Summary

In this study, the model Graph WaveNet was used to decode movement intention electroencephalography (EEG) or the method to record an electrogram from the brain's electrical activity as this is a step toward improving the performance of motor imagery brain-computer interface systems and could be used in aiding motor-impaired patients.

The central focus of this project was adapting the spatial-temporal graph model Graph WaveNet to intake EEG motor movement data and classify it into one of four movement labels and one "no movement" label.

The data set used consists of EEG data from 8 subjects, four different motor imagery tasks, namely the imagination of movement of the left hand, right hand, both feet, and tongue, and one "no movement". Two sessions on different days were recorded for each subject. The algorithm has been adapted to take the EEG data and output a motor imagery task classification that would be measured in terms of accuracy, Kappa coefficient, F1 score, Area under the ROC curve, and speed of prediction.

To use a spatial-temporal model these two dimensions had to be imputed. The input for the spatial dimension was in the shape of an adjacency matrix where each electrode, that is placed in the international 10-20 system, had an adjacency value to one another and the input for the temporal dimension consisted of a window of movement intention EEG measurements. While comparing different configurations of the size of the window, two hundred ms or 50 EEG measurements at 250Hz sampling rate was the most eminent in all measured metrics.

This adaptation was very successful, in the bellow described environment, for the movement categories, the overall accuracy over 4 categories being ~96%, and the accuracy for the "no movement" label planned to be future improved at ~60%. It is worth noting that the "no movement" labels kept in the experiments are the hardest, in terms of location in the timeframe, to label them being the border before and after the motor movement.

Although the proposed method based on the GWN framework has achieved very excellent performance in the MI-EEG decoding task, there are still limitations of the method. Firstly, due to the great difference in MI-EEG signals among different subjects, the proposed method cannot realize cross-subject MI-EEG decoding and as can be seen in one of the configurations there is also a difference between the EEG signals among the same subjects but in different days.

Adapting Graph WaveNet to predict movement intention from cue-based BCI trials

Vlad Mataoanu, *Student CS-IT9, MSc Computer Science, Aalborg University*
E-mail: vmatao20@student.aau.dk

Abstract—Electroencephalography (EEG) othwerwise known as the method to record an electrogram from the brain's electrical activity alongside a precise classification algorithm could be a powerful tool in aiding motor-impaired patients. The central focus of this project has been the adaptation of the spatial-temporal graph model Graph WaveNet([3]) to intake motor movement EEG data and classify it into one of 4 movement labels and one "no movement" label having as a future application sending them as commands to an exoskeleton. Methods: The data set ([11]) consists of EEG data from 8 subjects, four different motor imagery tasks, namely the imagination of movement of the left hand (class 0), right hand (class 1), both feet (class 2), and tongue (class 3) and one "no movement" (class 4). Two sessions on different days were recorded for each subject. The algorithm has been adapted to intake the EEG data and output a motor imagery task classification that would be measured in terms of accuracy, Kappa coefficient, F1 score, Area under the ROC curve, and speed of prediction. This adaptation was successful, in the environment which is discussed later on in this paper, for the movement categories. The overall accuracy over 4 categories being 96% and the accuracy for the no movement" label being 60% and aiming to be improved in the future. It is worth noting that the data labelled as "no movement" kept in the experiments are the most difficult to classify due to their location in the timeframe, them being the border before and after the motor movement

Index Terms—EEG, Graph WaveNet, Movement intention, Cue based trial.

1 Introduction

Multiple reasons could cause an individual to lose their motor skills. Stroke, spinal cord injury or Parkinson's disease often result in such loss and therefore daily activities become difficult to be performed. If physiotherapy fails, using an exoskeleton could be the only possible solution to provide a normal life to the individual or as close to a normal life as possible.

