Replies to the Reviewers' Comments

Thank you for your careful viewing of our project presentation and your constructive feedback. Please find below our responses to all of the reviewer comments.

Reviewer 1 Comments:

Group 81 research on the MRCP waveforms classification. The waveform classification is really a new field to me, and I'm quite interested in this field. The presentation talked about the dataset they used, several different ways of preprocessing, like CSP, Wavelet-BSS. Also they have tried several different models to solve this problem, like LSTM and SVM. They really did lots of work to improve the result. Good job guys!

Thank you.

bullet Some of the slides are messy. Maybe next time you can try to modify the size of the figure.

As per your suggestion, we have clarified the figures we use in our manuscript.

bullet A demo table of the dataset is much better than a screen shot of Matlab.

Thank you for your comment. Because our dataset consisted of over 60 hours of EEG recordings, we couldn't show enough information in a table. Additionally, the intent of the MATLAB screenshot was to demonstrate in more detail how the dataset is stored and organized.

bullet Add a Content page will be helpful. To be honest, I have to watch your video 2 or 3 times to figure out the mainline of your presentation.

We apologize for the confusion. We will make sure that the organization of our manuscript is better represented than in our project presentation.

bullet The problem description is not that clear. Actually you just introduced some background information about your task but not the problem you wanted to solve.

Thank you for your comment. We will better define the problem we are attempting to solve in the report. The problem we wanted to solve was to study how we can use machine learning techniques compared to the recently published methods for MI decoding, and to see if they would offer an improvement over them.

bullet I can't find the correlation between everyone's work. It's kind of like the advice 2. The video gives me a feel like you just done three or four small projects individually.

The correlation of work was as follows: Daniel wrote the code for the baseline LSTM model, as well as how to parse the datasets following some of Zhimin's initial research.

Nicholas was in charge of implementing some pre-processing techniques, which would hopefully increase the accuracy of the LSTM compared to traditional methods in the literature.

bullet The result is not that clear. If you guys are trying to solve a specific question, you need to show the result or the improvement. Or maybe you guys are making a review of this kind of problem.

Thank you for your comment. We have more clearly presented the results of our research in the manuscript.

bullet I actually pretty like your coding part. You guys are introducing your code passionately and clearly. It's much better than your presentation.

Thank you.

Reviewer 2 Comments:

The topic of choice was interesting and has a meaningful application. The literature review was extensive. It was a lot of information for a slide, but the presenter brought to light all of the important takeaways. SVM, Neural Networks, MLP, Preprocessing techniques include downsampling and low pass filters. The group considered the performance of the classifier online and offline, which seems fairly advanced for the scope of this project. They explored the different paradigms or domains in which to do learning. Feature engineering included the Common Spatial Patterns (increasing the difference between each movement) and the application of PCA and ICA. This project provided an introduction to a new kind of model, the LTSM ("Long Short-Term Memory"). This seems related to statistical signal processing techniques, and seems like a good choice given the dataset. It works to apply pattern recognition over a long and shorter span of time (i.e., 100 to 50).

Thank you.

bullet Application of wavelets in pre-processing was brilliant, I needed a bit more clarification on why a non-orthogonal one was selected, and how it affects ICA.

The ICA requires an orthogonal wavelet transformation because orthogonal wavelet transforms maintaining the variance-covariance matrix of the processed data. This makes it easier to apply PCA and ICA after the transform because they use either second order or higher order statistics reliant on the variance-covariance matrix. If we did not use an orthogonal wavelet, then transforming back from the ICA or PCA space may cause bithrogonality or other strange hiccups in the source space after we inverse transform everything.

bullet Was there an application of the kNN that you saw good performance in the Lit Review, in conjunction with the SVM? I'm not familiar with the wavelet you are using, but typically you have various frequencies encoded in the different wavelet decompositions, in cleaning a signal, the lowest frequencies of the signal are returned as a 'clean' signal. Perhaps there is information necessary to *your classification in the higher frequency components, not sure if you have access to that in your processing chain. Seems like it might be worth feeding it through your network though.*

In our research, we did not find an application of kNN with SVM. Our focus was to apply the LSTM to EEG processing.

bullet Great work overall, it seems that every member of the team worked at addressing a different technique presented in literature. Nice work.

