EEG classification of ADHD patients with Graph Neural Network

Artificial Intelligence (CSE 4633) Course Project Final Report

Yongjun Lee
Department of Mathematics and Statistics
Mississippi State University
Starkville, United States
yl646@msstate.edu

***Abstract*—Attention Deficit Hyperactivity Disorder (ADHD) is one of the most common mental disorders in the United States. In recent years, a growing number of researchers have been applying Deep Learning (DL) algorithms, specifically Convolutional Neural Networks (CNN), to classify ADHD patients from brain imaging data. While this approach yields high accuracy, it fails to provide insight into the brain connectivity of ADHD patients for clinical purposes. Previous work attempted to solve this issue by constructing a correlation network from EEG channels and training a CNN model with this data. In this research, we will construct a similar network using mutual information between each channel. However, a Graph Neural Network (GNN) model will be used instead to better accommodate the nature of constructed network data structure. We expect our model to outperform the previous CNN model while allowing us to understand the brain network connectivity of ADHD patients. Minimal preprocessing steps will be taken for network construction and training due to the nature of DL models. Further research can investigate how the model performance varies with different adjacency matrix construction or convolution methods.**

***Index Terms*—EEG, ADHD, Mutual Information, Graph Neural Network, Classification**

I. Introduction

Attention Deficit/Hyperactivity Disorder (ADHD) is one of the most prominent mental illness in the United States. An estimated 6.1 million (9.4%) of children 2–17 years of age, is said to have ever received an ADHD diagnosis [1]. Electroencephalogram (EEG) has been used for diagnosis and causality research for ADHD. EEG is a electrical brain activity recording technique that is more than a century old. Due to advancement in deep learning (DL) techniques, many models were proposed for feature extraction, and classification. Especially, convolutional neural networks (CNN) have been one of the most popular model for EEG research purposes. CNN has proven its performance in classification tasks of image-like data. However, use of CNN in EEG has its limitations. CNN does not provide useful interpretable results. Machine learning applications in EEG research is still facing a "black box" problem. Because of this reason, EEG is not considered as a single reliable source for ADHD patient diagnosis in the neuroscience community [3].

On the other hand, graph neural networks (GNN) has been gaining popularity since 2019. It is a deep learning model that can train on graph dataset and perform node classification, graph classification, and graph clustering tasks. It uses aggregation function to find node embeddings and/or graph embeddings. Function parameters are learned from feature vectors of the nodes in a graph. Instead color mapping EEG singals to image-like data to train CNN, each EEG electrode can be viewed as a node and their relationship can be embedded in a graph structure. We believe that this is a better representation of EEG signals, therefore outperforming CNN. We also expect GNN to provide insights to the connectivity of ADHD brain with the use of graph features. Also, by varying model structures, we expect to learn the intricacies of ADHD brain connectivity.

II. Related Work

A. Deep Learning in EEG Research

Roy et al. "review[ed] 154 papers that apply DL to EEG, published between January 2010 and July 2018, and spanning different application domains such as epilepsy, sleep, brain-computer interfacing, and cognitive and affective monitoring". According to the review, DL models using raw EEG data instead of hand-crafted features has achieved promising results. However, preprocessing and artifact removal is still considered as a crucial step. The median of subjects was 13, and median total recording time in minutes was 360, which tells us our number of recordings (n=121) and recording time (260 minutes) is sufficiently large. Out of all the reviewed papers, 40.3% used convolutional neural networks which is three times more compared to the second most used model, recurrent neural network (13%) [2].

B. Limits of Current EEG research

Adamou et al. systematically reviewed 21 papers that used EEG specifically for ADHD research. They point out the shortcomings of EEG for ADHD diagnosis and causality analysis. "The describing of raw EEG data without a theory driven study and standardized protocol, is problematic. It is possible that this approach is inhibiting the development of useful information regarding this potentially valuable method to aid diagnosis" [3]. They conclude that EEG is not yet a reliable form of ADHD diagnosis because of the apparent inconsistencies. Another limitation in ADHD research is that most datasets are not publicly available making it difficult for acquiring large dataset that includes various gender and age groups. Datasets from 42% of studies in the previous review was not publicly available [2].