Achieving reasonable control over the exoskeleton would be a challenge if the movement is predicted with high latency and low accuracy compared to the movement intention. That is why being time-wise as close to the subject's movement intention as possible is desired. Being non-invasive, EEG was used as the preferred brain-computer interface (BCI) compared to Electrocorticography or Microelectrode arrays where surgical procedures are required before their setup. Compared to Magnetoencephalography for example, the physical equipment of an EEG is compact, having the potential to become mobile. The proposed MI-EEG decoding method in this study has great promise to improve the performance of the motor imagery brain-computer interface system.

Deep learning has been used to handle large amount of complex data, as it is known to increase its performance with the increase of data quantity. Data from eight subjects was available (4608 movement intention events).

1.1 Background

1.1.1 EEG and BCI background

Electroencephalography (EEG) is an electrophysiological monitoring technique for recording electrical activity on the scalp, which has been proven to correspond to the macroscopic activity of the brain's surface layer beneath. The electrodes are positioned along the scalp, making the procedure non-invasive.

EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain. Clinically, EEG refers to the recording of the brain's spontaneous electrical activity over a period, as recorded from multiple electrodes placed on the scalp. [1]

A brain-computer interface (BCI), is a direct communication pathway between an enhanced or wired brain and an external device. BCIs are often

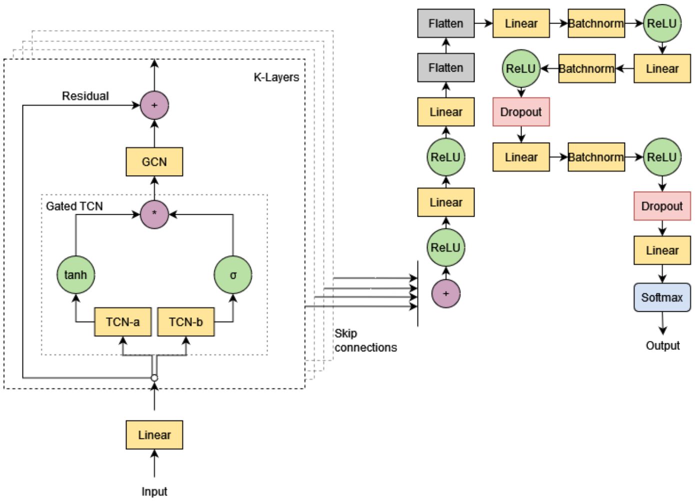


Fig. 1. The framework of the adapted Graph WaveNet

directed at researching, mapping, aiding, augmenting, or repairing human cognitive or sensory-motor functions. [2]

Thus, EEG is a monitoring method for recording electrical activity in the brain that is widely utilized as an input to the BCI.

1.2 Graph WaveNet

Graph WaveNet [3] is a deep learning model that is used to solve spatial-temporal data analysis problems. It is a CNN-based method that solves two issues: how to capture spatial and temporal dependencies at the same time, and how to improve the spatial-temporal graph modeling's ability to learn temporal dependencies.

It employs a graph convolution layer in which a self-adaptive adjacency matrix, that holds the adjacency value between any two nodes, is improved from the data via end-to-end supervised training to retain hidden spatial connections. The graph's predecessor is the self-adaptive adjacency matrix, which is used in its generation.

The framework of Graph WaveNet, which is in figure 1, consists of K spatial-temporal layers on the left and, on the right, an output layer. The initial step of the model focuses solely on temporal dependencies. The inputs are initially modified with a linear layer before being copied three times. Two copies are subjected to distinct temporal convolution layers, as well as tanh and sigmoid activation functions. After which element-wise multiplication is applied between the two. The sigmoid function forces the elements to a value between 0 and 1 and this is used to potentially weaken some of the outputs of the tanh layer. This method is known as Gating. After the gated TCN module, each spatial-temporal layer contains residual connections and is skip-connected to the output layer.

2 METHOD

2.1 Data

The data used in this project comes from the 2a dataset of the BCI Competition 2008 [10]. It contains 22 EEG channels, three EOG channels, a sampling rate of 250Hz, four action classes (left hand, right hand, feet, tongue), and two sets for each of the eight subjects (one labeled and one unlabeled). Figure 3 illustrates the EEG montage. The EOG channels were discarded from use as it was considered that in a real-life scenario a subject would not get a visual cue as to when a movement had to be executed.