Thank you.

Reviewer 3 Comments:

bullet A bit of more background on what is the MRCP would be helpful to understand how this information fit into the 22 channeled data.

The MRCP is a phenomena known to occur in EEG signals recorded over the motor cortex that results in a steady decrease in amplitude for both real movement and imagined movement. Whether or not this occurs for each channel in the data is beyond the scope of our work, but the intention is that perhaps the MRCP may be different for different kinds of movements (hands, legs, fingers, tongue, etc). We will clarify this in our manuscript.

bullet Assume that you are going to implement the other technique among the five mentioned techniques that would be useful to denoising, the kalman filter might not be a very good method given the context of your problem, as it will require the use of a motion model and/or observation model, unless you have understand how each of the waves is formed physically, or make use of UKF or EKF. SVM might be a good method to use. On the other hand, doing fourier analysis might be helpful to understand which are artifact signals and filtering them out.

Due to time constraints, we unfortunately cannot implement all of the five mentioned techniques. Rather, our intention was to use the LSTM model as a new model for decoding MI tasks and compare it to the SVM, which has been widely used in the literature.

bullet To address the overfitting issue, you might want to consider doing regularization (L2) while using the RNN, in our project we also used a very large RNN with LSTM cells, we added dropouts in between the LSTM cells to reduce the overfitting.

While we did not use weight regularization when training the RNN, we indeed have introduced dropout between layers to both reduce overfitting and improve testing accuracy.

bullet The preprocessing technique (Wavelet BSS) is well explained. For the Wavelet BSS in your *code explanation, it might be better to go slower for the audience to understand step by step. Clean data and noise data visualization is very good.*

Thank you for your comment. We will be sure that the explanation of Wavelet BSS is concise yet clear to understand in our manuscript.

bullet For SVM, you could always use more cases because the noise type might be more than one. The PowerPoint has the SVM slide, but I think the ending of the video is cut. For your case if you want to classify the data in multiple cases, you could also just use k-means.

Thank you for your suggestion. While we can use k-means clustering, we intend to compare the performance of the SVM, which is a widely used classification technique for decoding MI tasks, to our proposed LSTM model.

Machine Learning for EEG Motor Imagery Decoding

Zhimin Liang, Daniel Valenica, and Nicholas Wong
Department of Electrical and Computer Engineering
University of California San Diego, San Diego, U.S.A.

***Abstract*—Motor imagery (MI) decoding is an important task for brain-computer interfaces (BCIs), which could potentially pave the way to allow victims of spinal cord injury or other diseases to regain mobility via neural signal processing. However, the performance of previously used classifiers for MI decoding have provided unreliable results and are usually tuned per patient. In this article, we propose to utilize temporally-sensitive machine learning models for MI decoding of electroencephalographic (EEG) recordings. We show that combining data from several test subjects and training with well-known machine learning techniques yields the highest accuracy in MI decoding, as opposed to previous approaches. We also explore how pre-processing techniques can be utilized to build robust models that work across different recording paradigms.**

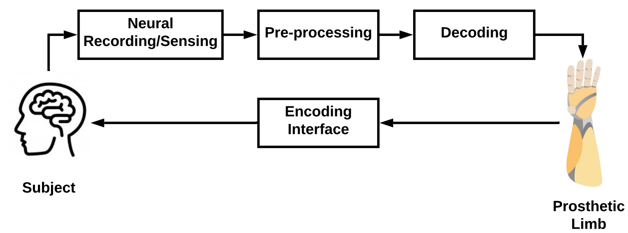

Fig. 1: A traditional setup for a brain-computer interface for controlling a prosthetic limb.