C. Previous Method: Mutual Information with CNN

In an attempt to solve the "black box" problem, Chen et al. proposed a new method that utilizes mutual information (MI) adjacency matrices for their image construction. They stacked MI matrices to create a three dimensional image-like data. This was used to train their CNN models. They claim that using MI and hand-crafted features allows them to have better interpretability. Their model reached 94.67% accuracy. We wanted to compare their model to our GNN models. Dataset used in this research is different from the one used in ours. For comparable results, their data preprocessing steps and CNN construction was replicated as closely as possible [4].

III. Proposed Method

This paper consists of three main parts. 1) We preprocessed our data using EPOS pipeline and obtained a node adjacency table by calculating MI for every pair of channels. 2) For graph construction, instead of using the MI values as weights, we used them as feature vectors. GNN models were created with varying k-hop values. We used Graph Convolutional Network (GCN), GraphSAGE (SAGE), and Diffusion Convolutional Recurrent Neural Network (DCRNN) [6-7]. 3) To recreate the results in the previous paper, the mutual information table was reorganized and turned into image-like 3 dimensional data in a similar fashion. Previous research proposes four CNN models. We selected their best performing model for comparison.

IV. Experiments

A. Data Collection

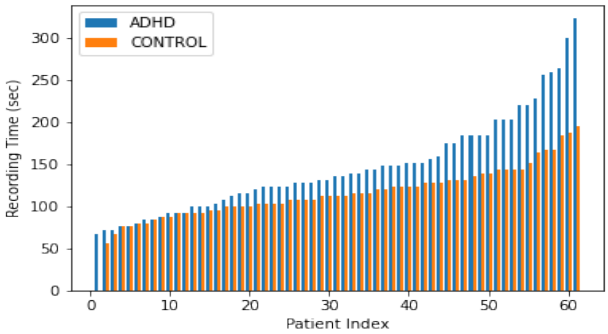
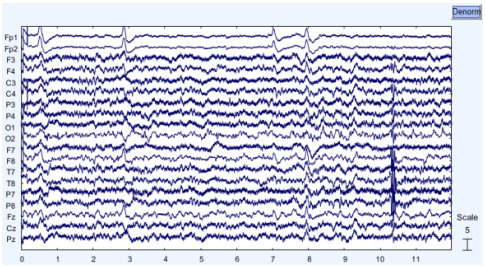
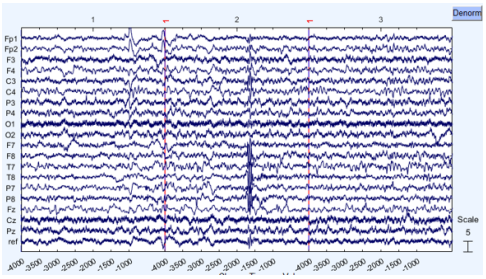


Fig. 1: Recording time comparison.

Due to the scarcity of publicly available EEG recordings of ADHD patients, we had limited options for acquiring the needed dataset. Our dataset was acquired from an open-source IEEE dataport [5]. Our data consists of EEG recordings of 61 children from the ADHDfrom ADHD group and 60 children from the control group (boys and girls,



(a) Raw EEG signal



(b) Preprocessed EEG signal

Fig. 2: Preprocessed signal has been epoched and outlier segments have been rejected.

ages 7-12). The recordings were made with a standard 10-20 EEG setup with 19 channels at 128 Hz. Children were asked to identify and count the number of characters on the computer screen. Recording was stopped once all of the given tasks were completed. The minimum recording time was 57 seconds whereas maximum recording time was 381 seconds. It can be observed that ADHD group generally took longer time to finish the tasks.

B. Preprocessing

EPOS is a "EEG (pre-)processing pipeline to achieve an automated method based on the semi-automated analysis proposed by Delorme and Makeig" [6]. EPOS was designed in an attempt to standardize EEG preprocessing procedures. This pipeline outlines 17 feature extraction steps for analysis. However, only steps one through five were applied since our model does not require extracted features. Detailed description of each steps are as follows:

- Average reference was calculated for detection of bad channels.
- Z-value statistical testing was performed to detect and reject bad channels.
- Rejected channel value was interpolated based on the neighboring channel values.
- All recordings were epoched with 4 seconds window with no overlap. This window size was determined from previous work.
- High pass filtering (1 Hz) was performed to dampen the signal for better interpretability.
- Independent Component Analysis was finally performed to detect and reject bad segments detection.

