were found during the research were [26], [27] and [28]. For this reason, this project could help shed more light on whether GNN models are appropriate for tasks of this type or not.

Author	**Classes**	**Acc**	**Architecture**	**Dataset**
Lo Giudice M. et al [5]	**PNES, ES, HC**	**85.7%**	**CNN**	**Regional Epilepsy Center, U. of Catanzaro**
Lo Giudice M. et al [29]	**PNES, ES**	**94.4%**	**CNN**	**Regional Epilepsy Center, U. of Catanzaro**
Ahmadi N. [28]	**PNES, ES**	**80%**	**KNN/SVM/DT /GBoost/RF**	**Ghent University Hospital, Belgium**
Zsom A. et al [30]	**PNES, ES**	**78%**	**XGBoost**	**Own acquisition**
Cura O. K. et al [31]	ES, HC	90.2%	KNN/SVM/BN	?
Cura O. K. et al [31]	ES, HC	90.3%	CNN	?
Varonne G. et al [32]	PNES, HC	98%	SVM	Regional Epilepsy Center, U. of Catanzaro
Varonne G. et al [32]	PNES, HC	98%	LDA	Regional Epilepsy Center, U. of Catanzaro
Varonne G. et al [32]	PNES, HC	81%	BN	Regional Epilepsy Center, U. of Catanzaro
Li Z. et al [33]	7 types of seizures	91%	GGN	TUH-EEG Corpus
Tang S. et al [26]	CF, GN, AB, CT	93%	GNN	TUH-EEG Corpus
Rahmani A. et al [27]	Focal, generalized, non-seizures	82.7%	Meta-GNN	TUSZ-dataset
Boonyakitanont P. et al [34]	ES, HC	70% (F1)	CNN	?
Brari Z. [35]	ES, HC	100%	SVM	Bonn University
Chen Z. [36]	ES, HC	98.62%	CNN	CHB-MIT Scalp database

Table 1: state of the art. [Own creation].

As seen in Table 1, only three projects were found to differentiate between ES and PNES. Out of these, one of them diagnoses healthy patients, whereas the other two focus on differentiating ES and PNES. For this reason, the two papers by Lo Giudice M. *et al* will be taken as a reference for both binary and ternary classification, as these achieve the highest accuracy for their respective classifications, with an accuracy of **94.4%** and **85.7%** for binary and ternary classification, respectively. Therefore, this project's primary objective will be to develop a GNN model with an accuracy superior or close to these values depending on the type of classification.

In addition to the fact that the existing models have an improvement margin, none of them were found to apply explainabiltiy for their predictions, which is one of the goals of this project. Therefore, this part of the thesis could bring new interesting insights into the current research of the field.

4. Methodology and results

4.1. Code and functionalities

As mentioned in the previous sections, this project would start by taking the code of the project made by [6] as a reference, which includes graph extraction, training and validation functionalities, and modifying it to suit the new classification. This code appears to achieve great results for Alzheimer's disease diagnosis through EEGs, so it was considered to be a good starting point.

Firstly, the code was forked from its public *GitHub* repository into a new one dedicated to this thesis, and it was tested and adapted to work in the working environment (the server provided to the researcher).

Secondly, different feature extraction methods were used to train a test model which utilized the architecture which performed the best during the research by [6], and a combination with particularly good results was with Power Spectral Density for node extraction and attention for the edges, yielding an accuracy of 91% for Alzheimer's disease, MCI and healthy control classification, matching that of the paper by [6], so these features will be used as a starting point for the experiments. Other combinations, such as statistical features and raw signals, only yielded around 68% of accuracy at most. All the fitting and validation was done with the same dataset used by [6].

Finally, the code was expanded with more scripts and adapted to ES and PNES classification, allowing alternative training methods like leave-one-group-out cross-validation.

4.2. Dataset

The main dataset used for this project was provided by the same researchers who directed the project taken as a reference: the Regional Epilepsy Center dataset from the University of Catanzaro. This will ensure that the accuracies are not biased by the quality of the datasets and therefore can be compared fairly.

Said dataset focuses on patients with Psychogenic Non-Epileptic Seizures and Epileptic Seizures. It specifically consists of interictal EEG data, which is obtained during the intervals between clinical attacks and does not contain seizures or ictal manifestations, which avoids ethical concerns related to seizure manipulation techniques. The dataset includes EEG recordings from 103 subjects, comprising 42 subjects with new-onset clinically diagnosed epilepsy, 42 subjects with video-recorded PNES, and 19 control subjects who are clinically healthy on EEG inspection. Only artifact-free EEG recordings were considered for analysis.

The EEG signals were acquired using a standard 10-20 system with 19 channels. Participants were selected based on their willingness to participate, informed consent, and normal interictal EEG at visual inspection. Those using psychotropic drugs chronically were excluded. The EEG recordings underwent visual inspection for artifact removal, were subsampled at 256 Hz, and stored in MAT (MATLAB File Format) files. Further processing involved band-pass filtering the recordings between 0.5-32 Hz using the EEGLab toolbox in MATLAB. The data was then partitioned into 5-second EEG segments, consisting of 1 280 samples (5 s · 256 Hz), with a 50% overlap to enhance the dataset's size through data augmentation.

Overall, the dataset comprises a total of 25 754 EEG segments, with 4 820 segments from control subjects, 9 846 segments from epileptic subjects, and 11 088 segments from PNES subjects. The EEG segments are recorded from 19 channels, namely: 'Fp1', 'Fp2', 'F7', 'F3', 'Fz', 'F4', 'F8', 'T3', 'C3', 'Cz', 'C4', 'T4', 'T5', 'P3', 'Pz', 'P4', 'T6', 'O1', and 'O2'.

4.3. Training

4.3.1 Training configuration

The training process for the classification model in this study involved careful configuration of various components to ensure optimal performance and accurate predictions. The following sections outline the key elements used during this phase.

All experiments were set up using Pytorch, a popular Python library used for machine learning.

Optimizer and learning rate

To optimize the model's performance, the Adam optimizer was employed with a learning rate of 0.001. Adam is a popular choice for optimization algorithms in deep learning due to its adaptive learning rate mechanism, which adjusts the learning rate for each parameter during training. This adaptive nature helps the optimizer converge faster and achieve better results.

Scheduler

A scheduler was employed to dynamically adjust the learning rate during training. Specifically, a "reduction on plateau" scheduler was utilized. This scheduler monitors the model's performance on a specified metric, such as validation loss or accuracy, and reduces the learning rate if the monitored metric fails to improve for a certain number of epochs. In this study, a patience of 10 epochs was set, meaning that if no improvement was observed for 10 consecutive epochs, the learning rate would be reduced. The reduction factor was set to 0.5, halving the learning rate when triggered. Additionally, a minimum learning rate of 1e-5 was defined to prevent the learning rate from becoming too small.

Loss function

For the classification task at hand, the cross-entropy loss function was chosen. As explained previously, cross-entropy is a commonly used loss function in classification problems, as it measures the dissimilarity between the predicted probabilities and the true class labels. By minimizing the cross-entropy loss, the model is incentivized to accurately predict the class probabilities for each input sample.

Epochs and batch size

The training process spanned 300 epochs for the train-validation-test split experiments (and 100 for Leave One Group Out cross-validation), where each epoch represents a complete pass through the entire dataset. This choice of epoch count strikes a balance between allowing the model to learn from the data and avoiding overfitting. A batch size of 128 samples was utilized, determining the number of samples processed in each iteration of the training loop. The batch size influences the model's convergence speed and memory requirements during training.

Train-validation-test split

For most experiments, the dataset was simply split into three groups: training data, validation data and testing data. The model itself was trained using the training data, and the best model was chosen by evaluating its performance with validation data. The performance of said model was verified using the testing data.

The data was distributed as follows:

- Training data: 70%.
- Validation data: 20%.
- Testing data: 10%.

Leave One Group Out cross-validation

To ensure robust evaluation of the model's performance and mitigate any biases, a Leave One Group Out cross-validation strategy was employed for one of the experiments. This approach involved dividing the dataset into groups based on individual patients. In each iteration of the cross-validation process, one patient's data was reserved for validation, while the remaining patients' data was used for training. This methodology ensures that the model is tested on patients that it has not encountered during training, enabling a more realistic assessment of its generalization capabilities.

Model architecture

Several architectures were designed to be tested during training experiments, each of them using graph convolutional layers of several types, where MLP indicates a Multi-Level Perceptron and FC refers to a fully connected layer with a Leaky ReLU activation function.

