Date of publication DEC-31, 2025, date of current version MAY-29, 2025
www.computingonline.net/ computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.24.4.4340

Use of some variants of Graph Neural Networks for the classification of Electroencephalograms of schizophrenia

PELAGIE FLORE TEMGOUA NANFACK[1,2], ELIE FUTE TAGNE[1,3]
[1]University of Buea, Faculty of Engineering and Technology , Buea, Cameroon
[2]Ministry of Scientific Research and Innovation, National Committee for Development of Technologies, Yaounde, Cameroon
[3]University of Dschang, Faculty of Science, Dschang, Cameroon

Corresponding author: Pelagie Flore Temgoua Nanfack (e-mail: pelagietemgoua@gmail.com).

ABSTRACT Graph Neural Networks (GNN) are a form of neural network that have shown their added value on non-Euclidean data such as electroencephalograms (EEG). The aim of this work was to evaluate some GNN variants for the classification of schizophrenia EEGs; namely Graph Convolution Network (GCN) and Graph of Attention (GAT). For this purpose, the Pycaret tool and a Convolutional Neural Network (CNN) were used for the classification of the obtained graphs. This made it possible to compare GCN and GAT associated with Pycaret on the one hand, and GCN and GCN coupled with a Convolutional Neural Network (CNN) for classification on the other. It was found that GCN and GAT offer accuracy of 85% and 80% respectively. GCN coupled with a CNN offers 80% accuracy. However, despite the fact that GAT is resource intensive, its confusion matrix shows that it offers better sensitivity and specificity than the other methods. In other words, the error rate is 5%.

KEYWORDS Graph Neural Network; Classification; Electroencephalogram; Schizophrenia.

I. INTRODUCTION

The electroencephalogram (or EEG) is a test that measures the brain's electrical activity through its different brain regions. EEG and Magnetic Resonance Imaging (MRI) are the two most commonly used techniques in cognitive science [1]. However, EEG is much more popular than MRI because it is inexpensive, readily available, easy to perform and monitor when follow-up is needed, and minimally invasive for subjects suitable for study. The classification of EEGs is a complex process due to, for example, the number of channels that may differ from one collection source to another and especially the non-linear placement of these channels. It requires thorough analysis and accurate modeling techniques. Neural networks can be used to classify EEGs, but they are generally limited to relatively simple data sets such as images and text. Reason why an EEG is represented in a non-Euclidean space, whereas an image and text are represented in a Euclidean space Figure 1. A non-Euclidean space is a space in that cannot find a set of coordinates which are mutually perpendicular, where the coordinate lines are all parallel to each other and where each grid square has the same area. More specifically, it is the study of geometry on surfaces which are not flat [2] [3]. Graph Neural Networks (GNN) are powerful tools for studying non-Euclidean domains [4] [5].

GNNs are used to generalize the concept of convolution in graphs [6]. Figure 2 shows the structure of a GNN compared to a Convolution Neural Network (CNN).

Graph Neural Networks are a new class of neural networks that can be used to process more complex data sets, such as EEGs. Schizophrenia is a multidimensional disorder with several subtypes, different neurobiological underpinnings [7]. The use of GNN variants to classify electroencephalogram (EEG) recordings of patients with schizophrenia is a rapidly expanding field. GNNs are known to improve accuracy and prediction time when the data is

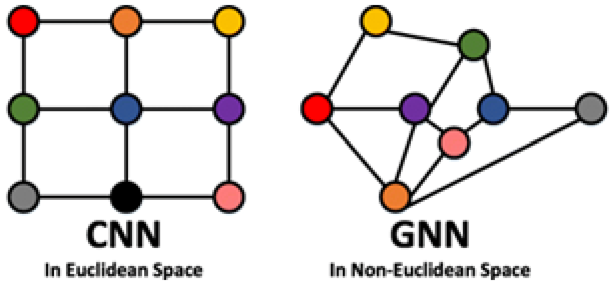


Figure 1. Representation of an image and text in contrast to an EEG.

non-linear [8]. GNN is a form of deep learning that can be applied in a wide range of fields, including medicine and bioinformatics [9]. For classifying schizophrenia EEGs, GNNs could provide a more accurate and faster method than other classification methods [10]. GNNs can capture complex patterns in the data, which means that they can be used to identify patterns in underlying psychiatric disorders. GNNs improve the diagnostic accuracy and learning speed of existing approaches [11]. GNNs can be integrated with computer systems to analyze patients' EEG values in real time and identify abnormalities and changes in their EEGs and provide an early diagnosis [12]. The general design pipeline for a GNN model contains three (03) calculation steps which are: the propagation module, sampling module et pooling module [9]. The propagation module that generates the graphs that will be considered as input for the learning system to be set up is composed of several variants, including Graph Convolution Network (GCN), Graph Attention Networks (GAT), Gated Graph Neural Networks (GGNN) Figure 3, etc.

The work of Jie Zhou et al. [9] presents a literature review of the different GNN variants and some of their applications. However, the applications mentioned do not concern schizophrenia or other brain or psychiatric diseases. Nevertheless, the applications mentioned do not concern schizophrenia or other brain or psychiatric diseases. In this paper, we will explore the potential advantages of using some variants of Graph Neural Networks (GNN) for EEG classification, namely GAT, and GNN. We will compare the performance of each of these variants on the one hand and GNN coupled with a Convolutional Neural Network on the other hand. We will discuss the advantages and disadvantages of each variant and how they can be applied to improve prediction accuracy. We will also compare the performance of the different variants and determine the best method for the classification of schizophrenia EEGs. Other variants could be studied in future work.

II. RELATED WORKS

Several other works have been done about the use of Artificial Intelligence for classification of EEG of schizophrenia. Some use Machine Learning (ML) others use Deep learning (DL). We started our work by [13] and [14]. The present work aims to have the best possible accuracy consider that EEGs can be collected from different sources, hence the different number of channels and the different frequencies of values. Most of these works are more interested in the use of Convolutional Neural Networks for the diagnosis of schizophrenia. As Shu Lih et al. [15] which uses deep convolutional neural network to set up a method for the diagnosis of schizophrenia with accuracies of 98.07% and 81.26% for non-subject based testing and subject based testing, respectively. With the same method except for a few parameters, Ahmad Shalbaf et al. [16] achieved accuracy, sensitivity and specificity of 98.60%, 99.65% and 96.92% respectively. Several other works using CNN complete classification accuracies of over 90% [17] [18] [19] and [20]. Rahul et al. [21] provide a summary of the ML and DL methods used to diagnose schizophrenia using EEGs, presenting the pre-processing methods used, the databases adopted and the accuracies obtained. On the other, the use of GNNs for EEG classification has been also discussed by several authors. GNNs have become popular in recent years and are used in many fields including medicine and are considered to be an effective alternative to traditional methods such as Convolutional Neural Networks (CNNs) [22]. GNNs have been applied to the classification of EEGs in various ways, for example, to discern signals from noise and identify abnormal activity [23]. They support a medical diagnosis by graphing medical data, such as brain electrical activity, anatomical structures, and digital pathologies [4] [5]. GNNs have proven to be a powerful tool for classifying EEGs, as they can identify subtle changes in the EEG signal [24]. In the same vein, Haifeng Li et al. have shown that GNNs can outperform traditional methods in EEG classification due to their ability to capture subtle correlations in the data [23] [25]. GNNs have also been used to select EEG channels for use in portable EEG headsets, indicating their potential for use in EEG classification tasks [23]. According to Yimin Hou et al., GNNs can improve the decoding performance of raw EEG signals during different types of motor imaging tasks [26]. Neeraj Wagh et al. have shown that GNNs can also be used to detect abnormalities in the EEG signal and to predict the presence of certain neurological diseases [27]. Concerning the use of GNNs for classifications of schizophrenia EEGs, very few authors have looked into it. Lei Du et al [28] in their work has shown that GNNs including Graph Convolution Networks (GCN) offer higher classification accuracy (85%) compared to simple Support Vector Machine (SVN) (80%). In the same way, Sanjay Ghosh et al. [29] found a higher F1 score with GCN compared to conventional SVN. However, most of this work does not give details on other parameters such as the confusion matrix, and the Receiver Operating Characteristic (ROC) curve. Indeed, in precision medicine, sensitivity and specificity, i.e., the precision matrix, should be one of the most important parameters, because an algorithm can have a very good precision but show too many errors in the