The subjects were sitting in a comfortable armchair in front of a computer screen. At the beginning of a trial (t = 0 s), a fixation cross has appeared on the black screen, and a short acoustic warning tone was played. After two seconds (t = 2 s), a cue in the form of an arrow pointing either to the left, right, downwards, or upwards (corresponding to one of the four classes left hand, right hand, foot, or tongue) appeared and stayed on the screen for 1.25 s. This prompted the subjects to perform the desired motor imagery task. At this point, no feedback was provided. The subjects were asked to carry out the motor imagery task until the fixation cross disappeared from the screen at t = 6 s. [11] The timing scheme of the procedure can be seen in figure 2.

Per subject, two sessions were recorded on different days for the 8 subjects. Each one of the 16 sessions is comprised of a total of 288 trials, 72 for each of the four potential classes, with the cues being presented in a random order to ensure the balance of the data. Thus, giving a total of 4608 peak movement intentions.

There will be 2 different settings discussed further in the experiments section: one where the model is trained on one of the 2 sessions for each subject and tested on the rest and one where it is trained on a fraction combination of the 2 sessions and tested on the whole 2 sessions.

Generating the data

To begin, because the data was sparsely labeled (4608 timestamps out of around 12 million timestamps were tagged as the peak of the movement), it was decided to cut windows around the labeled

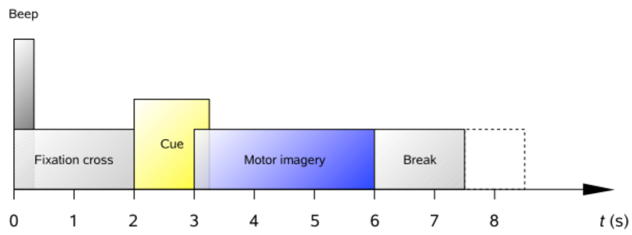


Fig. 2. Timing scheme of the procedure [11]

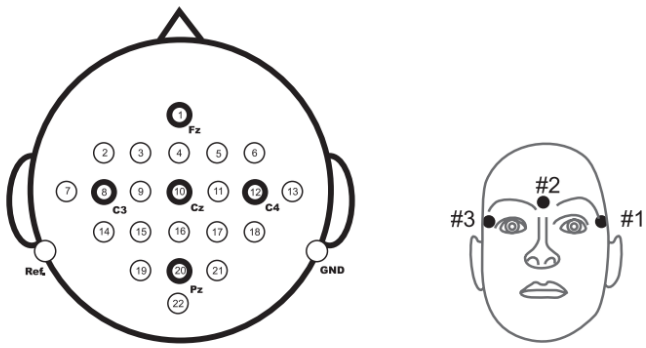


Fig. 3. Left: Electrode montage corresponding to the international 10-20 system. Right: Electrode montage of the three monopolar EOG channels. From the description of the 2a dataset that can be found here: [11]

timestamps to level out the amount of data across the different classes. In the end, the best windows set-up would have a length of 50 time-frames.

It can be seen in figure 4 that the data set-up of windows composed of 50 EEG measurements - meaning that 50 EEG snapshots will be used as temporal data and will be labeled one of the 4 movement labels or ”no movement”. This data setup is the exact one used for testing the various configurations of the model. The window is represented by the curly bracket and its color depicts the label of that window. If the window contains a movement event (in the figure colored in red and blue) it will be labeled as the movement event (containing the event) and if it does not, it is colored black and will be labeled as ”no movement”. With the best algorithm setup, there are 12 ”no movement” windows before the 50 movement labeled windows and 12 ”no movement” labeled windows after. Because of the limited time of the project, data in between the 74-sized slices was discarded as as it would have unbalanced the data without any data augmentation (more about this topic will be further detailed in the discussion section) and this is why a separator was necessary depicted as the grey colored rectangles. Without it there would be ”no movement” labeled windows with data from 2 distant slices.