C. MI Adjacency Matrix Construction

$$
\begin{aligned}
I(X;Y) &= H(X) - H(X \mid Y) \\
&= H(Y) - H(Y \mid X) \\
&= H(X,Y) - H(X \mid Y) - H(Y \mid X) \\
&= \iint_{x,y} P_{XY}(x,y) log\left[P_X Y(x,y) - P_X(x) P_Y(y)\right] \\
Cov(X;Y) &= \iint_{x,y} xy\left[P_X Y(x,y) - P_X(x) P_Y(y)\right]
\end{aligned}
$$

For each epoch, MI was calculated for every pair of channels. MI considers two signals as a probability distribution, and calculates the distance of joint probability distribution and marginal distribution with joint distribution as weights. Pearson correlation coefficient is frequently used to compare two variables. However, MI was chosen to closely replicate the environment from the proposed CNN model. There were a total of 3988 MI adjacency matrices. Explicit equations for MI in comparison to pearson correlation are shown above.

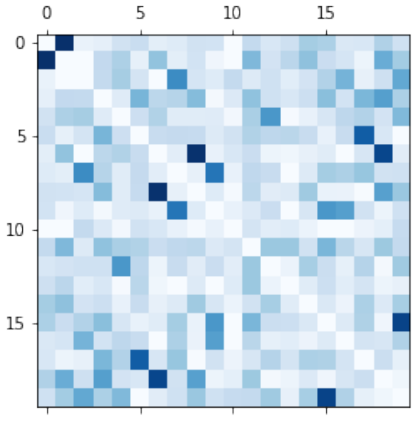


Fig. 3: Sample mutual information adjacency matrix. Graph edges are undirectional because matrix is symmetrical.

D. Graph Construction

MI adjacency matrix represents how close two node values are in probability. It makes more intuitive sense to represent these values as weights of a graph. However, the majority of the current GNN models train to learn node embeddings from feature vectors. So we considered each row in the MI table to be the initial feature vector for GNN layers with unweighted edges. Self-loops were allowed, but diagonal values in the MI matrix were set to 0.

E. Image Construction

Order of the channels in MI matrices is insignificant for graph construction. However, it affects CNN because MI matrices are considered as an image where convolution kernel is applied to nearby values. We followed the previous model's method to divide the nodes into 7 brain regions and reorder and reiterate the channels. This is to have electrodes that are physically close to each other get mapped close by in MI representation as well. Exact electrode channel sequence that we recreated is as follows: {3, 11, 1, 17, 2, 12, 4, 6, 14, 16, 10, 8, 19, 8, 18, 6, 4, 17, 3, 5, 7, 19, 9, 15, 7, 13, 5, 3, 11}. 9 channels were duplicated and 1 channel was repeated three times resulting in a MI table of size 2929. Four of the MI matrices were then stacked into a three dimensional tensor for CNN training.

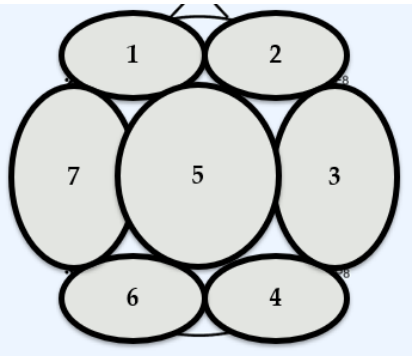


Fig. 4: Nodes were grouped by regions. Starting from 1, we have left frontal, right frontal, right temporal, parietal, left occipital, left temporal

F. Modeling and Training

Here we describe an overview of the GNN convolution layer. K-hop neighborhood of a node refers to all the nodes that can be reached from starting node with k-many edges. Let k denote k^{th}-hop layer; let h_v^{k-1} be a node with index v;let $\mathcal{N}(v)$ denote the set of neighbors of h_v; let $h_{\mathcal{N}(v)} = \{h_u^{k-1} \mid u \in \mathcal{N}(v)\}$ be set of all $k-1$ hop neighbors of h_v. Then, when $AGGREGATE$ function is applied to a node vector, h_v^{k-1}, it combines all the feature vectors in $h_{\mathcal{N}(v)}$ by taking an average. Learnable weight matrix W^k is applied to transform the number of features and find a new embedding for a given node [7]. Each convolution layer was followed by ReLU nonlinearity, and global mean was calculated to obtain graph-level embedding. Dropout layer with probability=0.5 was applied before the linear layer for binary classification. Softmax was used before the final prediction.