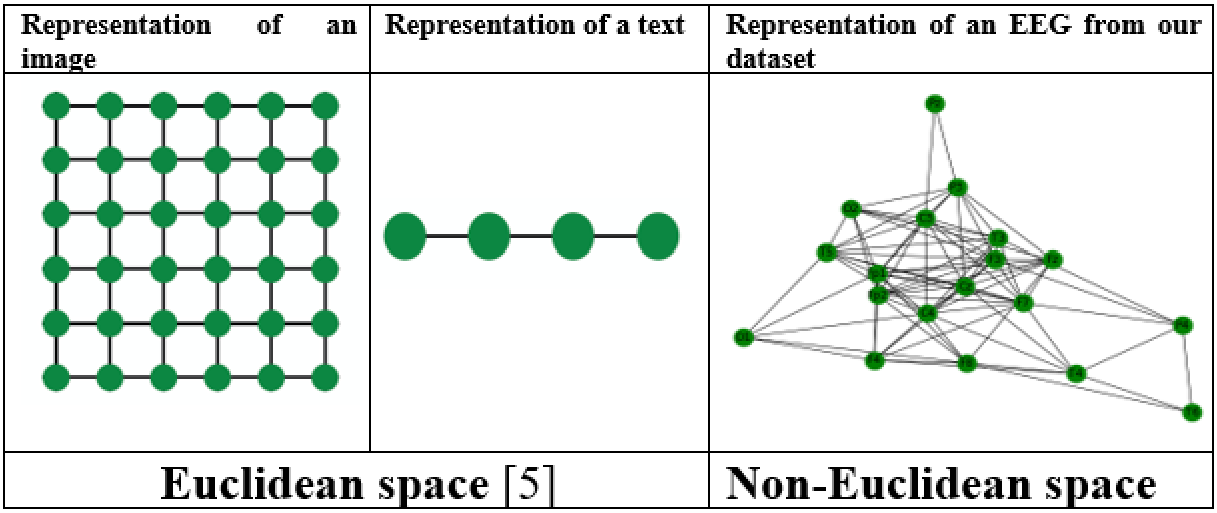

Representation of an image
Representation of a text
Representation of an EEG from our dataset
Euclidean space [5]
Non-Euclidean space

Figure 2. Comparison of CNN and graph Neural Network (GNN).

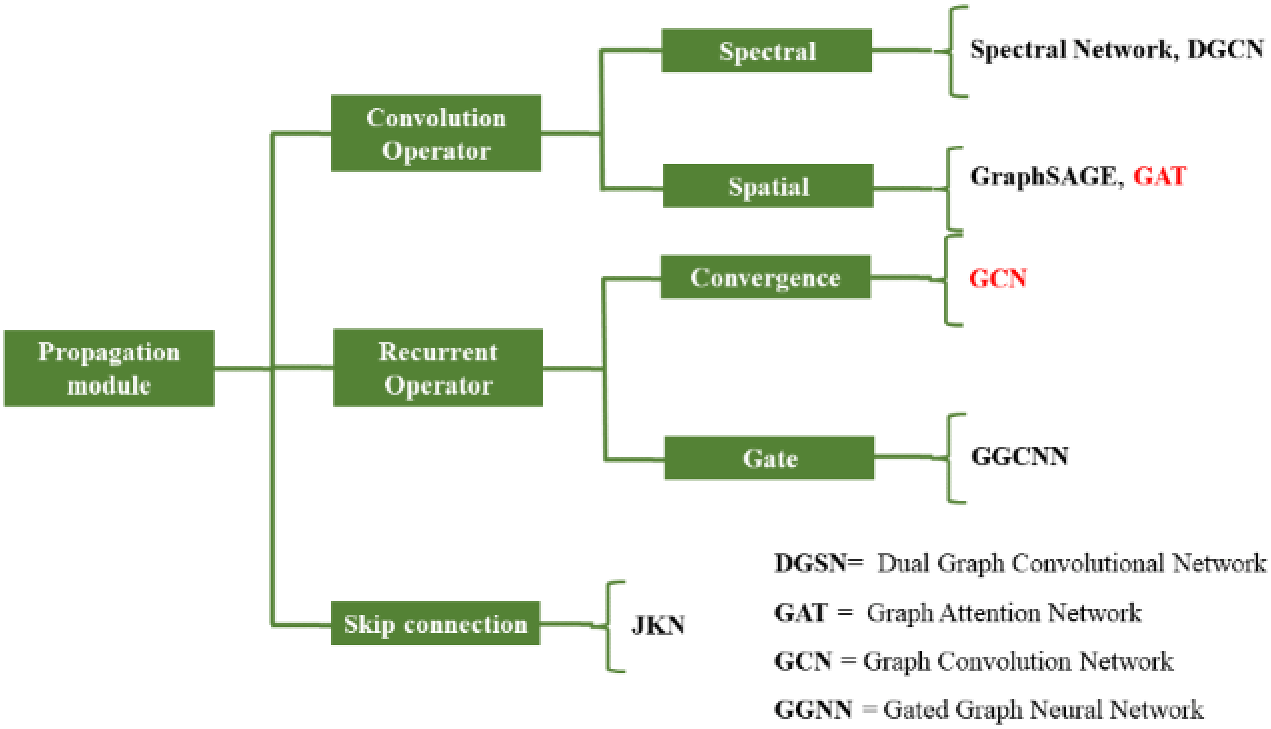


Figure 3. Presentation of some variants of GNN.

confusion matrix. Some that take these parameters into account omit the complexity of the EEG structure (graph). Klepl et al. [30] propose a survey of wide range of methods used to design GNN-based classifiers.

III. MATERIALS AND METHOD

A. MATERIALS

The dataset used here is [31] and downloaded on Repository for Open Data [32]. It includes 14 patients (7 men: 27.9 ± 3.3 years, 7 women: 28.3 ± 4.1 years) with paranoid schizophrenia, who were hospitalized at the Institute of Psychiatry and Neurology in Warsaw, Poland, and 14 healthy controls (7 men: 26.8 ± 2.9,7 women: 28.7 ± 3.4 years).

EEGs were recorded at a frequency of 250 Hz using the standard 10-20 EEG montage with 19 EEG channels: Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T6, O1, O2. The analysis of the EEGs was done using the MNE library [33] of the python language. Anaconda's Jupiter Lab Integrated Development Environment (IDE) is the programming platform used for this work. Pycaret [34]) is the tool that was used to train for the classification of EEGs. Indeed, this tool supports supervised learning tasks (classification and regression), clustering, anomaly detection, and natural language processing. With pycaret, you spend less time coding and more time on analysis. The analysis includes Exploratory Data Analysis, Data Preprocessing, Model Training, Model Explainability, and Machine Learning Operations (MLOps). PyCaret is a low-code machine-learning library in Python that aims to reduce the time needed to experiment with different machine-learning models [34] [35]. It helps Data Scientists perform all end-to-end experiments quickly and more efficiently. PyCaret being a low-code library, it allows to be more productive. The computer on which the work was done has the following characteristics:

- Processor: Intel(R) Core (TM) i7-4510U CPU @ 2.00GHz (4CPUs), 2.6GHz
- RAM : 12 GB
- Graphic card : NVIDIA GEFORCE

B. METHOD

There are several variants of GNN. These include Graph Convolution Network (GCN or GraphConv), Relational Graph Convolution (RelGraphConv), Graph Attention Network (GAT), Simplifying Graph Convolution (SGConv), etc [36]. Figure 2 gives an overview of these methods. However, only GCN and GAT will be discussed in this paper. Indeed, the most widely used GNNs in EEG classification are convolutional [37]. These networks use convolutional layers to learn hierarchical features from the raw EEG signal and then make predictions about the subject's condition. The advantage of convolutional GNNs is that they can learn complex spatial topologies from the EEG signal and can be used to identify subtle changes in the EEG signal which are indicative of changes in the patient's condition [38]. Kipf et al. [39] and Rishabh Anand [40] explain the functional mode. For GAT, Andac Demir et al. [6] have shown that it is effective for the classification of EEGs. GAT uses attention mechanisms to identify important connections in the EEG signal [41] and can be used to classify different types of brain activity [40] [42]. These last quotes present the principle of this method in more detail. The classical classification process using GNNs is shown in Figure 4.

The methodology applied is shown in Figure 5. and consists of three (03) modules: the collector, the analyzer, and the classifier.