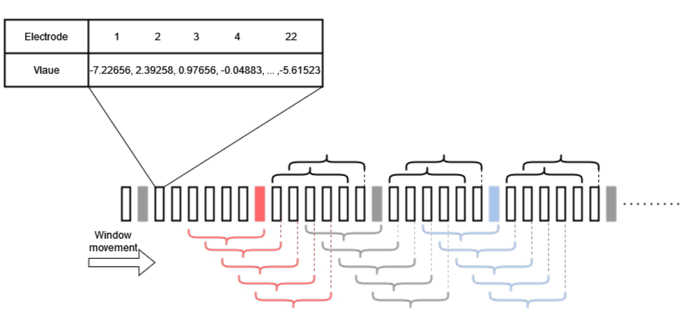


Fig. 4. Window of EEGs labeling pattern. The bracket represents the window of EEGs (typically 200 ms long) that is sequentially sent to the model for classifying. The red and blue rectangles are one row of EEG data that are labeled as movement events. The figure illustrate a perfect classification, whether a window contains a movement (red or blue) or not (black).

The EEG measurements were recorded at a sampling rate of 250Hz, meaning that 50 EEG measurements would equate to 200 ms long windows. All of the eight sets were used to generate the train and test batches. Each electrode was given Cartesian coordinates, electrode 1 being given the (0,0) coordinates and as it can be seen in 3, the electrodes 3 and 4 were given (-1,-1) and (0,-1) respectively. The rest of the electrodes were assigned coordinates that followed this pattern, with electrode 1 being the anchor point. As for the adjacency matrix, Electrodes with a Euclidean distance greater than 2 to one another would get an adjacency value of 0 and the electrodes with a distance of 2 or less to one another (e.g. 10 and 1) would get an adjacency calculated by the formula: (2-Euclidean distance)/2, while each electrode would get an adjacency value of 1 to themselves (e.g. 10 and 10). This following the example of the adjacency matrix from the Graph WaveNet algorithm ([3]).

2.2 Algorithm adaptation

The original version of the Graph WaveNet algorithm [3] was designed to forecast a given number of future data timestamps using past timestamps as input. The method had to be modified, since the goal of the project has not been to predict future EEG measurements based on previous ones, but to accurately classify historical data into one of four movement classes and one ”no movement” class.

Model

The output of the model from figure 1 before the two flatten layers is a tensor of shape: batch size * window length * the number of nodes * value of a node. The desired output is batch size * probability of action list where the batch size is a hyper parameter, and the action list is a list of probabilities for each class that sums up to 1 (e.g. (0.14, 0.08, 0.1, 0.7, 0.08) where the fourth movement has the highest probability). This was achieved by taking the previously mentioned tensor, flattening two of its dimensions, applying linear batchnorm2 and relu activation 3 times, dropout 2 times, and followed by a softmax.The final shape would be batch size * probability of action list.

3 Experiments and results

Part of the scope of this project was to find the best configuration of the model. This search started with the wnm (window and "no movement" size) which would dictate how much past data would be used by the model for classifying the movement and how many adjacent EEG measurements would be kept to be used for the "no movement" label.

The testing, except for the configurations 1,2 and 19, is done on a 12 - 50 - 12 ("no movement" - movement - "no movement") windows set-up. Meaning that 12 windows each containing 50 EEG measurements that do not contain a movement intention are saved before and after the 50 windows each of 50 EEG measurements that contain a movement intention. For configurations 1 and 2 the set-up is similar but smaller in window size: 4 - 12 - 4 and 6 - 26 - 6 respectively.

To keep the data balanced without data augmentation, as there are 4 movement labels, when 50 windows are labeled and saved for training, for each movement, approximately 12 "no movement" windows had to be collected as well.