Model	Architecture
GraphConv	GCNConv(in = IN, out = 32) + BatchNorm + LeakyReLU GCNConv(in = 32, out = 32) + BatchNorm + LeakyReLU GCNConv(in = 32, out = 64) + BatchNorm + LeakyReLU GlobalAddPool MLP(FC(192, 64), FC(64, 32), FC(32, OUT))
GATConv	GATConv(in = IN, out = 16) + BatchNorm + LeakyReLU GATConv(in = 16, out = 32) + BatchNorm + LeakyReLU GATConv(in = 32, out = 64) + BatchNorm + LeakyReLU Dropout(p = 0.2) GlobalAddPool MLP(FC(64·HEADS·HEADS, 64·HEADS), Dropout(p = 0.2), FC(64·HEADS, 32·HEADS), Dropout(p = 0.2), FC(32·HEADS, OUT))
SuperGATConv	SuperGATConv(in = IN, out = 16) + BatchNorm + LeakyReLU SuperGATConv(in = 16, out = 32) + BatchNorm + LeakyReLU SuperGATConv(in = 32, out = 64) + BatchNorm + LeakyReLU Dropout(p = 0.2) GlobalAddPool MLP(FC(64·HEADS·HEADS, 64·HEADS), Dropout(p = 0.2), FC(64·HEADS, 32·HEADS), Dropout(p = 0.2), FC(32·HEADS, OUT))
TrasformerConvV1	TransformerConv(in = IN, out = 16) + BatchNorm + LeakyReLU TransformerConv(in = 16, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 64) + BatchNorm + LeakyReLU Dropout(p = 0.2) GlobalAddPool MLP(FC(64·HEADS·HEADS, 64·HEADS), Dropout(p = 0.2), FC(64·HEADS, 32·HEADS), Dropout(p = 0.2), FC(32·HEADS, OUT))
TrasformerConvV2	TransformerConv(in = IN, out = 16) + BatchNorm + LeakyReLU TransformerConv(in = 16, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 64) + BatchNorm + LeakyReLU Dropout(p = 0.2) GlobalAddPool

	MLP(FC(64·HEADS·HEADS, 64·HEADS), Dropout(p = 0.2), FC(64·HEADS, 32·HEADS), Dropout(p = 0.2), FC(32·HEADS, OUT))
TrasformerConvV3	TransformerConv(in = IN, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 64) + BatchNorm + LeakyReLU Dropout(p = 0.2) GlobalAddPool MLP(FC(64·HEADS·HEADS, 64·HEADS), Dropout(p = 0.2), FC(64·HEADS, 32·HEADS), Dropout(p = 0.2), FC(32·HEADS, OUT))
TrasformerConvV4	TransformerConv(in = IN, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 32) + BatchNorm + LeakyReLU GCNConv(in = 32, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 64) + BatchNorm + LeakyReLU Dropout(p = 0.2) GlobalAddPool MLP(FC(64·HEADS·HEADS, 64·HEADS), Dropout(p = 0.2), FC(64·HEADS, 32·HEADS), Dropout(p = 0.2), FC(32·HEADS, OUT))
TrasformerConvV5	TransformerConv(in = IN, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 32) + BatchNorm + LeakyReLU TransformerConv(in = 32, out = 64) + BatchNorm + LeakyReLU GlobalAddPool MLP(FC(64·HEADS·HEADS, 64·HEADS), FC(64·HEADS, 32·HEADS), FC(32·HEADS, OUT))

Table 2: model architectures used for the experiments. [Own creation].

Some of these are architectures used directly from the paper by [6], but most are slight modifications applied to these to test their effects on overall performance.

By carefully configuring the optimizer, scheduler, loss function, epoch count, batch size, and cross-validation strategy and the model architecture, the training phase of this study aimed to train a classification model that can effectively differentiate the given classes.

4.3.2 Training results

Hyperparameter, architecture and feature search

Several combinations of hyperparameters, architectures and feature extractors were tested in experiments using train-validation-test splitting, which were used to select the configuration with the best accuracy. To make the search faster, binary classification of ES and PNES without healthy control subjects was used for these experiments.

Initially, PSD and attention edges were used as feature extractors, as those seemed to yield relatively good results when analyzing EEG signals in the paper by [6], and the TransformerConvV1 architecture was selected for the first few experiments for the same reason.

The first experiments were run with different head counts for the Transformer convolutional layers to perform a search on said hyperparameter, and after testing for 2, 3 and 4 heads, the best testing metrics were obtained when using **3 heads**, acquiring an accuracy of 91.75%, 92.06% and 91.83%, respectively. From this point on, 3 heads will be used for attention layers.

The next step was to find which feature extractor represented the different edge weights the best, so experiments were run for the Pearson Correlation Coefficient, the Phase Lag Index and Spectral Coherence, whose results were compared to those acquired by attention edges in the previous set of experiments. The results showed that using **attention** yielded slightly better performance, showing a testing accuracy of 92.44%, so this was chosen for the following runs.

After this, the different model architectures were designed and tested as well. Firstly, the Transformer layers were replaced by SuperGATConv, GATConv and GCNConv, but as expected, none of those surpassed the Transformer architecture, so the rest of the models were designed by making changes to it, like modifying the number of neurons of each layer, adding new graph convolutional layers and removing the dropout layers. The results showed that **TransformerConvV3** and **TransformerConvV5** are the best architectures out of the chosen alternatives, with testing accuracies of 92.74% and 92.9%, respectively.

The metrics of the aforementioned experiments can be seen on Table 3, and the evolution of the loss and the accuracy of the TransformerConvV5 experiment can be seen in Figure 6.

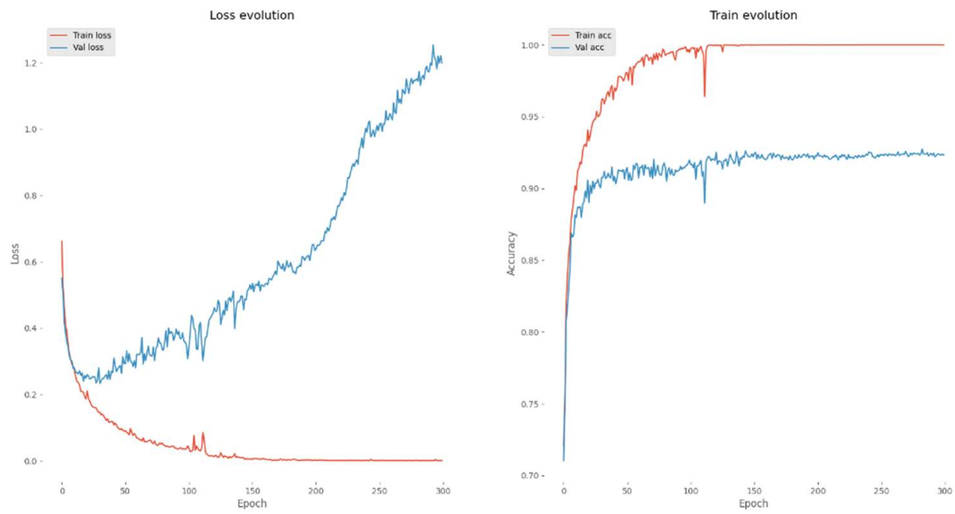


Figure 6: plots showing the evolution of the loss and the accuracy of the experiment for binary classification with TransformerConvV5.

Model	Heads	Nodes	Edges	Set	Acc.	Bal. Acc.	Prec.	Recall	F1-score	AUC
TrasformerConvV1	2	PSD	Att	Train	99.99%	99.99%	99.99%	99.99%	99.99%	1
				Val	92.48%	92.50%	92.40%	92.50%	92.44%	0.97
				Test	91.75%	91.77%	91.70%	91.77%	91.73%	0.98
	3	PSD	Att	Train	99.99%	99.99%	99.99%	99.99%	99.99%	1
				Val	92.79%	92.79%	92.76%	92.79%	92.78%	0.98
				Test	92.06%	91.96%	91.99%	91.96%	91.97%	0.98
	4	PSD	Att	Train	100.00%	100.00%	100.00%	100.00%	100.00%	1
				Val	91.85%	91.74%	91.86%	91.74%	91.79%	0.97
				Test	91.83%	91.80%	91.82%	91.80%	91.81%	0.98
GraphConv	3	PSD	SC	Train	97.03%	97.03%	97.02%	97.03%	97.02%	1
				Val	88.38%	88.31%	88.34%	88.31%	88.32%	0.95
				Test	87.93%	87.84%	87.89%	87.84%	87.86%	0.96
	3	PSD	PCC	Train	84.10%	84.09%	84.02%	84.09%	84.05%	0.93
				Val	79.64%	79.59%	79.64%	79.59%	79.61%	0.88
				Test	77.77%	77.71%	77.72%	77.71%	77.71%	0.86
	3	PSD	PLI	Train	90.06%	90.05%	90.02%	90.05%	90.03%	0.97
				Val	80.20%	80.15%	80.15%	80.15%	80.15%	0.88
				Test	81.74%	81.56%	81.56%	81.56%	81.56%	0.89
GATConv	3	PSD	Att	Train	99.90%	99.90%	99.90%	99.90%	99.90%	1
				Val	90.24%	90.24%	90.15%	90.24%	90.19%	0.96
				Test	89.99%	89.98%	89.96%	89.98%	89.97%	0.97
SuperGATConv	3	PSD	Att	Train	100.00%	100.00%	100.00%	100.00%	100.00%	1
				Val	92.56%	92.55%	92.51%	92.55%	92.53%	0.97
				Test	92.36%	92.31%	92.34%	92.31%	92.33%	0.97
TrasformerConvV2	3	PSD	Att	Train	98.83%	98.83%	98.82%	98.83%	98.82%	1
				Val	90.27%	90.27%	90.24%	90.27%	90.25%	0.96
				Test	89.30%	89.28%	89.23%	89.28%	89.25%	0.96
TrasformerConvV3	**3**	**PSD**	**Att**	**Train**	**99.99%**	**99.99%**	**99.99%**	**99.99%**	**99.99%**	**1**
				Val	**92.69%**	**92.62%**	**92.70%**	**92.62%**	**92.66%**	**0.98**
				Test	**92.74%**	**92.69%**	**92.72%**	**92.69%**	**92.70%**	**0.98**
TrasformerConvV4	3	PSD	Att	Train	99.69%	99.69%	99.68%	99.69%	99.69%	1
				Val	91.62%	91.61%	91.58%	91.61%	91.60%	0.97
				Test	91.06%	91.10%	91.01%	91.10%	91.04%	0.97
TrasformerConvV5	**3**	**PSD**	**Att**	**Train**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**1**
				Val	**93.07%**	**93.03%**	**93.04%**	**93.03%**	**93.04%**	**0.98**
				Test	**92.90%**	**92.93%**	**92.89%**	**92.93%**	**92.89%**	**0.98**