1) The collector

This module is a preprocessing stage. Before anything, it is important to mention that collected EEGs are performed using thirty second segments without artefacts (i.e. eye movements, cardiac activity, muscle contractions). Then, the signals of each EEG channel were filtered using a Butterworth filter of order 2 in the following physiological frequency bands: 2–4Hz (delta), 4.5–7.5 Hz (theta), 8–12.5 Hz (alpha), 13–30 Hz (beta), 30–45 Hz (gamma). This module is responsible for:

- Collect EEGs (.edf file)
- Read EEG using "‘ mne.io.read_raw "‘ function of the MNE python library
- Remove artifact
- The preprocessed result is forward to the "‘Analyzer"’ module

2) The analyzer

Once we get the result obtained from the "‘Collector"’ module, this section consists on three (03) main operations:

- Use Skip-Gram algorithm, particularly Word2Vec method [44] [45] to build a graph from the nodes of EEG, take turns each node as the center node. The graph is built based on some parameter that the most important is window size. Consider an array of nodes N, if N(i) is the input (center node), then N(i-2), N(i-1), N(i+1), and N(i+2) are the context nodes if the sliding window size is equal to 2. Basically, ie when window size is equal to 1, the context $C(N_i)$ of a node N_i is set of nodes adjacent to i. The Skip-Gram model aims to produce an output probability distribution vector given a target node input. This probability distribution vector (which sums to 1) reflects the probability for each node to appear in the target node's context window. As one might expect, the probability to be high for nodes who share the same context and low for nodes who don't. Figure 7 shows an example of the graph obtained after the skip-gram operation. The obtained adjacency matrix M is formally defined as: for $N = N_1 \dots N_n$, where n is the number of nodes of nodes of initial graph (EEG), M is the co-occurrence matrix of the vocabulary N if
$$\forall (N_i; N_j) \in N$$
, $M_{i;j} = 1$ if $N_i \in C(N_j)$, otherwise $M_{i;j} = 0$ [46]. Edge weights used here are fixed for GCN and learnable for GAT, based on [39], [42].
- GNN through its variants (GAT and GCN) is used to reduce the obtained graph by a convolution series follows the steps in a CNN except that the convolution operation is actually an aggregation. The variant of GNN (GAT or GCN) chosen allows a group of nodes and how they interact with each other to be described. Aggregation protects the integrity of a collection of nodes by defining a single control point/node, called the

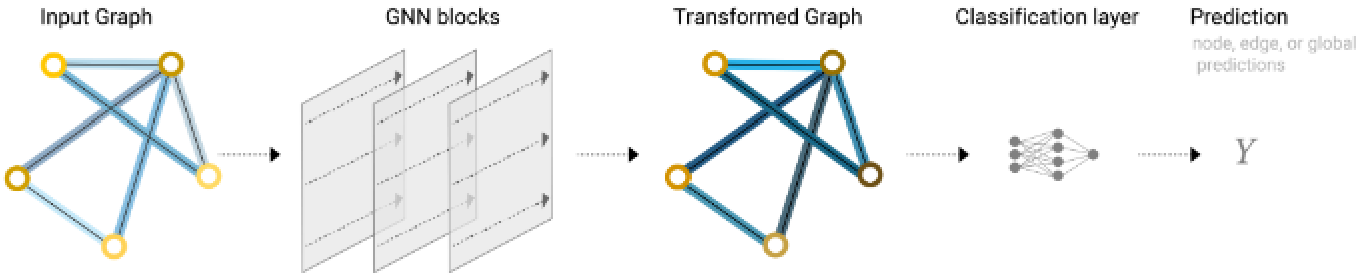


Figure 4. General process of classification using GNN [43].

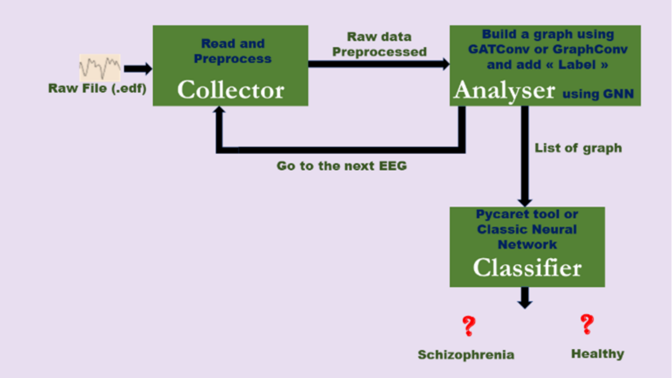


Figure 5. Proposed Method.

aggregate or center node, in the graph. This operation (aggregation) is performed at each hidden layer until the final graph is obtained Figure 6. Learning operation of GNN consists to use a loss function that encourages similar nodes to be close together and dissimilar nodes to be far apart. It means that the neural network is trained to generate low-dimensional vector representations (embeddings) for each node in a graph. During the GNN's training, parameters (weights and biases) are adjusted , the network aims to optimize its parameters to produce embeddings that capture meaningful information about the nodes in the graph. Then, GNN's parameters (weights and biases) are adjusted.

3) The classifier

GNNs can be used with many learning methods, including back-propagation, graph-based optimization methods, and Deep Learning. The selection of the GNN classification method depends on the task at hand. This module takes as input the list obtained in the previous module to perform the classification. In the context of this work, the classification was done using Pycaret on the one hand and a classical neural network on the other.

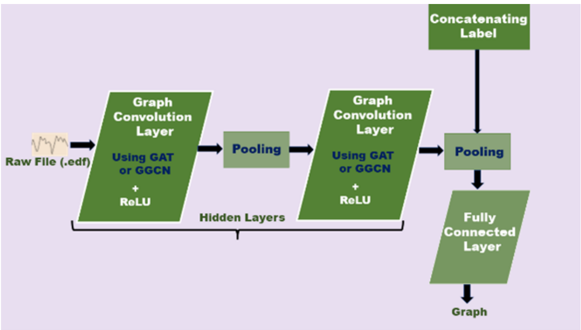


Figure 6. Presentation of analyser's module.

IV. RESULTS

The results obtained will be presented in three phases: the results of the classification obtained with the GCN method for the construction of the graph and pycaret for the classification, those obtained following the use of GAT and pycaret for the classification and finally the information received following the use of GAT to obtain the graph and a classical neural network for the classification. Before presenting these results, it is important to define some concepts such as: sensitivity, specificity, confusion matrix. Sensitivity can be defined as the rate of positive individuals correctly predicted by the model. To calculate

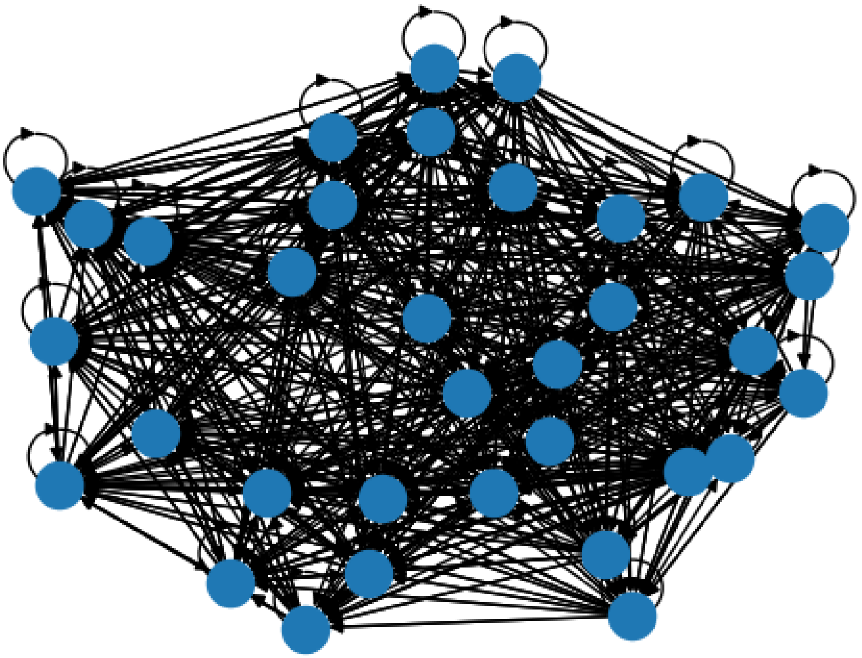
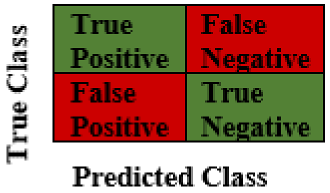


Figure 7. Example of the graph obtained after Skip-Gram Operation.