I. Introduction

The human brain is composed of billions of neurons, each of which communicate with one another via electro-chemical processes. The brain is responsible for nearly all functional behavior of the human body, for example, controlling the movement of limbs and extremities. Neuro-degenerative diseases and spinal cord injury can impede the brain's natural ability to communicate with the limbs and/or organs and hence impose severe limitations on the victims. Neuroscients are thus keen on studying the behavior of neurons, how their electro-chemical activities are correlated among groups of neurons or entire brain regions, and to explore how to form alternative neural pathways for brain signals. Brain-computer interfaces (BCI) are devices that can record and stimulate different parts of the brain. BCIs may then be used as alternatives pathways for neural signals to traverse, and thus may give enhanced mobility and range of movement for victims of neuro-degenerative diseases or spinal cord injury.

While there are various methods to record from the human brain, one of the most commonly use approach is electro-encephalography (EEG). EEG is a non-invasive recording method that involves placing recording electrodes on the scalp, each of which can record changes in voltage due to ionic currents in the brain [1]. EEG signals have been widely used to study different phenomena, such as epilepsy [2] and seizures [3]. The signals received by EEG recording electrodes are noisier than more invasive recording methods, such as electro-corticography (ECoG), which involves the placement of recording electrodes directly on the surface of the brain. However, EEG is widely popular for BCI systems due to their relatively quickly deployment and lack of surgical procedures. Another approach which does not require surgery is magneto-encephalography (MEG), which performs brain-imagery using the magnetic fields produced by ionic currents in the brain. MEGs, however, are expensive units that require significant power to operate.

One popular direction of BCIs is their ability to provide enhanced mobility and functionality of patients by granting users control of prosthetic limbs. Fig. 1 shows one possible configuration of a BCI for controlling a prosthetic limb. The movement of the prosthetic device is controlled by decoding motor imagery data from pre-processed neural recordings, in this case EEG. Motor imagery (MI) tasks has been a widely researched topic in recent years [4]–[6]. MI movement decoding involves processing of EEG waveforms of imaged movement of a particular limb or extremity, for example, the left arm, right arm, the tongue, or even individual fingers. The goal is to correctly classify the waveform as its respective MI movement. Before classification, several pre-processing steps are employed to prepare the data for subsequent classification. Filtering the data is commonplace as a pre-processing technique, usually around 10 - 100 Hz low-pass filtering. In Section II, we will go into more detail regarding the processing of EEG data.

Several classifiers have been used for MI decoding in recent years, including support vector machines (SVMs) [7]–[11], Neural networks, including multi-layer perceptrons and probabilistic neural networks [8], and matched filters [12], [13]. The classifier's tasks are to correctly label the EEG data (with or without preprocessing) as one of the MI tasks. SVMs have been widely used for pattern recognition and classification tasks. They perform transformations on the input features in order to distinguish them between the output classes. Neural networks (NNs) have been applied to various tasks, including pattern recognition and classification tasks, and they span a variety of network models and topologies. Multi-layer perceptrons, probabilistic neural networks, and other traditional feed-forward NNs, involve applying non-linear activation functions on the weighted inputs from previous layers. For classification tasks, the target outputs of the network are usually described with one-hot encoding (e.g., each class has a corresponding neuron in the output layer that should react the strongest to that class of input stimulus). Matched filters have also been applied to decoding MI tasks of EEG data. Matched filters function by correlating the EEG data with a known template (e.g., a template for each class of MI) to detect the presence of the template in the input EEG data (which is not yet classified/known).

While the previous approaches have shown accuracy ranging from the 70% to 90% range for classification of MI tasks of EEG data, these models do not inherently store temporal information. In order to conserve the temporal information of the data, and as an added side effect of significantly reducing the dimensionality of the model, we propose to utilize a recurrent neural network (RNN) of long short-term memory units (LSTM). RNNs use hidden layer units that may or may not apply similar non-linear activation functions to weighted inputs, however, the output value of the hidden layer(s) units depends on the current time step input to the hidden layer(s) as well as all previous outputs of the hidden unit, therefore representing a recurrent self-connection of the hidden layer(s). This means that the temporal information of the data is retained and also, allows the application of input stimuli in a temporal fashion.

We also utilize a SVM to compare the performance of the LSTM too when temporal concerns are ignored.