Table 3: results of the experiments run during architecture, feature and hyperparameter search. [Own creation].

Ternary classification

After finding the optimal configuration for the model, another experiment was run to test the model's performance when classifying not only PNES and ES, but also healthy patients. The configuration used for this experiment is with TransformerConvV3 architecture, 3 heads, PSD node extraction and PLI edge extraction.

The resulting metrics can be seen on Table 4.

Set	Acc.	Bal. Acc.	Prec.	Recall	F1-score	AUC
Train	100.00%	100.00%	100.00%	100.00%	100.00%	1
Val	90.81%	90.76%	90.97%	90.76%	90.86%	0.98
Test	91.12%	90.86%	91.89%	90.86%	91.32%	0.98

Table 4: results of experiment run on ternary classification.

Leave One Group Out cross-validation

An additional experiment was performed early-on with Leave One Group Out cross-validation and binary classification to test the performance of GNNs when dealing with inter-subject features instead of intra-subject parameters. Since this was an experiment performed in the earlier stages of the project, it was done with the TransformerConvV1 architecture, 2 heads, PSD node extraction and SC node extraction despite that not being the final optimal configuration.

The results can be seen on Table 5.

	Average accuracy	Minimum accuracy	Maximum accuracy	Standard deviation
Train	97.10%	92.22%	98.22%	0.01558
Test	97.58%	92.46%	100%	0.01585

Table 5: results of Leave One Group Out cross-validation experiment.

4.3.3 GNN explainability

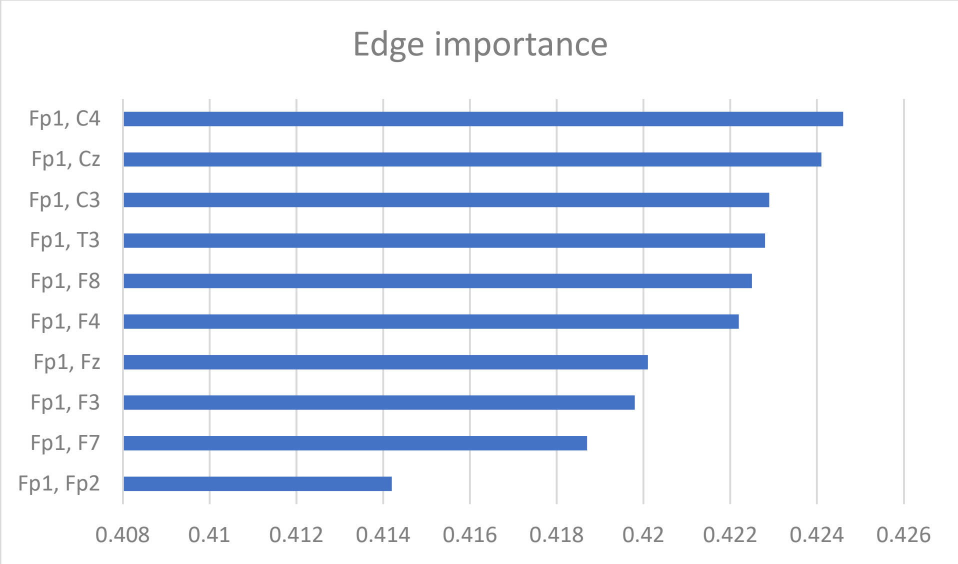
Finally, a significant advantage of GNNs compared to other machine learning approaches to EEG analysis is the high capacity of explainability they possess: since they use graphs to process their data and graph structures can be easily mapped to concepts comprehensible by humans (in this case, an EEG), it is relatively simple to determine a model's "reasoning" behind a prediction.

Fortunately, the Pytorch Geometric library, which was used to set up GNNs, includes a functionality which allows the use of the GNNExplainer algorithm [37]. This algorithm was used to acquire various explanations for the predictions of a subject with epilepsy from the testing dataset.

Firstly, a relevant analysis is the node importance, which allows us to see which of the graph's nodes (the electrodes) have a larger influence on the prediction. As we can see on Figure 7, the most important electrode is T6, which is an electrode situated by the back-right of the head during the recording of an EEG. Figure 8 shows the electrode distribution used during these in the used dataset.

Secondly, another useful analysis is the feature importance, which measures how important every node feature is during a prediction on average for the entire graph. This evaluation was performed on all the data belonging to the subject, and the results can be seen on Figure 9. Unlike what was observed by [28], the Beta band is not the most relevant feature in this case, but one of the lowest, as the Theta band appears to be the most important band for these predictions.

Thirdly, one more relevant feature is the edge importance, which shows the connectivity between all the electrodes.



In

Figure 10, we can see a plot with the 10 most relevant edges. It appears that the edges with the most importance for the predictions are those related to electrode Fp1, one of the front electrodes.

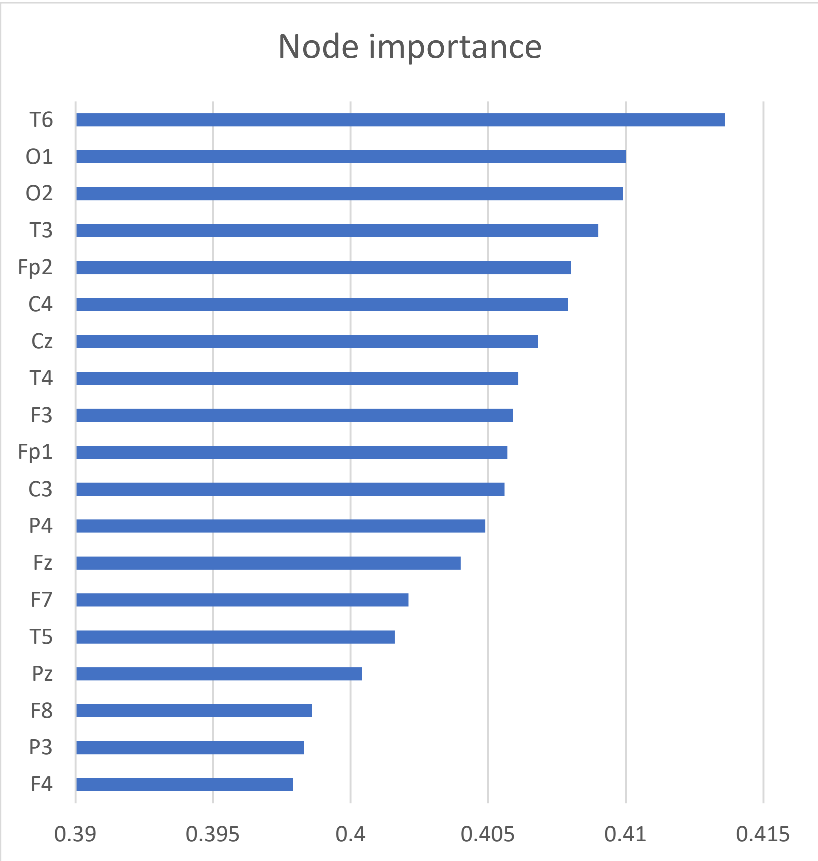


Figure 7: plot showing average node importance for a subject. [Own creation].