3 ways to approach the problem were compared, and illustrated in figure 5, as collecting all of the 24 "no movement" windows for each 50 movement windows would unbalance the data (1:1:1:1:2):

- all the "no movement" windows were collected and then selected at random until they match the size of one of the movement labels
- sequentially switching from collecting the 12 "no movement" windows before the movement windows to the "no movement" windows after, thus using 50% of this label's the data to be split into training and testing data
- collect 6 "no movement" windows before the movement windows and 6 "no movement" windows after

Following with configuration experiments with different train-test splits and different activation functions for the last 3 sets of layers. The configurations 15 to 18 show that having the graph convolution layer adds an additional 8% and that the adaptive adjacency matrix does not have significant impact on the specific configuration.

Having 2 sessions on different days per subject, for configuration 19 the data split was done so, one of the sessions was used for training and one for testing. For the rest of the configurations data from both sessions was used for training in a proportion indicated by the train/test column. As the results were not satisfactory, configurations 20 to 23 were used to check how little amount of data was needed from both sessions to get adequate results.

The parameters that are common and unchanged between the different configurations are: dropout of 0.3, 20 epochs, the number of nodes 22, the seed is 1, weight decay of 0.0001, adjtype set as doubletransition, and nhid as 32. The speed of prediction for all of the configurations was similar, being approximately 1 millisecond for each window of EEGs. Thus being able to keep pace with a live data stream of EEGs, in the current configuration being 250Hz or one EEG every 4 milliseconds.

The experiments were conducted under a computer environment with one Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and one NVIDIA GeForce RTX 2060 Mobile GPU card.

4 Discussion

It can be seen in table 2 that the "no movement" label has significantly worse accuracy than the other labels. As it was mentioned earlier, without data augmentation, keeping all of the "no movement" windows would greatly imbalance the data as it is sparsely populated with the movement intentions. The "before or after" scenario worked the best in the tested configurations giving the highest accuracy of 62.3% for this label. Only 35% of the kept "no movement" windows have been used for training in that configuration (70% of the train/test split * 50%, being the second way of keeping the data balanced mentioned in the previous section).

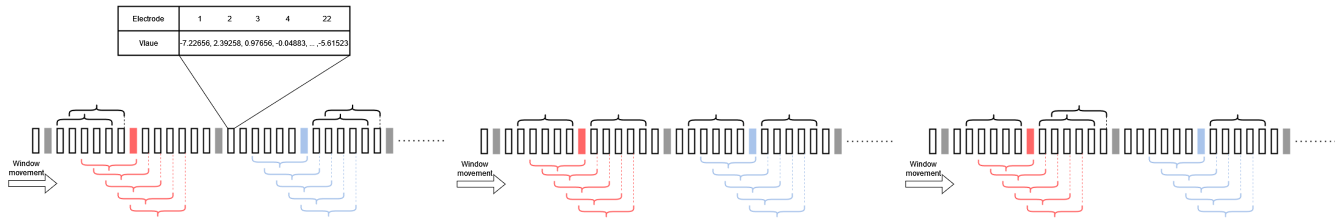
Augmenting the movement data in order to use all of the "no movement" data would be a first future step toward improving the capabilities of this model. This could be done in a similar manner as shown in the article [9]: where parts of the movement intention windows are combined until the desired data balance is achieved.

TABLE 1
Tested model configurations.
wnm Ratio of movement and no movement windows saved for training at each trial.
”no movement” How the ”no movement” windows are selected to be saved compared to the movement windows. An example can be seen in figure 5