The rest of this article is organizes as follows: Section II discusses the dataset used for this article as well as any preprocessing applied to the data. Section III discusses the model for MI task decoding of EEG data as well as the relative accuracy/performance compared to previously published work. Finally, Section V makes some concluding remarks.

II. Dataset and Pre-processing

A. *Dataset details*

The EEG dataset we use consists of 60 hours of recordings across 13 participants [14]. At least 19 recording electrodes were used, sampled at either 200 Hz or 1 kHz. The recordings consist of 4 recording paradigms with movement: 5F, CLA, FREEFORM, and HaLT. The 5F paradigm has participants to perform different MI tasks, such as imagining the movement of the thumb and each of the fingers one at a time. The CLA, FREEFORM, and HaLT paradigms has participants perform left hand, right hand, left leg, right leg, and tongue MI tasks. A graphical user interface was used to instruct the participant which MI task to perform. The participants, denoted as SUBJECT A through SUBJECT M consisted of 8 males and 5 females, in between the 20 and 35 year age range. No subject details were stored or used for further processing.

The data is stored as .mat files and is well annotated, meaning that a marker indicating the performed MI task is supplied along with the EEG recordings themselves for the given MI task. The data files contain a structure, one of which is an array with 22 channels of EEG data as the columns and time as the rows. 19 of the channels are recording electrodes, 2 are ground electrodes, and 1 is for synchronizing what action is occurring to the data. The data is given as voltage, which is corresponding to the potentials due to the ionic currents, and the marker is an integer denoting the MI task. While no custom filtering was applied to the EEG signal, a band-pass filter of 0.53 - 70 Hz was present in all EEG recordings at 200 Hz. The 1 kHz data had a band-pass filter of 0.52 to 100 Hz. An additional 50 Hz notch filter was also applied to remove electrical grid interference.

TABLE I: The motor imagery (MI) tasks and their corresponding markers for the given paradigms.

Marker	Paradigm	MI Task
1	CLA HaLT FreeForm NoMT	Left Hand
2		Right Hand
3		Passive/Neutral
4		Left Leg
5		Tongue
6		Right Leg
91		Session Break
92		Experiment End
99		Initial Relaxation
1	5F	Thumb
2		Index Finger
3		Middle Finger
4		Ring Finger
5		Pinkie Finger

The markers given for defining the MI tasks are given in Table I. For constructing our data from the datasets provided by [14], we keep only the EEG data pertaining to meaningful MI markers, such as markers 1 - 6 for CLA, HaLT, and FreeForm paradigms, and markers 1 - 5 for the 5F paradigm.

B. *Pre-processing techniques*

One of the most commonly applied pre-processing techniques on EEG data is the method of common spatial patterns (CSP) [15]. The CSP process involves constructing spatial filters in such a way that the variances in the time series data is optimal for discriminating between classes. In the case of MI, this produces filters that maximizes the difference between different MI tasks. The linear filters are then used to apply a linear transform to the input signal, given by the following equation:

$$S_{\text{CSP}}(t) = \mathbb{W}^T \times s(t), \tag{1}$$

where $s(t)$ denotes a vector of EEG signals at time t from all channels, and W denotes the an $N \times M$ matrix that performs the transformation, whose columns form the spatial filters. For this article, we use a publicly available[1] implementation of the CSP algorithm to generate the spatial filters.

The difference between pre- and post-CSP waveforms can be measured by computing the sum of squared difference (SSD), as $\text{SSD} = \sum_t^{\tau} ({w_1}_t - {w_2}_t)^2$, where t denotes each time step of the EEG waveform from t, τ denotes the total duration of the waveform, w_1 denotes one EEG waveform, and w_2 denotes the other EEG waveform. For example, the SSD before and after applying CSP for two sample waveforms is 16.9e+3 and 31.7e+6, respectively. This shows that the difference between the waveforms is significantly higher, and therefore may be useful for a classifier to more easliy discriminate between the two waveforms.