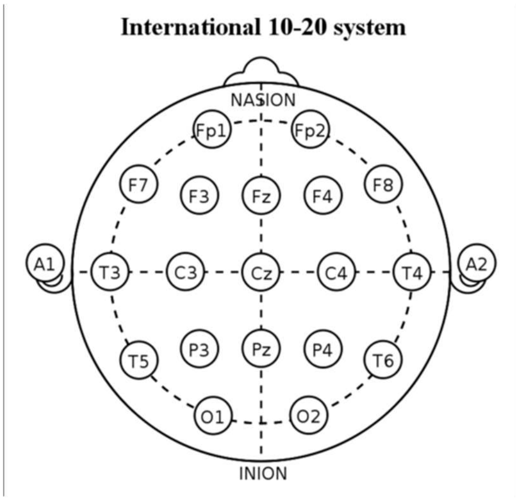


Figure 8: diagram of the placement of the 19 electrodes according to the international 10-20 system. [38]

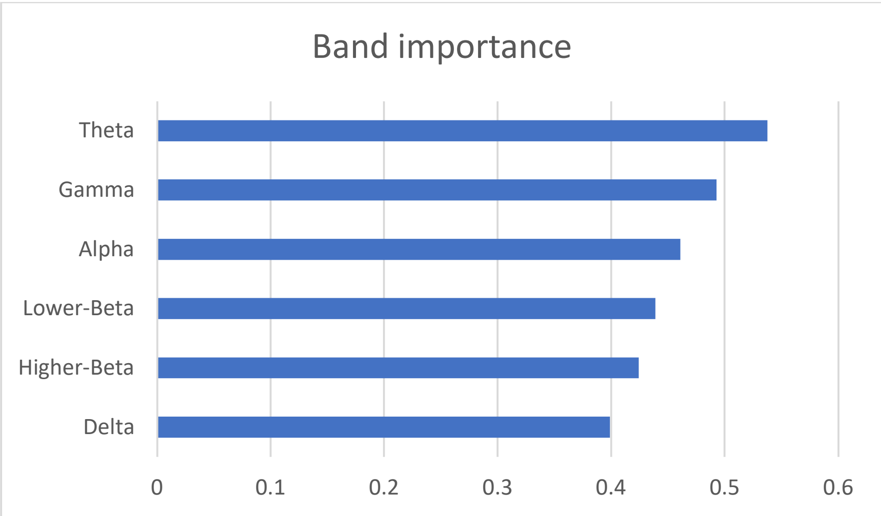


Figure 9: plot showing feature importance for the predictions on one subject. [Own creation].

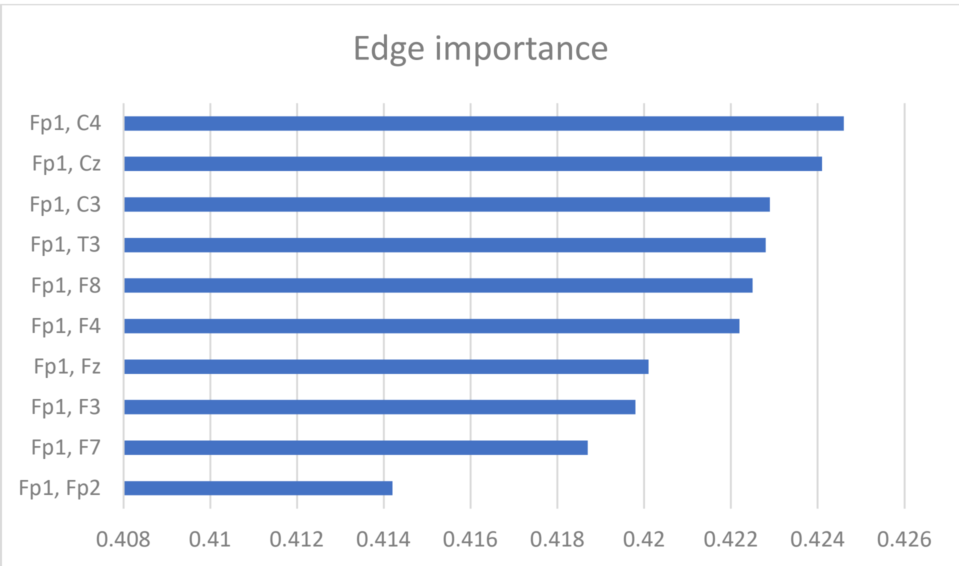


Figure 10: plot showing edge importance for the predictions on one subject. [Own creation].

5. Discussion

As previously shown, several experiments were conducted to find an optimal configuration for GNNs and to compare their performance to other approaches from the state of the art.

The first experiments aimed to differentiate between PNES and ES patients and to properly set up the model. On the one hand, the final accuracy (92.9%) obtained does not outperform the highest accuracy from the state of the art on that type of classification (94.4% [29]) when using train-validation-test splitting.

On the other hand, it does greatly surpass the accuracies obtained by the other found approaches, such as [28] and [30], with accuracies of 80% and 78%, respectively. This shows that GNNs are a viable and reliable alternative to other methods for PNES and ES discrimination.

In the second set of experiments, the classification task was extended to include healthy patients along with PNES and ES subjects. The obtained accuracy of 91.12% demonstrates the model's ability to classify three distinct classes with remarkable precision. The state of the art accuracy for this particular task was reported as 85.7%. The higher accuracy achieved in this study indicates the potential of the proposed model to improve the classification of healthy patients as well, contributing to the identification and differentiation of all three classes accurately better than any other model within the featured state of the art.

In addition, running a binary classification experiment using Leave One Group Out yields interesting results, as the obtained accuracy is higher than that of the train-validation-test method. This is rather surprising, as it was believed that not mixing data from the same subjects across training and testing sets would limit the model’s performance. This result might be a consequence of the training data set being larger than in other experiments, thus leading to the model being better fit.

It is important to note that the utilization of the same dataset as previous research ensures a fair comparison between the achieved results and the work used as a point of reference. The consistency in the dataset allows for a reliable assessment of the model's performance, emphasizing the significance of the obtained accuracies in advancing the field of PNES and ES classification.

In addition, the use of GNNs has an advantage not possessed by other machine learning approaches: the simplicity of explainability. As seen before, analyzing the explanations behind a set of predictions is easy and intuitive, as the explanations are directly associated to the input structure (in this case, an EEG). Therefore, this is another advantage of the proposed model over the state of the art.

In conclusion, the conducted experiments demonstrated notable achievements in the classification of PNES, ES, and healthy patients based on EEG data. The obtained accuracies surpassed the state-of-the-art results for ternary classification and achieved notable results for binary classification, indicating the efficacy and potential of using GNNs in this field.

6. Conclusion and future work

In this study, a research for PNES and ES classification using EEG data and Graph Neural Networks was presented. The proposed model demonstrated remarkable performance, outperforming most state-of-the-art results reported in previous research and greatly surpassing the performance of all known architectures for ternary classification.

By employing a GNN architecture, this model successfully differentiated between PNES and ES with an accuracy of 92.9%, which is still below the highest state of the art accuracy of 94.4% for this type of classification, but the proposed model manages to outperform this if Leave One Group Out cross-validation is used, yielding an accuracy of 97.58% instead. Furthermore, extending the classification task to include healthy patients yielded an accuracy of 91.12%, outperforming the highest known state of the art accuracy of 85.7%. These outcomes highlight the potential of this model in accurately classifying and distinguishing between these medical conditions using EEGs.

In addition, an exciting avenue for future work lies in acquiring biomarkers for binary classification using explainability. By using the model's explainability capabilities shown in this thesis, we can identify and extract meaningful biomarkers associated with PNES and ES, contributing to the development of objective diagnostic tools and potentially advancing personalized treatment strategies, even without using the model itself.

Lastly, this study also researched similar studies from the state of the art and compared the performance of their models, as well as the methods used for their respective classification and their datasets. As a result, this study also provides a chart with the currently available research on seizure classification.

In conclusion, this study showcases the success of the proposed model in classifying PNES and ES. Through further exploration and refinement, research can continue to advance the field of PNES and ES classification, ultimately benefiting patients and healthcare practitioners in diagnosing and treating these conditions.

A. Temporal planning

A.1. Task planning

For this project to succeed, it is important to define the tasks to perform and to plan them according to the methodology chosen for this project: the agile methodology.

Work on the project started around the end of January, and it is expected to last for roughly 20 weeks, with around 3.75 hours of work per day. Knowing this, the expected total time dedicated to the project will be approximately 540 hours.

The tasks, which will be defined shortly, will be given an expected dedication time depending on the amount of work they require. However, this value is a very rough approximation, since, at the time of planning, many of the tasks require procedures which will remain unknown until later during the project, in addition to the elevated cost in time of some actions (training models, for example).