True Class	True Positive	False Negative
	False Positive	True Negative
	Predicted Class	

Figure 8. Explaining a confusion matrix.

it, the following formula is used: sensitivity=(True Positive (TP))/(True Positive (TP)+False Negative). Specificity is the rate of negative individuals correctly predicted by the model. It is defined as follows: specificity=(True Negative (TN))/(True Negative (TN)+False Positive). In the context of a medical diagnosis, sensitivity specifies the percentage of people declared ill by the model who are actually ill in reality. On the other hand, specificity indicates the rate of people predicted not to be ill who are actually not ill [47]. In view of the two definitions above, the best model is the one with a high sensitivity and specificity. As for the confusion matrix, it is the main metric in terms of classification, as it allows all other metrics to be obtained Figure 8. It is presented as follows:

The dataset used was presented above, which is twenty-eight (28) EEGs. During the implementation work, the test set considered was a third (1/3), or approximately nine (09) EEGs. These values will help to understand the confusion matrices that will be described below.

A. USE OF GCN

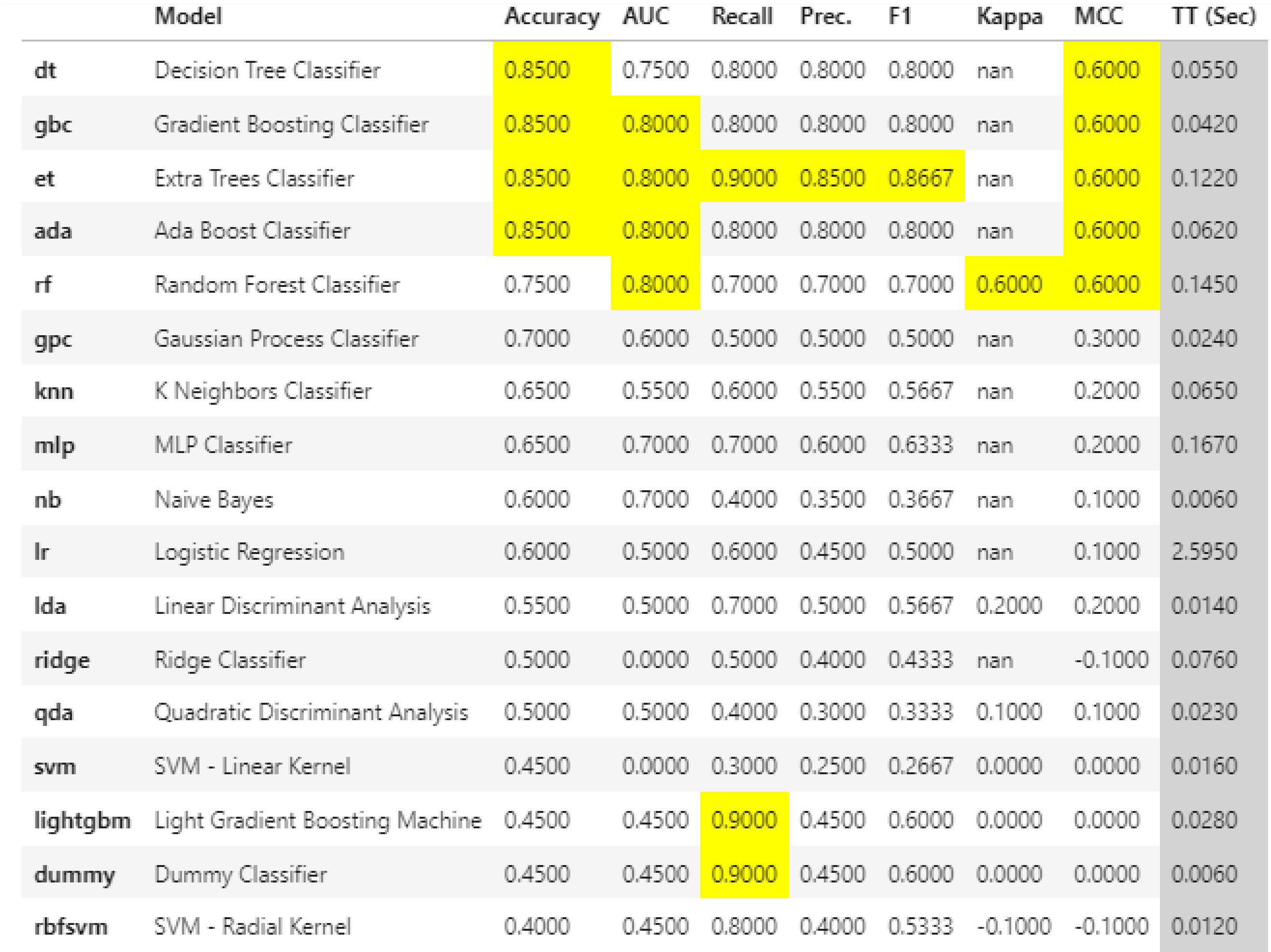
Figure 9 presents the values of the different metrics obtained with different classifiers in a decreasing order. The information presented in this figure shows that Decision Tree Classifier, Gradient Boosting Classifier, Extra Trees Classifier and Ada Boost Classifier are the four (04) best classifiers with an accuracy and recall of about 80%. That figure also shows that Light Gradient Boosting Machine and Dummy Classifier have a very good recall of 90%.

The ROC curve for the best decision tree classifier is shown in Figure 10. This curve shows a balance between the two classes with AUCs equal to 0.57 for all classes. This means that 57% of sick people are predicted as sick. Also, 57% of people who are not ill are reported as such when this model is used.

The confusion matrix shown in Figure 11. indicates that there are four (04) poorly predicted values against six (06) well-predicted values. Thus, according to this matrix, sensitivity=3/(3+3)= 50% and specificity=2/(2+1)= 66%. This means that for this model, although sensitivity is acceptable, specificity is higher. This model is not reassuring enough in that it is able to announce at 34% that an individual suffers from schizophrenia although he is not. Similarly, it could at 50% conclude that a person is healthy, but is.

B. USE OF GAT

The results obtained with GAT and the Pycaret classifier are presented in Figure 12. The four (04) best classifiers SVM - Radial Kernel, Linear Discriminant Analysis, Ada Boost Classifier, and Extra Trees Classifier have an average accuracy of about 64%; SVM - Radial Kernel being the best



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
dt	Decision Tree Classifier	0.8500	0.7500	0.8000	0.8000	0.8000	nan	0.6000	0.0550
gbc	Gradient Boosting Classifier	0.8500	0.8000	0.8000	0.8000	0.8000	nan	0.6000	0.0420
et	Extra Trees Classifier	0.8500	0.8000	0.9000	0.8500	0.8667	nan	0.6000	0.1220
ada	Ada Boost Classifier	0.8500	0.8000	0.8000	0.8000	0.8000	nan	0.6000	0.0620
rf	Random Forest Classifier	0.7500	0.8000	0.7000	0.7000	0.7000	0.6000	0.6000	0.1450
gpc	Gaussian Process Classifier	0.7000	0.6000	0.5000	0.5000	0.5000	nan	0.3000	0.0240
knn	K Neighbors Classifier	0.6500	0.5500	0.6000	0.5500	0.5667	nan	0.2000	0.0650
mlp	MLP Classifier	0.6500	0.7000	0.7000	0.6000	0.6333	nan	0.2000	0.1670
nb	Naive Bayes	0.6000	0.7000	0.4000	0.3500	0.3667	nan	0.1000	0.0060
lr	Logistic Regression	0.6000	0.5000	0.6000	0.4500	0.5000	nan	0.1000	2.5950
lda	Linear Discriminant Analysis	0.5500	0.5000	0.7000	0.5000	0.5667	0.2000	0.2000	0.0140
ridge	Ridge Classifier	0.5000	0.0000	0.5000	0.4000	0.4333	nan	-0.1000	0.0760
qda	Quadratic Discriminant Analysis	0.5000	0.5000	0.4000	0.3000	0.3333	0.1000	0.1000	0.0230
svm	SVM - Linear Kernel	0.4500	0.0000	0.3000	0.2500	0.2667	0.0000	0.0000	0.0160
lightgbm	Light Gradient Boosting Machine	0.4500	0.4500	0.9000	0.4500	0.6000	0.0000	0.0000	0.0280
dummy	Dummy Classifier	0.4500	0.4500	0.9000	0.4500	0.6000	0.0000	0.0000	0.0060
rbfsvm	SVM - Radial Kernel	0.4000	0.4500	0.8000	0.4000	0.5333	-0.1000	-0.1000	0.0120

Figure 9. Classification results using GCN coupled with Pycaret.