parameters								model configurations
wnm	”no movement”	train/test	lr	batch	activation	gcn	addaptadj	
12:4	before or after	70/30	0.001	32	relu	TRUE	TRUE	config 1
24:6	before or after	70/30	0.001	32	relu	TRUE	TRUE	config 2
50:12	before or after	70/30	0.001	32	relu	TRUE	TRUE	config 3
50:12	**random**	70/30	0.001	32	relu	TRUE	TRUE	config 4
50:12	**before and after**	70/30	0.001	32	relu	TRUE	TRUE	config 5
50:12	**before or after**	**60/40**	0.001	32	relu	TRUE	TRUE	config 6
50:12	before or after	**80/20**	0.001	32	relu	TRUE	TRUE	config 7
50:12	before or after	**70/30**	0.001	32	**tanh**	TRUE	TRUE	config 8
50:12	before or after	70/30	0.001	32	**sigmoid**	TRUE	TRUE	config 9
50:12	before or after	70/30	**0.01**	32	**relu**	TRUE	TRUE	config 10
50:12	before or after	70/30	**0.0001**	32	relu	TRUE	TRUE	config 11
50:12	before or after	70/30	**0.00001**	32	relu	TRUE	TRUE	config 12
50:12	before or after	70/30	**0.0001**	**16**	relu	TRUE	TRUE	config 13
50:12	before or after	70/30	0.0001	**64**	relu	TRUE	TRUE	config 14
50:12	before or after	70/30	0.0001	**128**	relu	TRUE	TRUE	config 15
50:12	before or after	70/30	0.0001	128	relu	**FALSE**	**FALSE**	config 16
50:12	before or after	70/30	0.0001	128	relu	FALSE	**TRUE**	config 17
50:12	before or after	70/30	0.0001	128	relu	**TRUE**	**FALSE**	config 18
50:12	before or after	**labeled/unlabeled**	0.0001	128	relu	TRUE	**TRUE**	config 19
50:12	before or after	**60/40**	0.0001	128	relu	TRUE	TRUE	config 20
50:12	before or after	**50/50**	0.0001	128	relu	TRUE	TRUE	config 21
50:12	before or after	**40/60**	0.0001	128	relu	TRUE	TRUE	config 22
50:12	before or after	**30/70**	0.0001	128	relu	TRUE	TRUE	config 23

TABLE 2
Results of configurations. Configurations: 1,2 and 19 had different testing environments detailed in the experiments and results section

configurations	4 labels				5 labels				average accuracy labels (%)				
	accuracy	kappa	F1	Auc	accuracy	kappa	F1	Auc	L1	L2	L3	L4	Lnm
config 1	39,14%	0,22	0,42	0,66	27,61%	0,12	0,30	0,62	26,20	44,21	41,54	44,59	10,26
config 2	26,52%	0,05	0,32	0,54	20,56%	0,03	0,23	0,53	13,11	52,59	10,50	29,55	10,48
config 3	80,1%	0,77	0,83	0,95	66,9%	0,59	0,68	0,9	80,07	78,05	81,04	81,23	41,53
config 4	76,2%	0,72	0,79	0,93	63,4%	0,54	0,65	0,88	80,24	71,90	78,14	74,56	38,63
config 5	75,8%	0,73	0,8	0,92	63,5%	0,55	0,65	0,88	75,65	75,21	74,53	77,86	39,80
config 6	72,7%	0,66	0,75	0,9	59,2%	0,5	0,61	0,86	71,10	72,08	76,39	71,16	33,40
config 7	75,1%	0,72	0,79	0,92	64,1%	0,55	0,65	0,87	74,18	71,90	79,09	75,32	43,03
config 8	29,0%	0,05	0,36	0,56	20,9%	0,05	0,27	0,58	0,27	43,45	8,17	63,82	5,46
config 9	23,5%	0,04	0,35	0,55	27,7%	0,09	0,31	0,59	2,33	4,72	76,85	10,41	35,79
config 10	19,0%	0	0,4	0,5	21,7%	0,01	0,28	0,51	0,00	75,72	0,00	0,00	27,00
config 11	97,0%	**0,98**	**0,98**	**1**	85,6%	**0,82**	**0,87**	0,98	98,29	97,59	98,12	98,01	61,81
config 12	95,9%	0,96	0,96	**1**	85,0%	0,81	0,86	**0,99**	95,92	97,13	97,16	97,22	**62,31**
config 13	96,1%	0,96	0,97	**1**	84,1%	0,8	0,85	0,98	97,03	96,50	97,85	96,93	59,01
config 14	97,0%	0,97	**0,98**	**1**	**85,7%**	**0,82**	**0,87**	0,98	98,09	98,08	**98,22**	97,76	62,09
config 15	**97,4%**	**0,98**	**0,98**	**1**	85,4%	**0,82**	0,86	0,99	**98,64**	**98,66**	98,20	97,89	60,58
config 16	89,3%	0,88	0,92	0,98	79,9%	0,75	0,8	0,96	87,46	91,89	88,84	90,00	61,21
config 17	90,2%	0,88	0,91	0,98	77,6%	0,72	0,79	0,95	91,00	91,15	88,52	90,14	53,27
config 18	97,9%	0,98	**0,99**	**1**	**85,7%**	**0,82**	**0,87**	0,98	97,95	97,73	97,88	**98,13**	62,06
config 19	20,84%	0,01	0,26	0,51	25,3%	0,06	0,24	0,54	24,03	21,22	17,66	20,44	33,89
config 20	94,3%	0,93	0,94	0,99	81,9%	0,78	0,81	0,98	94,10	96,16	93,72	92,18	56,57
config 21	91,1%	0,88	0,91	0,98	78,1%	0,72	0,79	0,96	91,23	91,33	88,86	90,20	53,3
config 22	83,77%	0,81	0,86	0,97	73,23%	0,66	0,74	0,94	83,84	84,13	84,30	82,79	52,95
config 23	70,33%	0,66	0,74	0,91	64,52%	0,55	0,65	0,88	73,05	70,44	71,5	66,31	53,36