A.1.1 Task definition

The project's tasks will be organized in task groups, which will allow for these to be focused on a single sub-objective:

Meetings

Throughout the thesis, several meetings will be organized between the researcher and the tutor in order to validate the progress and to solve questions related to the project. It is unknown when these meetings will take place, so no particular date will be assigned to them in the planning.

Project management (PM)

These tasks are related to the definition of the project itself, as well as its scope and planning (such as this very section). All of these tasks will be executed sequentially.

- **Defining context and scope (PM.1):** the goals of the projects will be established, as well as its risks and limits.
- **Defining temporal planning (PM.2):** the tasks, the actions to take regarding the risks and the resources will be defined.
- **Defining economic management and sustainability (PM.3):** the budget of the project is planned and its impact on sustainability will be defined.
- **Integrating final project management document (PM.4):** these tasks are summarized in one final section from the document.

State of the art research (SA)

Prior to implementation, a background on the matters referenced in this thesis will be researched to gain a greater understanding of the procedures which will be taken in later tasks.

- **Learning about GNNs (SA.1):** the way GNNs work and how they can be applied will be researched. This is required in order to start the practical work.
- **Researching about similar projects with CNNs (SA.2):** these projects will be taken into account while measuring the performance of the acquired models.
- **Researching about similar projects with GNNs (SA.3):** similar projects will be explored in order to gain insight on the current progress in the area to see if this project matches it.

- **Inspecting code from existing project (SA.4):** a project with a similar goal will be chosen as a reference, and it will be analyzed so that it can be worked with as a basis.
- **Learning about signal processing (SA.5):** while this project will not directly deal with signal processing, it is still preferred to have a basic background just in case.
- **Learning about legal background (SA.6):** the legislation regarding the use of neural networks to perform a diagnose of this kind is an important factor to take into account, so research will be performed to discover the legal boundaries of the thesis.

Replicating results by similar project (R)

This task group marks the beginning of the practical part of this thesis. For here on out, the model training will be done through a server provided to the researcher, which will be accessed via SSH and will provide enough power to perform the fitting in a reasonable amount of time. In addition, the code and the models will be uploaded to a repository in GitHub, which will be kept updated.

The main goal of this section is to experiment with an already existing GNN architecture designed to analyze EEGs and adapt it to the thesis' objective, as well as checking if the code and the architecture work as they should on the server.

- **Analyzing code and functions of existing project (R.1):** the functioning of the aforementioned project's code will be analyzed in order to understand how the pipeline successfully processes EEGs and uses them to train the chosen architecture.
- **Adjusting model and pre-processing (R.2):** various alternatives for the architecture of the model, the pre-processing of the datasets and the hyperparameters will be picked to verify which of these performs the best. These alternatives will be chosen based on the results achieved by the author of the chosen project, and the main reason why they will be tested to make sure that the code for the project works as expected in the provided server.
- **Running experiments (R.3):** while picking these alternatives, several experiments will be run to evaluate which one achieves the highest performance. This task will include the execution time for the model training, so the real time required to complete this step is unknown, since these tasks might need to be repeated several times to make sure everything works as intended.

Expanding the work for ES and PNES (E)

In this group of tasks, the project from the previous section will be modified to classify epileptic and psychogenic non-epileptic seizures to suit this new classification.

- **Researching about ES and PNES (E.1):** research will be done on these two illnesses to gain more insight into their differences and the current state of research on them, in case there is any information which can be used to improve the model.
- **Finding dataset for ES and PNES (E.2):** a new EEG dataset similar to that of the project from the previous task group will be acquired to train the new model for ES and PNES classification.
- **Adjusting architecture, pre-processing and hyperparameters (E.3):** similar to A.3, but applied to the new model.
- **Running experiments (E.4):** similarly to the previous experiments, this task will make it possible to find an optimized model which classifies the new diseases with good performance. During this task, the accuracy will be compared to the state of the art as well.

Documentation and defense (D)

- **Documentation (D.1):** this task will be performed throughout the entirety of the project in parallel to the other tasks, since these will be documented constantly to make sure that no relevant part of the thesis is left undocumented by accident.
- **Thesis defense (D.2):** once D.1 is finished, the preparation for the final thesis defense will be prepared and studied.

A.1.2 Task summary

At Figure 11, a Gantt chart can be seen along with a table, summarizing the tasks which were defined during the previous section, as well as the estimated time of dedication to each and their dependencies.

Since the agile methodology was chosen for this project, the chart is also divided in 6 sprints, most associated to one task group primarily. The tasks completion will be monitored using Trello.

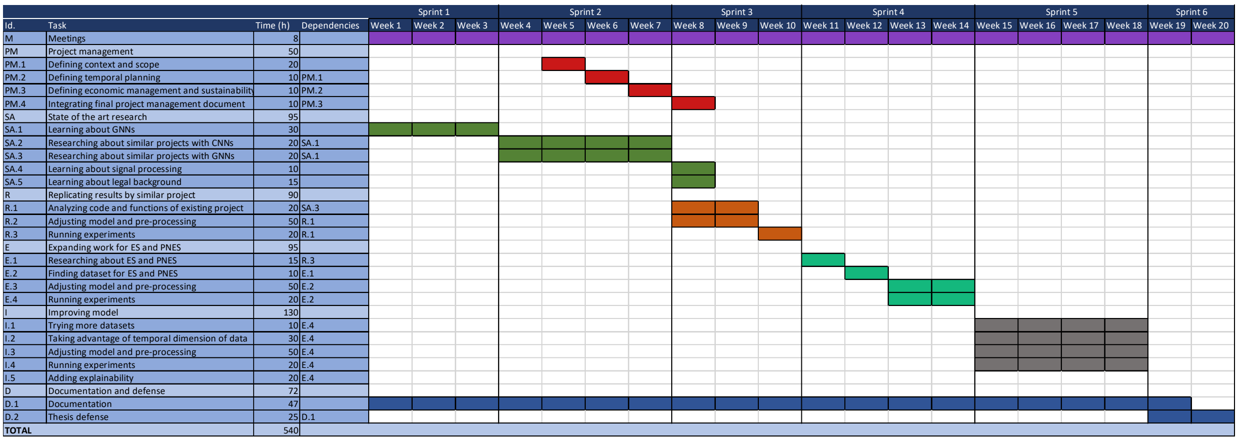


Figure 11: Gantt chart for the project. [Own creation].

A.1.3 Deviation from original planning

Originally, this project was planned to be focused on finding a GNN model which diagnosed Alzheimer's disease and improve the architecture, the preprocessing and the hyperparameters to increase the accuracy. However, the focus was quickly changed to adapting this model so that it diagnosed ES and PNES instead.

One of the reasons for this change was that more potential was seen in this new goal, since the improvement of changing an approach which already uses GNNs to solve a problem is not thought to be as significant as the improvement achieved by applying GNNs to a problem where these had not been used before, which is the case of the ES and PNES diagnosis.

The other reason is that researchers from the Mediterrania University of Reggio Calabria granted the researcher access to a significantly complete EEG dataset with patients with ES, with PNES and for healthy control, which made the focus on these illnesses far easier than expected, since access to a starting dataset was possessed.

As a consequence of this decision, the Gantt chart was affected, and some of the tasks were modified to focus more on this new objective. For example, after the state-of-the-art research, the following task groups would have focused on improving the existing model and finding more EEG datasets related to Alzheimer's disease. Some of the later tasks were meant to investigate the possibility of expanding the model for ES and PNES as well, and those were given a larger priority in the final chart.

A.1.4 Resources

On the one hand, the human resources of the project correspond to the researcher and the director, which will directly take part in the project.

On the other hand, the material resources are the following:

- **Laptop:** this laptop is a *Huawei Matebook E*, and it will be mainly used to write the documentation and for research.
- **Server:** to perform the experiments, a server will be accessed through SSH. This server uses an AMD Ryzen 3950X processor, 128 GB of DDR4 memory, an SSD hard drive of 1 TB, an internal hard drive of 8 TB, and one NVidia RTX 3090 GPU.
- **Software:** several websites and applications will be used to plan and monitor the project, such as Trello and GitHub, and a code editor will be utilized as well. All the software will be of free use.

A.2. Risk management

Another key part of the project planning is to ensure that all risks and potential obstacles have an associated alternative plan. This way, there should be no situations which critically endanger the success of the thesis.