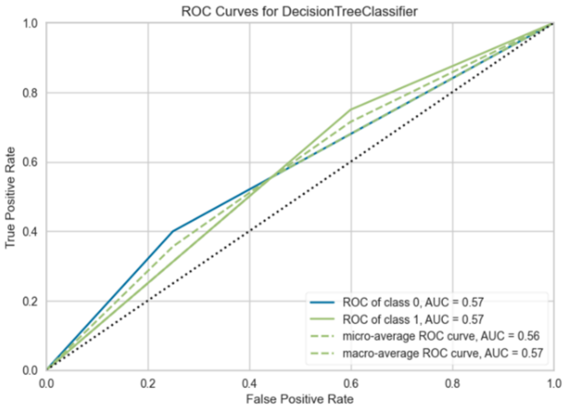


Figure 10. ROC of the best classification method Decision Tree Classifier for GCN.

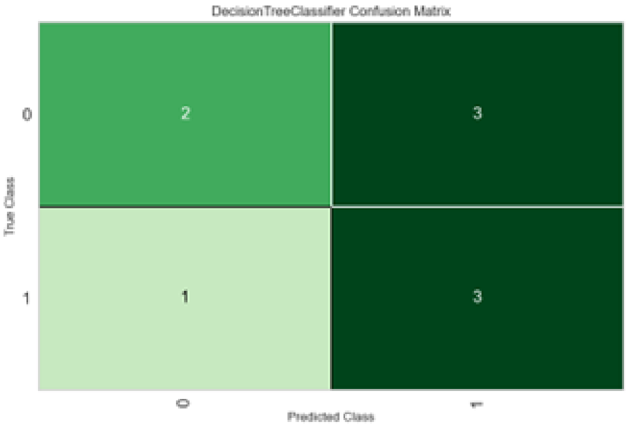


Figure 11. Confusion matrix of the best classification method Decision Tree Classifier for GCN.

classifier with an accuracy of 75% and a recall of 90%. The ROC curve in Figure 13 shows that SVM - Radial Kernel has a very high sensitivity and specificity of 95%. This means that when this model is used, 95% of sick people are predicted to be sick. And 95% of people who are not sick are reported as sick. This figure shows a very large area under the curve; this means that this classifier makes the least possible errors; about 5%.

As shown in Figure 14, the confusion matrix of the best classification model for GAT highlights the fact that the SVM - Radial Kernel Classifier model performs only two (02) of the nine (09) tests performed.

Indeed, sensitivity=3/(3+1)= 75% and specificity=4/(4+1)=80%. This makes GAT a very suitable model for predictions from this dataset.

C. USE OF GCN COMBINED WITH A NEURAL NETWORK

In this part, GCN is used to generate the graph and then a classical neural network is built to implement the classification. This neural network is a Convolution Neural Network (CNN) with two (02) hidden layers. The result obtained shows an accuracy of 80% after 120 epochs, which is slightly lower than that obtained with GCN and Pycaret. However, Figure 15. shows a drop in error with each iteration.

V. DISCUSSION

This part is devoted to the comparison of the three parts presented in terms of results. The results with GCN, GAT, and GCN coupled to CNN for classification. This comparison will be made between GCN and GAT on the one hand and between GCN and GCN coupled with CNN on the other hand. The parameters of comparison are the time and resources used for the generation of the list of graphs, of precision, of AUC on the one hand, and the learning time and the time and resources for the classification, of precision on the other hand. It is important to note that the dataset size is the same for all these methods: twenty-eight (28) EEGs; fourteen (14) from sick people and fourteen (14) from healthy people.

A. COMPARISON OF GCN AND GAT

Some authors such as Lee et al. [37] consider GCN to be the most widely used GNN method. In this section, we will compare GCN and GAT from the results obtained on our dataset. It is important to remember that the classification time is the same for these two (02) methods. Table 1 provides a summary of the values of these comparators.

This table shows that GCN has a very low graph generation time, and good accuracy but a rather high error rate (43%). On the other hand, GAT has a fairly high graph generation time, and good accuracy, but a very low error rate (5%).

B. GCN COUPLED WITH CNN

In this part, GCN is used to generate the dataset that will be the input of the classification module. The aim is to compare the CNN and pycaret classifiers.

As show in Table 2, the results of some comparison metrics. Through these values, it appears that GCN and pycaret as a classifier use significantly fewer resources than GCN coupled with CNN. Also, simple GCN gives an accuracy close to GCN coupled with CNN. However, according to the results presented above, the error rate obtained when classifying with CNN is very decreasing compared to the error rate obtained with GCN which is quite high.

Table 1. Comparison GCN vs GAT

GNN Method	**Graph generation time**	**Accuracy of the best classifier**	**AUC of the best classifier**
GCN	45 seconds with an average of 1.62 seconds/iteration	85%	57 % Schizophrenia 57 % Healthy
GAT	52 minutes and 06 seconds with an average of 111.65 seconds/iteration	80%	95 % Schizophrenia 95 % Healthy

Table 2. Comparison GCN vs GCN coupled with CNN

Classifier	**Classification time**	**Accuracy of the classifier**
Pycaret	32 seconds	85%
CNN	45 minutes and 24 seconds with an average of 111.65 seconds/iteration	80%

C. ABLATION STUDY FROM CNN TO GAT VIA GCN

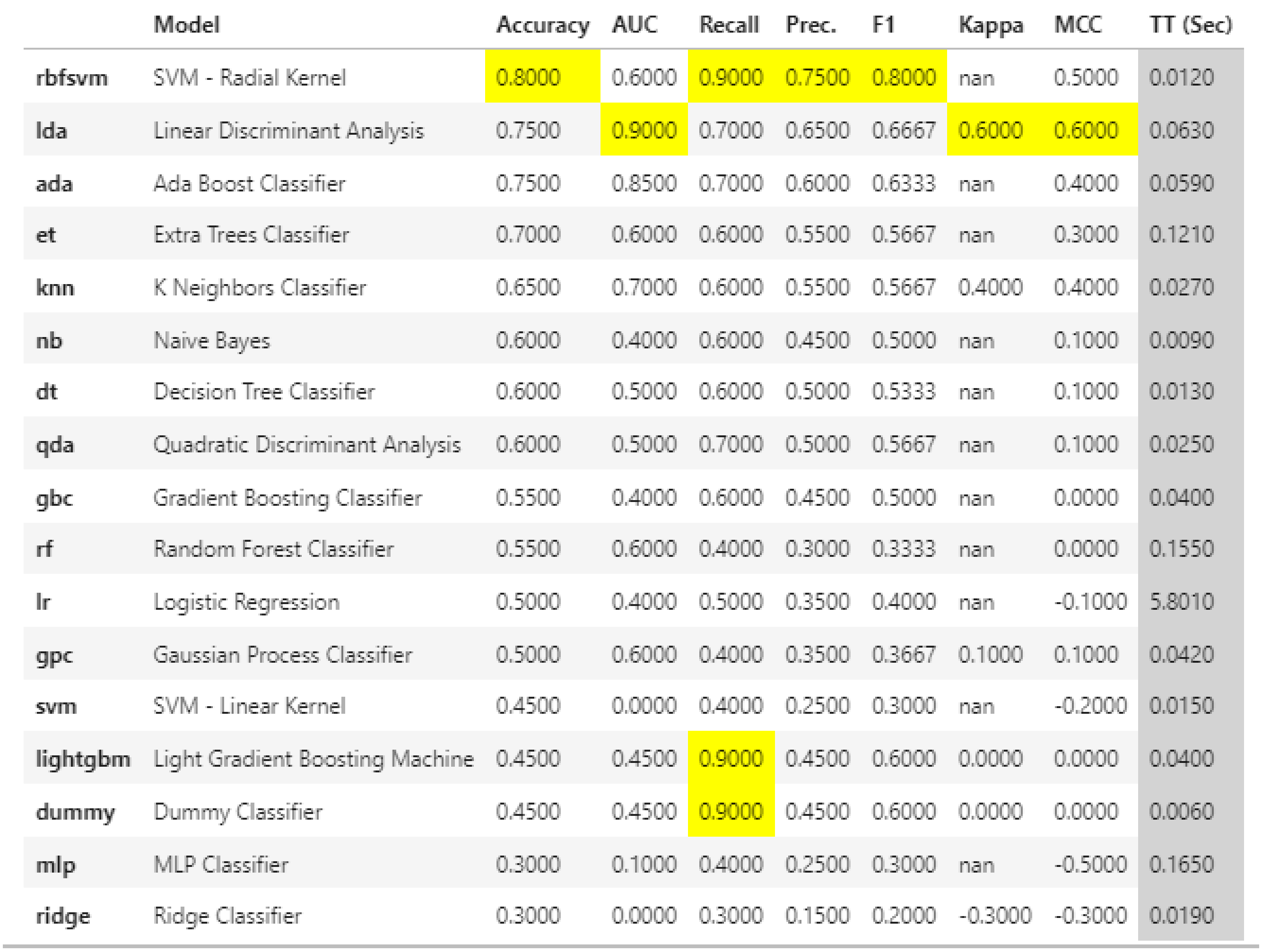
This study will no longer take into account classification time, accuracy or graph generation time, as these have already been done in the sections above. It will take the form of two tables. The first 3 compares basic CNN, GCN and GAT in terms of architecture or modification, purpose, expected outcome, training time, inference time and peak VRAM. Some values in the table are "‘Not defined"’ because CNN was just used for classification in the case of this study.