[1]Python implementation of CSP available at: https://github.com/spolsley/common-spatial-patterns

C. Wavelet-BSS

Wavelet-BSS is a layered approach to prepocessing EEG data, applying multiple standard strategies sequentially to more strongly identify noise and remove it [16]. The data undergoes a series of transformations and dimension reductions to identify which channels are the noisiest, and which channels contain the most signal, shown in Fig. 2(a). Within the altered space of identified noise/signal, we apply nonlinear and/or linear standard cleaning techniques. Once cleaned, we invert the data back into the source space.

For our study, we chose to apply the Wavelet-BSS strategy, which first transforms the data using a mother wavelet of choice; either morlet, or meyer, to approximate the shape of the data within chunks of time domains. Then, we apply a BSS algorithm either through Principle Component Analysis (PCA), or Independent Component Analysis (ICA). ICA is typically preferred in EEG preprocessing due to its detection of components lowest gaussianality since greater gaussianality implies more noise, as opposed to PCA's detection of greatest variance [17]. Similar to PCA, ICA constructs a coefficient matrix W called the mixing matrix, reduced mixing matrix W_p constructed from the components of least gaussianality, and psuedo-inverse matrix W_p^{-1} to transform back to the source space. The purpose of transforming into the ICA space is to make noise more apparent for other filtering techniques, as well as remove any channels we think may contain too much noise or that do not contribute to signal detection, thus simultaneously removing noise, and dimensionality of our data. Filtering techniques within the ICA space conducted included iterations of outlier detection, and detrending using cross channel averages for every time point.

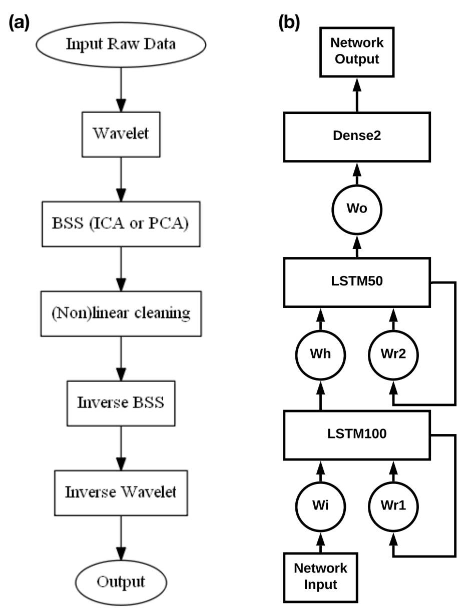

Fig. 2: (a) The wavelet-BSS flow and (b) the topology of the designed LSTM RNN.

Note that the ICA requires an orthogonal wavelet transformation because orthogonal wavelet transforms maintaining the variance-covariance matrix of the processed data. This makes it easier to apply PCA and ICA after the transform because they use either second order or higher order statistics reliant on the variance-covariance matrix. If we did not use an orthogonal wavelet, then transforming back from the ICA or PCA space may cause bithrogonality or other strange hiccups in the source space after we inverse transform everything.

III. Models

A. LSTM

One approach to perform MI decoding is to use neural network models that can inherently process time-series data, such as Recurrent Neural Networks (RNNs). While traditional feed-forward neural networks' outputs only rely on the input value, the output of the RNN relies on the current input value as well as previous values encoded in the hidden layer. We first develop an RNN for a two-class MI classification task consisting of two hidden layers using long short-term memory (LSTM) units, and one fully connected output layer of sigmoid units. Fig. 2(b) shows the topology of our designed LSTM RNN. The hidden layer consists of two recurrent layers, one with 100 LSTM units, and another with 50 LSTM units. The output layer consists of a dense (fully-connected) layer with 2 sigmoid units. Wi denotes the weights from the input to the first LSTM layer, Wr1 denotes the self-recurrent weights of the first LSTM layer, Wh denotes the weights between the first and second LSTM layers, Wr2 denotes the self-recurrent weights of the second LSTM layer, and Wo denotes the weights from the second LSTM layer to the output dense layer. The number of output units depends on the number of MI tasks to classify, i.e., each MI class has its own output unit.

The input to the network is given as a 21-element vector, which denotes one input per channel, and t inputs are given sequentially. The MI markers, which are the target outputs of the network, are one-hot encoded, and the output units employ the sigmoid activation function because their outputs will naturally be within the range of the target outputs.