For this reason, the following list defines the contingency plans in case any of these risks arises:

- **Deadlines:** this is one of the main obstacles this project can face, and it is also the most likely, since the planning is, as stated in the previous sections, very approximate due to the lack of information required to properly estimate some of the tasks, in addition to the fact that many of the following obstacles can increase the likelihood of running out of time. However, the project planning attempts to account for this: the required time for each sprint is overestimated on purpose so that, even in a worst-case scenario, the project can succeed in time. In addition, the agile methodology is flexible, and the times for each sprint can be adjusted as necessary.
- **Insufficient data:** this risk is another one with difficult solutions, since the lack of data is a problem which cannot be solved. However, there are two different parts of the project where new data is needed, and each can have its own alternative plan. Firstly, during the group of tasks T, more data is needed to improve the accuracy of the already existing model, and, since data of this kind has already been used to train the model, it should be possible to find more. In any case, in an extreme situation, this group of tasks could be skipped. Secondly, during group E, the model will require datasets regarding the illnesses of choice. An easy solution for this is to choose different diseases which feature enough data for the illness to be classified properly.
- **Unbalanced data:** this is another likely scenario, since datasets are rarely perfect, and some output might have a larger dataset than the rest, or vice versa, causing the model to perform badly when classifying illnesses with less data. A possibility to fix this would be to use less data from the disease or diseases with more samples to balance these out, as well as adjusting the architecture, the pre-processing and the hyperparameters to get better results.
- **Computational risk:** this risk is rather unlikely, since access to a server will be provided through SSH, which should have a good enough performance. However, in case even this server was not powerful enough, a possible solution would be to use less samples to train the model or to reduce the size of the model. Both of these solutions, however, would compromise the performance of the model, which is obviously undesirable, but this scenario is not likely enough for this to be a significant concern. In addition, if time complexity is an issue, the sprint time can be adjusted, as stated earlier.
- **Lack of knowledge about signal processing:** it would not be surprising for this issue to happen, since the researcher lacks the physics background required to deal with signal processing. However, this can be fixed by extending the time dedicated to learn about these.

B. Budget and sustainability

B.1. Budget

In this section, we will define the hypothetical roles which would be required for this project (despite the fact that all of these will be fulfilled by the tutor and the researcher) and the salaries these would receive, in addition to the amortization and indirect costs.

B.1.1 Staff costs

Firstly, in order to define the costs for the hypothetical roles in this project, it is necessary to choose these roles to begin with. The role which in reality will be carried out by the project tutor is the **Project Manager**. However, the rest of the roles will be performed by the researcher. To research about the required subjects and to run all the experiments, the project will need a **Junior Researcher**. In addition, the GNN models will be implemented and tested by a **Junior Software Developer** and a **QA Tester**. These roles and their corresponding salaries (calculated by applying social security payments to the gross salaries of these roles, by multiplying them by 1.35) are listed below on Table 6.

Role	Salary (€/h)
Project Manager	29.86 [39]
Junior Researcher	16.55 [40]
Junior Software Engineer	15.05 [41]
QA Tester	14.95 [42]

Table 6: salaries of the roles required for the project.

Secondly, to be able to calculate the budget for each of these roles, we will estimate the number of hours which will be dedicated to the tasks by each of these roles. This estimation is shown on **Error! Reference source not found.**.

Id.	Task	PrM (h)	JR (h)	JSD (h)	QAT (h)	Total (h)
M	Meetings	8	0	0	0	8
PM	Project management	0	50	0	0	50
PM.1	Defining context and scope	0	10	0	0	10
PM.2	Defining temporal planning	0	10	0	0	10
PM.3	Defining economic management and sustainability	0	10	0	0	10
PM.4	Integrating final project management document	0	10	0	0	10
PM.5	Preparing first presentation	0	10	0	0	10
SA	State of the art research	0	95	0	0	95
SA.1	Learning about GNNs	0	30	0	0	30
SA.2	Researching about similar projects with CNNs	0	20	0	0	20
SA.3	Researching about similar projects with GNNs	0	20	0	0	20
SA.4	Learning about signal processing	0	10	0	0	10
SA.5	Learning about legal background	0	15	0	0	15
A	Designing model architecture	0	35	65	20	120
A.1	Replicating results by existing project	0	0	20	0	20
A.2	Taking advantage of temporal dimension of data	0	15	15	0	30
A.3	Adjusting model and pre-processing	0	0	30	20	50
A.4	Running experiments	0	20	0	0	20
T	Training model	0	30	30	20	80
T.1	Finding more datasets	0	10	0	0	10

T.2	Adjusting model and pre-processing	0	0	30	20	50
T.3	Running experiments	0	20	0	0	20
E	Expanding work for more diseases	0	45	30	20	95
E.1	Researching about more diseases	0	15	0	0	15
E.2	Finding datasets for more diseases	0	10	0	0	10
E.3	Adjusting model and pre-processing	0	0	30	20	50
E.4	Running experiments	0	20	0	0	20
D	Documentation and defense	0	92	0	0	92
D.1	Documentation	0	67	0	0	67
D.2	Thesis defense	0	25	0	0	25
TOTAL		**8**	**347**	**125**	**60**	**540**

Table 7 Lastly, the total staff cost of the project will be calculated by multiplying the total hours of dedication of each role by its corresponding salary. The final results are shown on Table 8.

B.1.2 Generic costs

B.1.2.1 Amortization

To calculate the amortization of the project's material resources, only the hardware resources will be taken into account, as all the software which will be used is of free use.

The approximate amortization will be calculated with this formula:

$$Amortization = \frac{Cost}{N_{years\ lifespan} \cdot N_{days\ of\ work} \cdot N_{hours\ per\ day}} N_{hours\ of\ use}$$

Two hardware resources will be used: a laptop (*Huawei Matebook E*) and a server (*AMD Ryzen 3950X processor*). The server will only be used to run the experiments (60 hours of use), while the laptop will be used for every other task (480 hours). In addition, the project is expected to last for 140 days with 3.75 hours of work per day. Knowing this, the amortization values for these two resources can be calculated, as they are shown on Table 9.

Id.	Task	PrM (h)	JR (h)	JSD (h)	QAT (h)	Total (h)
M	Meetings	8	0	0	0	8
PM	Project management	0	50	0	0	50
PM.1	Defining context and scope	0	10	0	0	10
PM.2	Defining temporal planning	0	10	0	0	10
PM.3	Defining economic management and sustainability	0	10	0	0	10
PM.4	Integrating final project management document	0	10	0	0	10
PM.5	Preparing first presentation	0	10	0	0	10
SA	State of the art research	0	95	0	0	95
SA.1	Learning about GNNs	0	30	0	0	30
SA.2	Researching about similar projects with CNNs	0	20	0	0	20
SA.3	Researching about similar projects with GNNs	0	20	0	0	20
SA.4	Learning about signal processing	0	10	0	0	10
SA.5	Learning about legal background	0	15	0	0	15
A	Designing model architecture	0	35	65	20	120
A.1	Replicating results by existing project	0	0	20	0	20
A.2	Taking advantage of temporal dimension of data	0	15	15	0	30
A.3	Adjusting model and pre-processing	0	0	30	20	50
A.4	Running experiments	0	20	0	0	20
T	Training model	0	30	30	20	80
T.1	Finding more datasets	0	10	0	0	10
T.2	Adjusting model and pre-processing	0	0	30	20	50
T.3	Running experiments	0	20	0	0	20
E	Expanding work for more diseases	0	45	30	20	95
E.1	Researching about more diseases	0	15	0	0	15
E.2	Finding datasets for more diseases	0	10	0	0	10
E.3	Adjusting model and pre-processing	0	0	30	20	50
E.4	Running experiments	0	20	0	0	20
D	Documentation and defense	0	92	0	0	92
D.1	Documentation	0	67	0	0	67
D.2	Thesis defense	0	25	0	0	25
TOTAL		**8**	**347**	**125**	**60**	**540**

Table 7: hours dedicated to every task by each role. [Own creation].

Role	**Cost (€)**
Project Manager	176.96
Junior Researcher	4499.42
Junior Software Engineer	1170.75
QA Tester	717
TOTAL	**6564.13**

Table 8: cost of every role. [Own creation].

Resource	**Cost (€)**	**Lifespan (years)**	**Time of use (h)**	**Amortization (€)**
Laptop	999 [43]	4 [45]	480	228.34
Server	698.22 [44]	3 [45]	60	26.60
TOTAL				**254.94**

Table 9: amortization of each hardware resource. [Own creation].

B.1.2.2 Indirect costs

Besides the cost of the project's resources, some additional costs are necessary to be able to fulfill all the goals successfully.

- **Internet cost:** the Internet cost for the researcher is approximately 37 €/month. Since this project is expected to last for 5 months, this cost would be **185 €** in total.
- **Electricity cost:** to calculate the estimated electricity cost for this project, the monthly electricity cost for the researcher will be used as a reference. In this case, this cost is around 70 €/month, so the final cost will be **350 €**.

Therefore, the total indirect cost is **535 €**.