The second 4 presents the comparisons of expected conclusions and obtained conclusion.

In sum, the elements presented above allow us to conclude that GAT, despite its very high resource consumption, provides a better performance than GCN and GCN coupled with CNN, as it produces a very low error rate. This is the case with the dataset used which is twenty-eight (28) EEGs. It remains to be seen whether GAT will maintain the same performance with a larger dataset.

VI. CONCLUSION AND FUTURE WORKS

In this paper, the GNN was used through two of its methods, GCN and GAT, to classify schizophrenia EEGs and to



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rbfsvm	SVM - Radial Kernel	0.8000	0.6000	0.9000	0.7500	0.8000	nan	0.5000	0.0120
lda	Linear Discriminant Analysis	0.7500	0.9000	0.7000	0.6500	0.6667	0.6000	0.6000	0.0630
ada	Ada Boost Classifier	0.7500	0.8500	0.7000	0.6000	0.6333	nan	0.4000	0.0590
et	Extra Trees Classifier	0.7000	0.6000	0.6000	0.5500	0.5667	nan	0.3000	0.1210
knn	K Neighbors Classifier	0.6500	0.7000	0.6000	0.5500	0.5667	0.4000	0.4000	0.0270
nb	Naive Bayes	0.6000	0.4000	0.6000	0.4500	0.5000	nan	0.1000	0.0090
dt	Decision Tree Classifier	0.6000	0.5000	0.6000	0.5000	0.5333	nan	0.1000	0.0130
qda	Quadratic Discriminant Analysis	0.6000	0.5000	0.7000	0.5000	0.5667	nan	0.1000	0.0250
gbc	Gradient Boosting Classifier	0.5500	0.4000	0.6000	0.4500	0.5000	nan	0.0000	0.0400
rf	Random Forest Classifier	0.5500	0.6000	0.4000	0.3000	0.3333	nan	0.0000	0.1550
lr	Logistic Regression	0.5000	0.4000	0.5000	0.3500	0.4000	nan	-0.1000	5.8010
gpc	Gaussian Process Classifier	0.5000	0.6000	0.4000	0.3500	0.3667	0.1000	0.1000	0.0420
svm	SVM - Linear Kernel	0.4500	0.0000	0.4000	0.2500	0.3000	nan	-0.2000	0.0150
lightgbm	Light Gradient Boosting Machine	0.4500	0.4500	0.9000	0.4500	0.6000	0.0000	0.0000	0.0400
dummy	Dummy Classifier	0.4500	0.4500	0.9000	0.4500	0.6000	0.0000	0.0000	0.0060
mlp	MLP Classifier	0.3000	0.1000	0.4000	0.2500	0.3000	nan	-0.5000	0.1650
ridge	Ridge Classifier	0.3000	0.0000	0.3000	0.1500	0.2000	-0.3000	-0.3000	0.0190

Figure 12. Classification results using GAT and Pycaret.

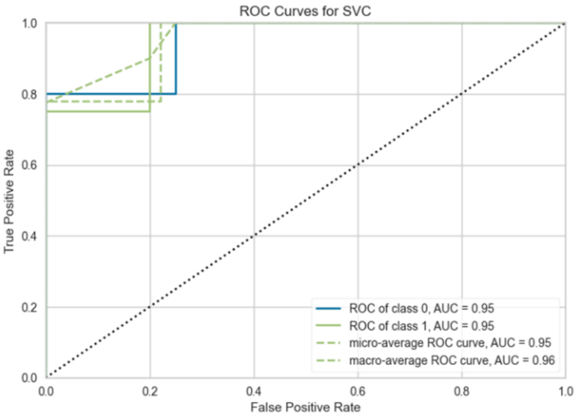


Figure 13. ROC of the best classification method SVM - Radial Kernel Classifier for GAT.

compare these variants based on the results obtained to choose the one that best suits the resolution of the project in which this work is involved. This project aims to develop artificial intelligence for the early diagnosis of schizophrenia by EEG. The results show that GAT offers better sensitivity and specificity than GCN and GCN coupled with CNN, even though it is very resource-intensive and has a slightly lower accuracy than the others. The sensitivity reaches 75% and the specificity is close to 80% which is very representative of reality. This means that the results obtained with GAT allow us to say that 75% of an individual is ill, even though he or she is indeed ill, and 80% that an individual is not suffering from schizophrenia, even though he or she is not suffering. The next step is to increase the dataset so that we can say that GAT is undoubtedly the right method for the classification of schizophrenia EEGs. It is important to note, however, that an increase in dataset size equates to an increase in computational resources and perhaps a better learning result with Pycaret, which performs much better on a large dataset. Also, since GAT offers better results in terms of accuracy, sensitivity and specificity, but is very hardware intensive compared to GCN for example. A

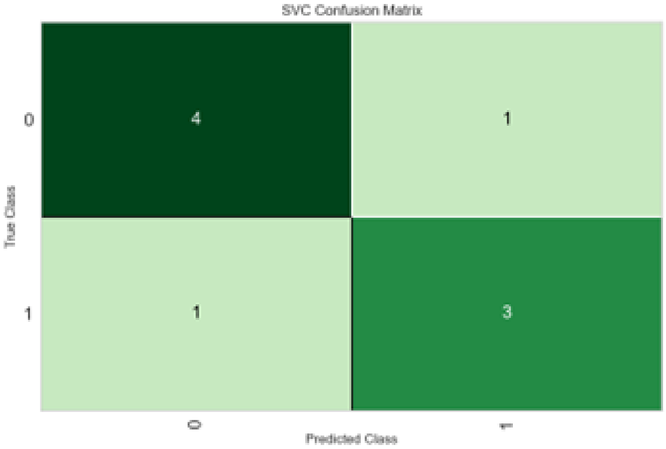


Figure 14. Confusion matrix of the best classification method SVM - Radial Kernel Classifier for GAT.

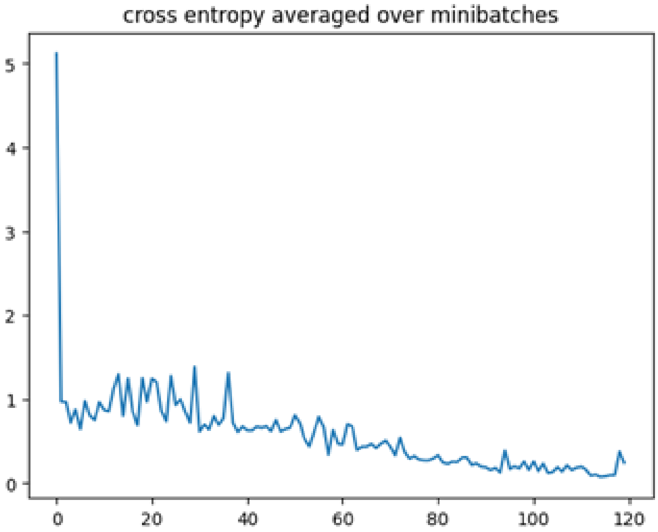


Figure 15. Confusion matrix of the best classification method SVM - Radial Kernel Classifier for GAT.

possible improvement would be to use a neural network to improve the Skip-Gram step during the construction of the graph with GCN before carrying out the classification using Pycaret among others. This would perhaps make it possible to gain in resources. It would also be useful in future work to take into account the actual clinical diagnostic steps in the learning process so that these results can be used in hospitals. That is to say, to first include data related to Physical exams (weight gain, blood pressure, temperature, age, etc.) and biological examinations (emoparasite, stool, tertiary phase syphilis, etc.) before arriving at Neurological exams such as EEG as discussed throughout this paper. Future work could also explore improved GNN models such as the Adaptive Feature and Topology Graph Convolutional Neural Network (AAGCN) model proposed by [48], which incorporates an adaptive layer whose main advantage is that it efficiently extracts hidden features and topological information, thereby improving the expressive power and classification performance of these networks. Another model is three-dimensional adaptive graph convolutional neural network (3D-AGCN) [49] that provides a new auxiliary diagnostic method for schizophrenia. Other [50] combine CNN and GNN for early detection of schizophrenia using EEG data. These latest discoveries are avenues to be explored in our future work.