B. SVM

In contrast, State Vector Machines (SVM) are a commonly used approach for MI classification. They use a kernel to transform the data to be easily divided via planes into regions. We used a linear kernel for speed increase at a performance cost.

The input to the SVM is a 2D matrix. Each row represents one MI action. It contains the voltage data for the entire duration of the action across 21 channels. Each row is organized like: [channel 1 data at start time, ..., channel 1 data at end time, channel 21 data at start time, ..., channel 21 data at end time, ...]. The MI markers are kept as integers.

IV. Results

We tested our model in two approaches: (i) testing a fine-tuned network for each individual session for each subject; (ii) combining all sessions from all subjects for a given paradigm. The idea approach (i) is based on the fact that EEG data is inherently noisy and shows a lot of variations

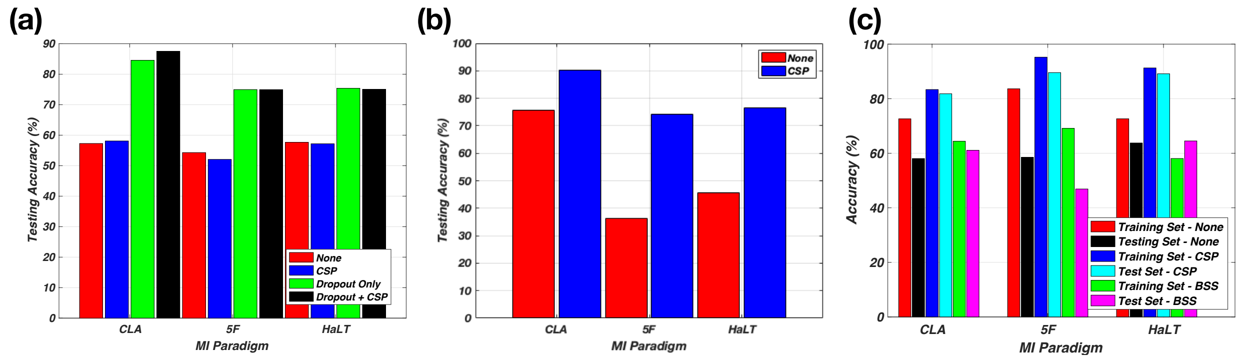

Fig. 3: The results for (a) the LSTM model using per-session training and test sets, (b) the results of the SVM using multi-subject data, and (c) the results of the LSTM model using multi-subject data and various pre-processing strategies.

for individual test subjects. Some of the problems with this approach is that the amount of data available for each subject and session is limited, which makes the problem of overfitting and generalization difficult to solve. This gives rise to approach (ii), in which significantly more data is used for training the model, and may reduce the overfitting problem and increase the models ability to provide a general solution.

The average testing accuracy results of approach (i) are shown in Fig. 3(a). We tested out the accuracy of the LSTM with three different modes: (i) no pre-processing, (ii) CSP pre-processing of the data, (iii) no pre-processing but including Dropout, and (iv) CSP pre-processing and Dropout. We can see that dropout consistently increases the performance of the LSTM, and in the case of the CLA paradigm, the CSP pre-processing further improves the testing accuracy. However, the training sets were small and the network consistently showed 95 to 99% accuracy on the test set, but significantly lower accuracy on the testing set. This shows that indeed there is an overfitting problem.

For comparison, we also show the results of approach (ii) for SVM in Fig. 3(b). Compared to the LSTM alone appears better for the CLA paradigm but worse for 5F and HaLT. In contrast to LSTM, CSP alone significantly increases the performance of SVM. Bringing the performance of SVM for all paradigms to be comparable to LSTM with Dropout and CSP combined. This includes the FREEFORM paradigm, where SVM with CSP achieved 64% accuracy. Following the results of approach (i), we then combine the data for approach (ii). The results of approach (ii), shown in Fig. 3(c), show that the testing accuracy on the CLA, 5F, and HaLT paradigms was 89%, 82%, and 89%, respectively. It is our intuition that access to significantly more training data improved the results considerably. Also, note that the overfitting was reduced drastically, with the average performance on the training set of 90% across the three paradigms, while the average performance of the testing set was 86%. Note that the FREEFORM paradigm, which is the self-paced version of the CLA paradigm was not used for training because it provided only three sessions worth of data. However, we were able to use the model trained on the CLA paradigm and evaluated the model using the FREEFORM data, which yielded 65% accuracy. This shows that the LSTM has generalized the EEG data, provided that the self-paced waveforms are different than those of the cue-based CLA paradigm. Note that these results are obtained after training each LSTM model with only 30 epochs.