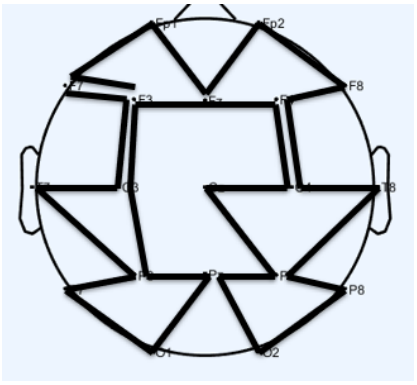


Fig. 5: Detailed node ordering is shown. It is clear that there are overlapping nodes and repeated nodes.

GCN is one of the most popular convolution methods for GNNs. It takes the mean of all the neighboring nodes including itself [7]. On the other hand, SAGE concatenates node feature vector with the mean of neighbors before applying weight matrix [8]. We also modeled DCRNN for comparison [9]. GNN is a fast growing area of research and there are more convolution layers that can be taken into consideration. Each of the three models were trained for k-hop values ranging from one to four. Models were trained using 10-fold cross validation with 50 epochs per fold. Test dataset was constructed separately by taking the mean of all MI matrices for each patient. For the CNN model, we replicated proposed model 'a' [4]. Mean of all images for each patient was used for the test dataset for CNN as well. For all models, NAdam

Fig. 6: Comparison CNN Model (Chen et al.)

optimizer was used with the following parameters: learning rate=0.001, betas = (0.9,0.999), momentum decay=0.004. All models were implemented using PyTorch, and PyGeometric.

V. Results and Discussions

TABLE I: Accuracy Comparison of All Models

DCRNN				
K-hop	1	2	3	4
Train	81.09	81.98	85.59	86.35
Validation	80.15	84.67	85.46	89.47
Test	81.59	82.85	85.68	87.14
GCN				
K-hop	1	2	3	4
Train	76.91	82.81	90.28	90.84
Validation	76.13	84.42	88.44	89.95
Test	75.53	81.92	89.92	90.25
SAGE				
K-hop	1	2	3	4
Train	82.26	95.52	98.22	98.86
Validation	82.91	92.46	96.48	96.98
Test	82.05	95.46	97.97	98.65
CNN				
Train	99.33			
Validation	100			
Test	94.21			

We first noticed that CNN suffered overfitting problems more than any GNN models. CNN ob-

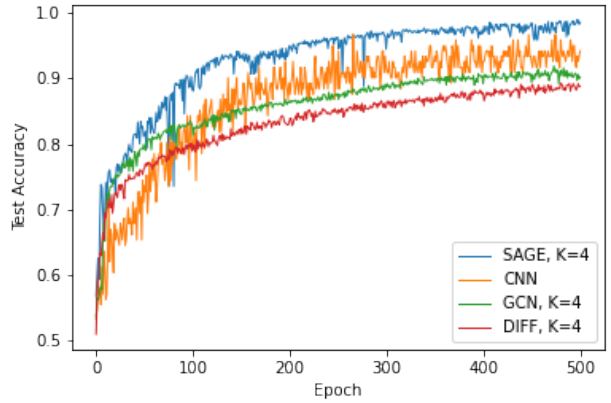


Fig. 7: Best performance models were SAGE, CNN, GCN, DCRNN in order.