B.1.2.3 Contingency cost

Finally, despite all the elements taken into account, there can always be unexpected issues (such as the need to extend the project time, which would logically increase the time of dedication and, therefore, all the costs). To avoid running out of budget in such cases, a contingency budget will be added, corresponding to 15% of the total cost. That is:

$$\begin{aligned} C_{contingency} &= 15\% \cdot \left(C_{staff} + C_{amortization} + C_{indirect}\right) \\ &= 15\% \cdot (6564.13\ € + 254.94\ € + \ 535\ €) = \mathbf{1103.11}\ € \end{aligned}$$

B.1.2.4 Deviation cost

Several risks were stated earlier in this document. However, these are not expected to cause an increment in the budget, since the solutions either would not require more money, or the amount is completely unknown (such as the deadline risk). However, the latter case would be covered by the contingency cost.

B.1.3 Total budget of the project

The total cost of the project considering all the previous elements is:

$$\begin{aligned} C_{total} &= C_{staff} + C_{amortization} + C_{indirect} + C_{contingency} \\ &= 6564.13\ € + 254.94\ € + \ 535\ € + 1103.11\ € = \mathbf{8457.18}\ € \end{aligned}$$

B.1.4 Management control

Despite all the measures taken during the previous sections, it is very likely that the final (hypothetical) cost deviates from the expected budget, either due to an underestimation or an overestimation. To control this, the following values will be used:

- **Estimated budget:** the previously calculated value for the project during a period of time. This estimation may vary throughout the thesis.
- **Real cost:** this is calculated similarly to the budget, but with known values for the current window of time the project has been worked on.
- **Deviation:** the difference between the estimated budget and the real cost for a specific period. This value can help adjust the estimated budget as necessary by analyzing what caused the discrepancy in case the budget was underestimated. Otherwise, the overestimated budget would be kept to prevent unexpected problems.

B.2. Sustainability

B.2.1 Self-assessment

Most of the prior knowledge about project sustainability by the researcher is associated to the social impact of machine learning, since the existence of different types of biases (gender, ethnicity, etc.) the provided datasets can include, as well as the negative impact an inaccurate or badly tested model can have on society in the case of widespread use, especially if it is used without supervision or for critical decisions. Besides this, machine learning can contribute positively to society by providing useful services which improve the quality of life of some people (such as this very project).

In addition, some of that knowledge corresponds to the environmental aspect of artificial intelligence, since training some models can require vast amounts of energy, and that can logically have a significant negative impact on the planet. However, the researcher has also participated in a project promoting city circularity, consisting on the concept and the user experience design of an app with the purpose of distributing recycling building materials, while also providing machine learning tools to help users. Purposes like this can be highly helpful for the environment and, therefore, for environmental sustainability.

However, the researcher was not very aware of the existence of the economic aspect of sustainability before starting this project, and despite knowing several methods related to the economic viability of a business, he had not applied these concepts yet, and he had not tied them to the concept of sustainability. On top of that, he lacks knowledge of many of the existing sustainability indicators and services, not only of the economic dimension, but also of the environmental and social dimensions (such as the carbon footprint, the ISO 26000, etc.).

In conclusion, while there are concepts known beforehand about sustainability (especially those related to the field of computer science), there are many things that are yet unknown by the researcher, many of which will be learned or consolidated thanks to the realization of this project.

B.2.2 Environmental dimension

Unfortunately, this project does not particularly contribute much to the environmental aspect of sustainability, since it is rather oriented towards a social context, so it does not bring anything noticeably positive to the environment.

However, it can certainly have a negative impact on it, as it requires a considerable usage of computing resources to train models and run experiments on these. The particular impact is unknown, as the exact electricity consumption and the appropriate indicators are unknown, but sadly there is not much that can be done to prevent this, since model fitting is a crucial part of this project and it cannot be avoided.

B.2.3 Economic dimension

On the other hand, this project can bring considerable economic benefits to society. In fact, this is one of the main motivations behind the thesis itself: the use of GNNs to help diagnose mental illnesses can help greatly reduce the cost for medical centers, since it is significantly cheaper than hiring an expert, and also quicker, so this project could save not only time, but also money. Besides that, private machine learning companies or insurances could benefit from the use of the resulting model by offering its service to diagnose cases.

While the creation of this project involves several costs (electricity to train models, the acquisition of the datasets by medical centers through EEGs, etc.), these are considered to be covered by the extensive benefits this thesis might bring.

B.2.4 Social dimension

Naturally, this is the dimension of sustainability which is benefitted the most by this project, since it is one of its primary motivations. The use of a model to diagnose mental diseases through an accurate model could directly save the lives of people suffering from these by providing not only a more accurate diagnose, but also a quicker one, and illnesses of this kind often require an immediate response. In addition, individuals who do not suffer from these but believe they might will get to be relieved earlier by getting a more immediate result.

However, some social aspects must be considered before putting the resulting model into practice, since the use of machine learning on medical fields is strictly regulated in some countries, so these regulations will have to be researched prior to the distribution of this model. Additionally, putting legal issues aside, there may be ethical problems with its use depending on how it is done, since relying solely on the model and nothing else could cause both false positive and false negative diagnoses, both of which could deeply harm peoples' lives. For this reason, the use of this model should be performed carefully and mindfully.

Finally, regarding the personal growth acquired by the researcher after finishing this thesis, this project will naturally grant extensive knowledge about neural networks and this particular application of them, but it will also help the researcher know more about sustainability, and it will also be a positive and helpful introduction to projects of this size regarding deep learning.

Glossary

- **Epileptic Seizures (ES):** phenomenon which causes a person's neurons to send an excessively high rate of signals, which causes the victim to lose control of their body and their perception of their surroundings, often convulsing and becoming unconscious. This loss of control is known as a seizure.
- **Psychogenic Non-Epileptic Seizures (PNES):** type of seizure which, unlike epileptic seizures, is not caused by an abnormal rate of neural signals, but by psychological causes such as trauma. These seizures cannot be treated with drugs, and the only known method to cure or improve these is therapy.
- **Graphs:** data structure represented by a set of nodes (or vertices) which are connected by a set of edges. These can store attributes associated to each node and edge, and sometimes even to the graph itself.
- **Electroencephalograms (EEGs):** recordings of the small electrical signals emitted by the brain, which are acquired through some electrodes placed around the subject's head.
- **Neural networks:** computational machine learning model designed to mimic biological neurons formed by nodes called perceptrons. These take several input values and compute an output value out of these. These can usually be trained to predict values from a given input.
- **Graph Neural Networks (GNNs):** type of neural network which take a graph as input and take advantage of the relations between nodes and the different attributes and perform several operations on this graph in order to fulfill a task, which can be a node-level task (predicting attributes of nodes), an edge-level task (predicting data of edges) or a graph-level task (predicting attributes about the entire graph).
- **Convolutional Neural Networks (CNNs):** this is a particular case of GNNs where all the nodes belong to a 2-dimensional space, and their adjacency is based on their position.

Bibliography

[1] *Epilepsy and Seizures* [Internet]. National Institute of Neurological Disorders and Stroke. [cited 2023May26]. Available from: https://www.ninds.nih.gov/health-information/disorders/epilepsy-and-seizures

[2] *Epilepsy* [Internet]. World Health Organization. 2023 [cited 2023May26]. Available from: https://www.who.int/news-room/fact-sheets/detail/epilepsy

[3] *Electroencephalography (EEG)* [Internet]. Chandan Diagnostic. [cited 2023Jun19]. Available from: https://chandandiagnostic.com/ourservices/electroencephalography

[4] Tavory T. *The new EU artificial intelligence proposal and its effect on medical device regulation* [Internet]. Yigal Arnon. 2021 [cited 2023Mar20]. Available from: https://www.arnon.co.il/en/content/new-eu-artificial-intelligence-proposal-and-its-effect-medical-device-regulation

[5] Lo Giudice, M. et al. *Convolutional neural network classification of rest EEG signals among people with epilepsy, psychogenic non epileptic seizures and control subjects* [Internet]. Multidisciplinary Digital Publishing Institute. 2022 [cited 2023May26]. Available from: https://www.mdpi.com/1660-4601/19/23/15733

[6] Ávila Martínez D. *Graph Neural Networks for Electroencephalogram Analysis* [thesis]. Barcelona: Universitat Politècnica de Catalunya; 2022 [cited 2023Feb28].