References

[1] I'MTech, "Ces technologies qui décryptent le fonctionnement du cerveau," Feb. 2019. [Online]. Available: https://imtech.imt.fr/2019/02/04/technologies-decrypter-cerveau/

[2] "Introduction to Non-Euclidean Geometry - EscherMath." [Online]. Available: https://mathstat.slu.edu/escher/index.php/Introduction_to_Non-Euclidean_Geometry

[3] "Maths - Non-Euclidean Spaces - Martin Baker." [Online]. Available: https://www.euclideanspace.com/maths/geometry/space/nonEuclid/index.htm

Table 3. Comparison basic CNN, GCN and GAT

Neural Network Method	Architecture or Modification	Purpose	Expected Outcome	Training Time	Inference Time	Peak VRAM
CNN	3-4 convolutional layers with ReLU, BatchNorm, and max-pooling. And Fully connected head for classification or regression.	Start by extracting spatial features using methods that don't rely on graph structures.	Good performance on grid-structured data as images, but may struggle with non-Euclidean or relational data	Not defined	Not defined	Not defined
GCN	Replace CNN layers with GCN layers, aggregates features from neighbors using mean pooling and no attention mechanism; fixed weight sharing.	Test the impact of graph structure vs. grid structure.	Improved performance on graph data, but may lack adaptability due to fixed aggregation.	45 seconds	01 minutes and 17 seconds	1.4 GB
GAT	Replace GCN layers with GAT layers, uses attention coefficients to weigh neighbor importance and multi-head attention for stability.	Evaluate dynamic attention vs. fixed GCN aggregation.	Better performance than GCN due to adaptive neighbor weighting, but higher computational cost.	52 minutes and 06 seconds	52 minutes and 38 seconds	2 GB

Table 4. Comparison of expected conclusions and obtained conclusion for CNN, GCN and GAT

Comparator	Expected Conclusion	Obtained Conclusion
CNN vs GCN/GAT	GCNs and GATs are expected to excel on graph-structured data, but their performance might lag behind CNNs on traditional grid-like data.	Not defined
GCN vs GAT	When a graph's neighbor relationships vary, GATs should prove more effective than GCNs	Relationships vary over time during GNN training and GAT is more effective than GCN
Lightweight GAT	For many scenarios, linearized attention or sampling might offer the optimal balance of performance and efficiency. However, if the graph is simple, a single-head attention mechanism could be entirely sufficient.	Single-head attention has been used because of simplicity of graph (nodes don't have many characteristics).

[4] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson, "A Survey on Graph-Based Deep Learning for Computational Histopathology," 2021, publisher: arXiv Version Number: 2. [Online]. Available: https://arxiv.org/abs/2107.00272

[5] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson, "Graph-Based Deep Learning for Medical Diagnosis and Analysis: Past, Present and Future," Sensors, vol. 21, no. 14, p. 4758, Jul. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/14/4758

[6] A. Demir, T. Koike-Akino, Y. Wang, and D. Erdogmuş, "EEG-GAT: Graph Attention Networks for Classification of Electroencephalogram (EEG) Signals," 2022, publisher: Unpublished. [Online]. Available: https://rgdoi.net/10.13140/RG.2.2.19854.23365

[7] M. Kule and M. M. Kaggwa, "Adherence to Typical Antipsychotics among Patients with Schizophrenia in Uganda: A Cross-Sectional Study," Schizophrenia Research and Treatment, vol. 2023, pp. 1–12, Feb. 2023. [Online]. Available: https://www.hindawi.com/journals/schizort/2023/7035893/

[8] C. Nauck, M. Lindner, K. Schürholt, and F. Hellmann, "Towards Dynamic Stability Assessment of Power Grid Topologies using Graph Neural Networks," 2022, publisher: arXiv Version Number: 3. [Online]. Available: https://arxiv.org/abs/2206.06369

[9] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," AI Open, vol. 1, pp. 57–81, 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2666651021000012

[10] B. Mahmud, S. Mahi, M. Shuvo, S. Islam, and M. K. Morol, A comparative analysis of Graph Neural Networks and commonly used machine learning algorithms on fake news detection, Mar. 2022.

[11] Y.-C. Huang, "Condition assessment of power transformers using genetic-based neural networks," IEE Proceedings - Science, Measurement and Technology, vol. 150, no. 1, p. 19, 2003. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/ip-smt_20020638

[12] Z. Chen, P. P. Kulkarni, I. R. Galatzer-Levy, B. Bigio, C. Nasca, and Y. Zhang, "Modern Views of Machine Learning for Precision Psychiatry," preprint, Apr. 2022. [Online]. Available: https://www.techrxiv.org/articles/preprint/Modern_Views_of_Machine_Learning_for_Precision_Psychiatry/19502131/1

[13] T. Pélagie Flore, J. Kuate Fotso, B. Isaoura, and P. Ele, "Intelligent system for early diagnosis of paediatric cerebral malaria from an electroencephalogram," Current Medicine Research and Practice, vol. 10, no. 5, p. 210, 2020. [Online]. Available: http://www.cmrpjournal.org/text.asp?2020/10/5/210/299514

[14] P. F. Temgoua Nanfack and A. M. M. Junior, "Study of Metrics that Could be Considered as Inputs to an Intelligent System for Diagnosing Schizophrenia based on an Electroencephalogram (EEG)," European Journal of Engineering and Technology Research, vol. 5, no. 3, pp. 292–296, Mar. 2020. [Online]. Available: https://www.ej-eng.org/index.php/ejeng/article/view/1811

[15] S. L. Oh, J. Vicnesh, E. J. Ciaccio, R. Yuvaraj, and U. R. Acharya, "Deep Convolutional Neural Network Model for Automated

Diagnosis of Schizophrenia Using EEG Signals," Applied Sciences, vol. 9, no. 14, p. 2870, Jul. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/14/2870

[16] A. Shalbaf, S. Bagherzadeh, and A. Maghsoudi, "Transfer learning with deep convolutional neural network for automated detection of schizophrenia from EEG signals," Physical and Engineering Sciences in Medicine, vol. 43, no. 4, pp. 1229–1239, Dec. 2020. [Online]. Available: https://link.springer.com/10.1007/s13246-020-00925-9

[17] C.-R. Phang, C.-M. Ting, S. B. Samdin, and H. Ombao, "Classification of EEG-based Effective Brain Connectivity in Schizophrenia using Deep Neural Networks," in 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER). San Francisco, CA, USA: IEEE, Mar. 2019, pp. 401–406. [Online]. Available: https://ieeexplore.ieee.org/document/8717087/

[18] C.-R. Phang, F. Noman, H. Hussain, C.-M. Ting, and H. Ombao, "A Multi-Domain Connectome Convolutional Neural Network for Identifying Schizophrenia From EEG Connectivity Patterns," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 5, pp. 1333–1343, May 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8836535/

[19] K. Singh, S. Singh, and J. Malhotra, "Spectral features based convolutional neural network for accurate and prompt identification of schizophrenic patients," Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, vol. 235, no. 2, pp. 167–184, Feb. 2021. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0954411920966937

[20] M. Hussain, N. A. Alsalooli, N. Almaghrabi, and E.-u.-H. Qazi, "Schizophrenia Detection on EEG Signals Using an Ensemble of a Lightweight Convolutional Neural Network," Applied Sciences, vol. 14, no. 12, p. 5048, Jun. 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/12/5048

[21] J. Rahul, D. Sharma, L. D. Sharma, U. Nanda, and A. K. Sarkar, "A systematic review of EEG based automated schizophrenia classification through machine learning and deep learning," Frontiers in Human Neuroscience, vol. 18, p. 1347082, Feb. 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnhum.2024.1347082/full

[22] "How AWS uses graph neural networks to meet customer needs," Mar. 2022. [Online]. Available: https://www.amazon.science/blog/how-aws-uses-graph-neural-networks-to-meet-customer-needs

[23] A. Demir, T. Koike-Akino, Y. Wang, M. Haruna, and D. Erdogmus, EEG-GNN: Graph Neural Networks for Classification of Electroencephalogram (EEG) Signals, Jun. 2021.