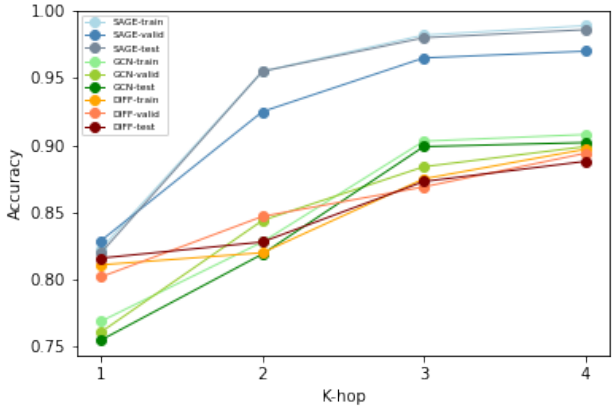


Fig. 8: For all three GNN models, accuracy increased along with k-hop values.

tained 100% accuracy on training and validation set, but test performance was 94.21% showing around 6% gap. This is likely due to the difference in number of layers used in GNN models and CNN model. Implimented CNN model has six convolution layers total whereas GNN models had four at maximum. Also, SAGE outperformed proposed CNN for k-hop values two, three, and four, showing our initial claim. For K=4, SAGE gave test accuracy of 98.65%.

GCN's accuracy was noticeably lower than SAGE. This was also an expected result since GCN "do not scale to large graphs or are designed for whole-graph classification." [8]. We believe that SAGE's structure being less prone to

over-smoothing contributed to this result. Interesting observation was that accuracy of all three models seemed to be increasing as k-hop values increased. As k-hop value gets bigger, model training is influenced by nodes that are farther away. Generally, if k-hop values get too big, GNN suffers from over-smoothing problem. We suspect that the nature of the brain connectivity data of ADHD patients allows our models to perform better when information from nodes that are far is taken into consideration.

VI. Further Research

There are number of ways this work can be extended for better accuracy.

- There are number of hyperparameters that can be tuned in the model such as: optimizer, hidden channels, number of epochs.
- To account for individual patient information and prevent data leaking, we will consider group k-fold method.
- Various node-level, graph-level metrics for sensitivity analysis can be added to our feature vector. We will observe how the model adjusts when different types of data are introduced.

There is room for better interpretibility as well.

- First, it will be interesting to observe what happens to model performance as we continue to increase k-hop values.
- By using a different method to construct Adjacency matrix or a feature matrix, we will be able to infer the type of information that is most relevant to ADHD research.
- We will omit some of the channels at random or by region to observe if there is a significant change in model performance. We will be able to infer which nodes or regions contribute more for prediction.
- Other models such as Graph Isomorphism Network, or BrainGNN model can be considered as well. These models provide their own interpretable results that can be reviewed.

References

[1] Danielson, Melissa L et al. "Prevalence of Parent-Reported ADHD Diagnosis and Associated Treatment Among U.S. Children and Adolescents, 2016." Journal of clinical child and adolescent psychology : the official journal for the Society of Clinical Child and Adolescent Psychology, American Psychological Association, Division 53 vol. 47,2 (2018): 199-212. doi:10.1080/15374416.2017.1417860

[2] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based Electroencephalography Analysis: A systematic review," Journal of Neural Engineering, vol. 16, no. 5, p. 051001, 2019.

[3] M. Adamou, T. Fullen, and S. L. Jones, "EEG for diagnosis of ADULT ADHD: A systematic review with narrative analysis," Frontiers in Psychiatry, vol. 11, 2020.

[4] H. Chen, Y. Song, and X. Li, "A deep learning framework for identifying children with ADHD using an EEG-based Brain Network," Neurocomputing, vol. 356, pp. 83–96, 2019.

[5] Ali Motie Nasrabadi, Armin Allahverdy, Mehdi Samavati, Mohammad Reza Mohammadi, June 10, 2020, "EEG data for ADHD / Control children", IEEE Dataport, doi: https://dx.doi.org/10.21227/rzfh-zn36.

[6] J. Rodrigues, M. Weiß, J. Hewig, and J. J. Allen, "EPOS: EEG processing open-source scripts," Frontiers in Neuroscience, vol. 15, 2021.

[7] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).

[8] Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." Advances in neural information processing systems 30 (2017).

[9] Li, Yaguang, et al. "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting." arXiv preprint arXiv:1707.01926 (2017).

[10] Demir, Andac, et al. "EEG-GNN: Graph Neural Networks for Classification of Electroencephalogram (EEG) Signals." 2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC). IEEE, 2021.