(a) before or after (b) before and after (c) random

Fig. 5. wnm and ”no movement” windows selection for training example. With a window size of 5 (5 EEG measurements) and wnm of 5:2

TABLE 3
The 4-label accuracy comparison of my method and some state-of-the-art methods. In this table, S1–S9 denotes the eight subjects in the experimental dataset, respectively.

Method	S1	S2	S3	S5	S6	S7	S8	S9	Average accuracy (%) std
FBSF-TSCNN [7]	85,8	60,1	87,8	48,6	56,9	83	81,6	80,2	71,57 ± 14,71
C2CM [8]	87,5	65,28	90,28	62,5	45,49	89,85	83,33	79,51	75,47 ± 15,08
Multi-Branch 3D CNN [4]	77,4	60,14	82,93	75,84	68,99	76,04	76,85	84,66	75,36 ± 7,27
TSSM + LDA [6]	81,8	62,5	88,8	62,9	58,5	86,6	85,1	90	77,03 ± 12,45
Functional brain network [5]	82,8	65,5	87,9	72,4	70,7	82,8	87,9	89,7	79,96 ± 8,58
TB 3D CNN [9]	93,8	70,8	93,53	79,89	60,05	96,14	92,15	84,3	83,83 ± 12,07
My method (Addapted GWN)	**97,24**	**97,04**	**97,42**	**97,75**	**97,44**	**97,49**	**96,99**	**96,48**	97,23 ± 0,34

TABLE 4
The 4-label Kappa value of my method and some state-of-the-art methods. In this table, S1–S9 denotes the eight subjects in the experimental dataset, respectively.

Method	S1	S2	S3	S5	S6	S7	S8	S9	Average accuracy (%) std
FBSF-TSCNN [7]	0,77	0,33	0,77	0,35	0,36	0,71	0,72	0,83	0,61 ± 0,2
C2CM [8]	0,81	0,468	0,838	0,315	0,426	0,773	0,755	0,736	0,64 ± 0,19
Multi-branch 3D CNN [4]	0,7	0,459	0,788	0,647	0,538	0,653	0,702	0,713	0,65 ± 0,1
TSSM + LDA [6]	0,83	0,537	0,887	0,5	0,273	0,861	0,778	0,727	0,67 ± 0,2
Functional brain network [5]	0,77	0,54	0,84	0,63	0,61	0,77	0,84	0,86	0,73 ± 0,11
TB 3D CNN [9]	0,92	0,59	0,913	0,727	0,466	0,948	0,892	0,787	0,78 ± 0,16
My method (Addapted GWN)	**0,98**	**0,98**	**0,98**	**0,98**	**0,98**	**0,98**	**0,97**	**0,97**	0,98 ± 0,01

Due to time constraints finding the best model configuration with the presented changes is incomplete, as it would require each parameter change to be done in combination with others. As well as having multiple rounds of running the experiments to account for the entropy.