[24] D. Klepl, F. He, M. Wu, D. J. Blackburn, and P. G. Sarrigiannis, "EEG-based Graph Neural Network Classification of Alzheimer's Disease: An Empirical Evaluation of Functional Connectivity Methods," Neuroscience, preprint, Jun. 2022. [Online]. Available: http://biorxiv.org/lookup/doi/10.1101/2022.06.14.496080

[25] A. Rahmani, A. Venkitaraman, and P. Frossard, "A Meta-GNN approach to personalized seizure detection and classification," 2022, publisher: arXiv Version Number: 1. [Online]. Available: https://arxiv.org/abs/2211.02642

[26] X. Lun, S. Jia, Y. Hou, Y. Shi, Y. Li, H. Yang, S. Zhang, and J. Lv, GCNs-Net: A Graph Convolutional Neural Network Approach for Decoding Time-resolved EEG Motor Imagery Signals, Sep. 2022, vol. PP, journal Abbreviation: IEEE Transactions on Neural Networks and Learning Systems Publication Title: IEEE Transactions on Neural Networks and Learning Systems.

[27] N. Wagh and Y. Varatharajah, EEG-GCNN: Augmenting Electroencephalogram-based Neurological Disease Diagnosis using a Domain-guided Graph Convolutional Neural Network, Nov. 2020.

[28] D. Lei, K. Qin, W. H. L. Pinaya, J. Young, T. Van Amelsvoort, M. Marcelis, G. Donohoe, D. O. Mothersill, A. Corvin, S. Vieira, S. Lui, C. Scarpazza, C. Arango, E. Bullmore, Q. Gong, P. McGuire, and A. Mechelli, "Graph Convolutional Networks Reveal Network-Level Functional Dysconnectivity in Schizophrenia," Schizophrenia Bulletin, vol. 48, no. 4, pp. 881–892, Jun. 2022. [Online]. Available: https://academic.oup.com/schizophreniabulletin/article/48/4/881/6586020

[29] S. Ghosh, E. Bhargava, C.-T. Lin, and S. S. Nagarajan, "Graph Convolutional Learning of Multimodal Brain Connectome Data for Schizophrenia Classification," Bioengineering, preprint, Jan. 2023. [Online]. Available: http://biorxiv.org/lookup/doi/10.1101/2023.01.05.522960

[30] D. Klepl, M. Wu, and F. He, "Graph Neural Network-based EEG Classification: A Survey," 2023, publisher: [object Object] Version Number: 2. [Online]. Available: https://arxiv.org/abs/2310.02152

[31] E. Olejarczyk and W. Jernajczyk, "Graph-based analysis of brain connectivity in schizophrenia," PLOS ONE, vol. 12, no. 11, p. e0188629, Nov. 2017. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0188629

[32] E. Olejarczyk and W. Jernajczyk, "EEG in schizophrenia," Sep. 2017, publisher: RepOD. [Online]. Available: https://repod.icm.edu.pl/dataset.xhtml?persistentId=

[33] MNE, "MNE — MNE 1.1.1 documentation," Sep. 2022. [Online]. Available: https://mne.tools/stable/index.html

[34] PyCaret, "Home - PyCaret," 2022. [Online]. Available: https://pycaret.org/

[35] A. Munjal, R. Khandia, and B. Gautam, "A MACHINE LEARNING APPROACH FOR SELECTION OF POLYCYSTIC OVARIAN SYNDROME (PCOS) ATTRIBUTES AND COMPARING DIFFERENT CLASSIFIER PERFORMANCE WITH THE HELP OF WEKAAND PYCARET," INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH, pp. 59–63, Dec. 2020. [Online]. Available: https://www.worldwidejournals.com/international-journal-of-scientific-research-(IJSR)/recent_issues_pdf/2020/December/a-machine-learning-approach-for-selection-of-polycystic-ovarian-sy December_2020_6430160911_5416514.pdf

[36] "NN Modules (Tensorflow) — DGL 0.4.3post2 documentation." [Online]. Available: https://docs.dgl.ai/en/0.4.x/api/python/nn.tensorflow.html#graphconv

[37] J. Lee, J. W. Choi, and S. Jo, "Subject-Independent Motor Imagery EEG Classification Based on Graph Convolutional Network," in Pattern Recognition, C. Wallraven, Q. Liu, and H. Nagahara, Eds. Cham: Springer International Publishing, 2022, vol. 13189, pp. 268–281, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-031-02444-3_20

[38] S. Wein, A. Schüller, A. M. Tomé, W. M. Malloni, M. W. Greenlee, and E. W. Lang, "Forecasting brain activity based on models of spatiotemporal brain dynamics: A comparison of graph neural network architectures," Network Neuroscience, vol. 6, no. 3, pp. 665–701, Jul. 2022. [Online]. Available: https://direct.mit.edu/netn/article/6/3/665/111069/Forecasting-brain-activity-based-on-models-of

[39] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," 2016, publisher: arXiv Version Number: 4. [Online]. Available: https://arxiv.org/abs/1609.02907

[40] "Math Behind Graph Neural Networks," Mar. 2022. [Online]. Available: https://rish-16.github.io/posts/gnn-math/

[41] J. Lu, Y. Tian, S. Wang, M. Sheng, and X. Zheng, "PearNet: A Pearson Correlation-based Graph Attention Network for Sleep Stage Recognition," Oct. 2022, arXiv:2209.13645 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2209.13645

[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," Feb. 2018, arXiv:1710.10903 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1710.10903

[43] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. Wiltschko, "A Gentle Introduction to Graph Neural Networks," Distill, vol. 6, no. 8, p. 10.23915/distill.00033, Aug. 2021. [Online]. Available: https://distill.pub/2021/gnn-intro

[44] Z. Xiong, Q. Shen, Y. Xiong, YijieWang, and W. Li, "New Generation Model of Word Vector Representation Based on CBOW or Skip-Gram," Computers, Materials & Continua, vol. 60, no. 1, pp. 259–273, 2019. [Online]. Available: http://www.techscience.com/cmc/v60n1/28351

[45] A. Celikkanat and F. D. Malliaros, "Exponential Family Graph Embeddings," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 3357–3364, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5737

[46] P. Compagnon and K. Olliver, "Graph embeddings for social network analysis, state of the art," 2016, publisher: Master's thesis, INSA Lyon. [Online]. Available: https://www.researchgate.net/profile/Paul-Compagnon/publication/331714802_Graph_Embeddings_for_Social_Network_Analysis_State_of_the_Art/links/5c890149a6fdcc381750fc1a/Graph-Embeddings-for-Social-Network-Analysis-State-of-the-Art.pdf

[47] C. Tremblay, "Sensitivity, Specificity & ROC curve / Sensibilité, Spécificité et courbe ROC," Nov. 2021. [Online]. Available: https://kobia.fr/classification-metrics-sensitivity-specificity-roc/

[48] B. Wang, B. Cai, J. Sheng, and W. Jiao, "AAGCN: a graph convolutional neural network with adaptive feature and topology learning," Scientific Reports, vol. 14, no. 1, p. 10134, May 2024. [Online]. Available: https://www.nature.com/articles/s41598-024-60598-2

[49] G. Community, "Implement your own word2vec(skip-gram) model in Python," 2024, section: Python.

[Online]. Available: https://www.geeksforgeeks.org/implement-your-own-word2vecskip-gram-model-in-python/

[50] L. Reetha and R. Gnanajeyaraman, "A hybrid CNN-GNN approach for enhanced early detection of schizophrenia using EEG data," in 2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC). Silchar, India: IEEE, Feb. 2025, pp. 810–816. [Online]. Available: https://ieeexplore.ieee.org/document/10969251/

PELAGIE FLORE TEMGOUA NANFACK is currently a researcher at the Ministry of Scientific Research and Innovation and PhD student in the Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea in Cameroon. I mainly work in computational neuroscience even if as a curious person, I have other interests such as applications programming, the Internet of Things and energy data processing.

ELIE TAGNE FUTE is an Associate Professor, Lecturer/Researcher in the Department of Computer Engineering (Faculty of Engineering and Technology, University of Buea) and the Department of Mathematics and Computer Science (Faculty of Science, University of Dschang), Cameroon. He holds a Ph.D. in Computer Science of the University of Technology of Belfort-Montbeliard and the University of Yaounde I. He is currently the Head of Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea, Cameroon. His major research works are focused on multi-agent patrolling, sensor networks, multi-sensor patrolling, Data Security Approaches for Distributed Systems, Cybersecurity, Information Systems Security, Machine learning & AI.