[7] iWorx Systems, Inc. *The Electroencephalogram (EEG)* [Internet]. Rogers State University; 2013 [cited 2023Feb28]. Available from: https://www.rsu.edu/wp-content/uploads/2015/06/TheElectroencephalogramEEGCorticalArousal.pdf

[8] *Greek Alphabet Soup – Making Sense of EEG Bands* [Internet]. NeuroSky. 2023 [cited 2023Jun11]. Available from: https://neurosky.com/2015/05/greek-alphabet-soup-making-sense-of-eeg-bands/

[9] *Kumar Saini M. What is Power Spectral Density?* [Internet]. TutorialsPoint. 2022 [cited 2023Jun11]. Available from: https://www.tutorialspoint.com/what-is-power-spectral-density

[10] Biltoft C. A. et al. *Spectral Coherence and the Statistical Significance of Turbulent Flux Computations* [Internet]. AMS. 2009 [cited 2023Jun11]. Available from: https://journals.ametsoc.org/view/journals/atot/26/2/2008jtecha1141_1.xml

[11] Turney S. *Pearson Correlation Coefficient (r) | Guide & Examples* [Internet]. Scribbr. 2022 [cited 2023Jun11]. Available from: https://www.scribbr.com/statistics/pearson-correlation-coefficient/

[12] Subramaniyam N. P. *EEG Connectivity Using Phase Lag Index* [Internet]. Sapien Labs. 2019 [cited 2023Jun11]. Available from: https://sapienlabs.org/lab-talk/eeg-connectivity-using-phase-lag-index/

[13] Stam C.J. et al. *Phase lag index: Assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources* [Internet]. Wiley Online Library. 2007 [cited 2023Jun11]. Available from: https://onlinelibrary.wiley.com/doi/epdf/10.1002/hbm.20346

[14] Kröse B, Van der Smagt P. *An introduction to Neural Networks* [Internet]. UVA. 1996 [cited 2023Feb28]. Available from: https://www.infor.uva.es/~teodoro/neuro-intro.pdf

[15] Elijah Koech K. *Cross-Entropy Loss Function* [Internet]. Medium. 2020 [cited 2023Jun11]. Available from: https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e

[16] Brownlee J. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning* [Internet]. Machine Learning Mastery. 2017 [cited 2023Jun11]. Available from: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[17] Saha S. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way* [Internet]. SaturnCloud. 2018. [cited 2023Jun19]. Available from: https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/

[18] Sanchez-Lengeling B, Reif E, Pearce A, Wiltschko AB. *A gentle introduction to graph neural networks* [Internet]. Distill. 2021 [cited 2023Feb28]. Available from: https://distill.pub/2021/gnn-intro/

[19] Scarselli F. et al. *The Graph Neural Network Model* [Internet]. IEEE. 2009. [cited 2023Jun19]. Available from: https://ieeexplore.ieee.org/document/4700287

[20] Veličković P. et al. *Graph Attention Networks* [Internet]. Arxiv. 2017. [cited 2023Jun19]. Available from: https://arxiv.org/abs/1710.10903

[21] Akhil. *An Introduction to Graph Attention Networks* [Internet]. Medium. 2020. [cited 2023Jun20]. Available from: https://medium.com/@eakhil711/an-introduction-to-graph-attention-networks-d41ed52e5b1e

[22] Kim D. et al. *How to Find Your Friendly Neighborhood: Graph Attention Design with Self-Supervision* [Internet]. OpenReview. 2021. [cited 2023Jun20]. Available from: https://openreview.net/forum?id=Wi5KUNlqWty

[23] Shi Y. et al. *Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification* [Internet]. Arxiv. 2020. [cited 2023Jun20]. Available from: https://arxiv.org/abs/2009.03509

[24] Vaswani A. et al. *Attention Is All You Need* [Internet]. Arxiv. 2017. [cited 2023Jun20]. Available from: https://arxiv.org/abs/1706.03762

[25] Doshi K. *Transformers Explained Visually (Part 2): How it works, step-by-step* [Internet]. Medium. 2021. [cited 2023Jun20]. Available from: https://towardsdatascience.com/transformers-explained-visually-part-2-how-it-works-step-by-step-b49fa4a64f34

[26] Tang S. et al. *Modeling Multivariate Biosignals With Graph Neural Networks and Structured State Space Models* [Internet]. Arxiv. 2022 [cited 2023May26]. Available from: https://doi.org/10.48550/arXiv.2211.11176

[27] Rahmani A. et al. *A Meta-GNN approach to personalized seizure detection and classification* [Internet]. ResearchGate. 2022 [cited 2023Jun11]. Available from: https://www.researchgate.net/publication/365181025_A_Meta-GNN_approach_to_personalized_seizure_detection_and_classification

[28] Ahmadi N. et al. *EEG-based classification of epilepsy and PNES: EEG microstate and functional brain network features* [Internet]. Springer Open. 2020 [cited 2023Jun11]. Available from: https://braininformatics.springeropen.com/articles/10.1186/s40708-020-00107-z

[29] Lo Giudice, M. et al. *Permutation Entropy-Based Interpretability of Convolutional Neural Network Models for Interictal EEG Discrimination of Subjects with Epileptic Seizures vs.*

Psychogenic Non-Epileptic Seizures [Internet]. National Library of Medicine. 2022 [cited 2023May26]. Available from: https://pubmed.ncbi.nlm.nih.gov/35052128/

[30] Zsom A. et al. *Ictal autonomic activity recorded via wearable-sensors plus machine learning can discriminate epileptic and psychogenic nonepileptic seizures* [Internet]. National Library of Medicine. 2022 [cited 2023May26]. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7821908/

[31] Cura O. K. et al. *Epileptic EEG Classification Using Synchrosqueezing Transform with Machine and Deep Learning Techniques* [Internet]. IEEE. 2021 [cited 2023May26]. Available from: https://ieeexplore.ieee.org/document/9287347

[32] Varonne G. et al. *A Comprehensive Machine-Learning-Based Software Pipeline to Classify EEG Signals: A Case Study on PNES vs. Control Subjects* [Internet]. National Library of Medicine. 2020 [cited 2023May26]. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7071461/

[33] Li Z. et al. *Graph-generative neural network for EEG-based epileptic seizure detection via discovery of dynamic brain functional connectivity* [Internet]. Scientific reports. 2022 [cited 2023May26]. Available from: https://www.nature.com/articles/s41598-022-23656-1

[34] Boonyakitanont P. et al. *ScoreNet: A Neural Network-Based Post-Processing Model for Identifying Epileptic Seizure Onset and Offset in EEGs* [Internet]. National Library of Medicine. 2021 [cited 2023May26]. Available from: https://pubmed.ncbi.nlm.nih.gov/34797766/

[35] Brari Z. *A Novel Machine Learning Model for the Detection of Epilepsy and Epileptic Seizures Using Electroencephalographic Signals Based on Chaos and Fractal Theories* [Internet]. Mathematical Problems in Engineering. 2021 [cited 2023May26]. Available from: https://www.hindawi.com/journals/mpe/2021/2107113/"

[36] Chen Z. *Automatic seizure detection based on Machine Learning and EEG* [Internet]. DTU. 2022 [cited 2023Jun11]. Available from: https://upcommons.upc.edu/bitstream/handle/2117/386204/174158.pdf?sequence=2&isAllowed=y

[37] Ying R. et al. *GNNExplainer: Generating Explanations for Graph Neural Networks* [Internet]. Arxiv. 2019 [cited 2023Jun11]. Available from: https://arxiv.org/abs/1903.03894

[38] Ivano Triggiani A. *Diagram showing the placement of the 19 scalp electrodes used for the present electroencephalographic (EEG) recordings* [Internet]. ResearchGate. [cited 2023Jun11]. Available from: https://www.researchgate.net/figure/Diagram-showing-the-placement-of-the-19-scalp-electrodes-used-for-the-present_fig1_295672882

[39] Glassdoor. *Salary: Project manager in Madrid, Spain 2023* [Internet]. Glassdoor. 2023 [cited 2023Mar13]. Available from: https://www.glassdoor.com/Salaries/madrid-project-manager-salary-SRCH_IL.0,6_IM1030_KO7,22.htm

[40] Glassdoor. *Salary: Junior researcher (march, 2023)* [Internet]. Glassdoor. 2023 [cited 2023Mar13]. Available from: https://www.glassdoor.com/Salaries/spain-junior-researcher-salary-SRCH_IL.0,5_IN219_KO6,23.htm

[41] Payscale. *Average junior software engineer salary in Spain* [Internet]. PayScale. 2023 [cited 2023Mar13]. Available from: https://www.payscale.com/research/ES/Job=Junior_Software_Engineer/Salary

[42] Glassdoor. *Salary: Qa Tester* (March, 2023) [Internet]. Glassdoor. 2023 [cited 2023Mar13]. Available from: https://www.glassdoor.com/Salaries/spain-qa-tester-salary-SRCH_IL.0,5_IN219_KO6,15.htm

[43] McLellan C, editor. *Huawei Matebook E (2022) review: An impressive surface pro 8 alternative with a top-quality OLED screen* [Internet]. ZDNET. 2022 [cited 2023Mar13]. Available from: https://www.zdnet.com/article/huawei-matebook-e-2022-review/

[44] Knapp M. *AMD Ryzen 9 3950X review* [Internet]. TechRadar. 2019 [cited 2023Mar13]. Available from: https://www.techradar.com/reviews/amd-ryzen-9-3950x

[45] Walter D. *Signs it's time to buy a new PC* [Internet]. Business News Daily. 2023 [cited 2023Mar13]. Available from: https://www.businessnewsdaily.com/65-when-to-replace-the-company-computers.html