The Euclidean distance of 2, found in the adjacency matrix calculation, is an arbitrary threshold. The model could yield better results with a different threshold value, as there could be unaccounted for spatial dependencies between further electrodes.

Although the proposed method based on the GWN framework has achieved very excellent performance in the MI-EEG decoding task, there are still limitations of the method. Firstly, due to the great difference in MI-EEG signals among different subjects, the proposed method cannot realize cross-subject MI-EEG decoding and as can be seen in the 19th configuration there is a difference between the EEG signals among the same subjects but on different days.

5 Conclusion

In this study, Graph WaveNet is proposed as a mean to predict movement intention from cue-based BCI trials. First, the algorithm is adapted to intake data from the BCI Competition, the adjacency matrix is changed to include the logic from the international 10-20 system of the electrodes montage found in figure 3, and to output classification of this data into one of the four movements intentions and a no movement. This adaptation was successful in the presented environment, the overall best accuracy over four categories being approximately 97% and the one for the five labels being approximately 85%.

REFERENCES

[1] Niedermeyer E. and da Silva F.L. (2004). Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Lippincott Williams and Wilkins. ISBN 978-0-7817-5126-1

[2] Krucoff, Max O.; Rahimpour, Shervin; Slutzky, Marc W.; Edgerton, V. Reggie; Turner, Dennis A. (1 January 2016). ”Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation”. Frontiers in Neuroscience. 10: 584. doi:10.3389/fnins.2016.00584. PMC 5186786. PMID 28082858

[3] Graph WaveNet for Deep Spatial-Temporal Graph Modeling Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Chengqi Zhang Centre for Artificial Intelligence, FEIT, University of Technology Sydney, Australia Faculty of Information Technology, Monash University, Australia arXiv:1906.00121v1 [cs.LG] 31 May 2019

[4] Zhao X, Zhang H, Zhu G, You F, Kuang S and Sun L 2019 A multi-branch 3D convolutional neural network for EEG-based motor imagery classification IEEE Trans. Neural Syst. Rehabil. Eng. 27 2164–77

[5] Ai Q, Chen A, Chen K, Liu Q, Zhou T, Xin S and Ji Z 2019 Feature extraction of four-class motor imagery EEG signals based on functional brain network J. Neural Eng. 16 026032

[6] Xie X, Yu Z L, Lu H, Gu Z and Li Y 2017 Motor imagery classification based on bilinear sub-manifold learning of symmetric positive-definite matrices IEEE Trans. Neural Syst. Rehabil. Eng. 25 13

[7] Chen J, Yu Z, Gu Z and Li Y 2020 Deep temporal-spatial feature learning for motor imagery-based brain-computer interfaces IEEE Trans. Neural Syst. Rehabil. Eng. 28 2356–66

[8] Sakhavi S, Guan C and Yan S 2018 Learning temporal information for brain-computer interface using convolutional neural networks IEEE Trans. Neural Netw. Learn. Syst. 29 5619–29

[9] Two-branch 3D convolutional neural network for motor imagery EEG decoding Lie Yang et al 2021 J. Neural Eng. 18 0460c7

[10] BCI 2008 competition http://www.bbci.de/competition/iv/ Accessed on 20-12-2021

[11] BCI 2008 2a dataset's description provided by the Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology, (Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, Gert Pfurtscheller) http://www.bbci.de/competition/iv/desc_2a.pdf Accessed on 20-12-2021