V. Conclusion

In this article, we presented a recurrent neural network (RNN) model that is capable of classifying motor imagery (MI) electroencephalographic (EEG) waveforms reliably across three different recording paradigms. As opposed to previous approaches, which often rely on finely tuned classifiers, we show that training the RNN on a large dataset from various test subjects produces a robust classifier that is able to classify MI movements from different test subjects. The RNN is based on long short-term memory units, and achieves an average testing accuracy of 86 % across three different recording paradigms. In addition, it's performance is comparable to SVM with the benefit of keeping the temporal nature of data.

References

[1] E. Niedermeyer and F. L. da Silva, *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[2] T. N. Alotaiby, S. A. Alshebeili, T. Alshawi, I. Ahmad, and F. E. A. El-Samie, "Eeg seizure detection and prediction algorithms: a survey," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 183, 2014.

[3] U. R. Acharya, S. V. Sree, G. Swapna, R. J. Martis, and J. S. Suri, "Automated eeg analysis of epilepsy: a review," *Knowledge-Based Systems*, vol. 45, pp. 147–165, 2013.

[4] Y. Wang and S. Makeig, "Predicting intended movement direction using eeg from human posterior parietal cortex," in *International Conference on Foundations of Augmented Cognition*. Springer, 2009, pp. 437–446.

[5] E. López-Larraz, L. Montesano, Á. Gil-Agudo, and J. Minguez, "Continuous decoding of movement intention of upper limb self-initiated analytic movements from pre-movement eeg correlates," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, p. 153, 2014.

[6] A. Shakeel, M. S. Navid, M. N. Anwar, S. Mazhar, M. Jochumsen, and I. K. Niazi, "A review of techniques for detection of movement intention using movement-related cortical potentials," *Computational and mathematical methods in medicine*, vol. 2015, 2015.

[7] E. Yom-Tov and G. Inbar, "Detection of movement-related potentials from the electro-encephalogram for possible use in a brain-computer interface," *Medical and Biological Engineering and Computing*, vol. 41, no. 1, pp. 85–93, 2003.

[8] O. Bai, P. Lin, S. Vorbach, J. Li, S. Furlani, and M. Hallett, "Exploration of computational methods for classification of movement intention during human voluntary movement from single trial eeg," *Clinical Neurophysiology*, vol. 118, no. 12, pp. 2637–2655, 2007.

[9] A. T. Boye, U. Q. Kristiansen, M. Billinger, O. F. do Nascimento, and D. Farina, “Identification of movement-related cortical potentials with optimized spatial filtering and principal component analysis,” *Biomedical Signal Processing and Control*, vol. 3, no. 4, pp. 300–304, 2008.

[10] Y. X. Kato, T. Yonemura, K. Samejima, T. Maeda, and H. Ando, “Development of a bci master switch based on single-trial detection of contingent negative variation related potentials,” in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 4629–4632.

[11] M. Jochumsen, I. K. Niazi, N. Mrachacz-Kersting, D. Farina, and K. Dremstrup, “Detection and classification of movement-related cortical potentials associated with task force and speed,” *Journal of neural engineering*, vol. 10, no. 5, p. 056015, 2013.

[12] I. K. Niazi, N. Mrachacz-Kersting, N. Jiang, K. Dremstrup, and D. Farina, “Peripheral electrical stimulation triggered by self-paced detection of motor intention enhances motor evoked potentials,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 20, no. 4, pp. 595–604, 2012.

[13] I. K. Niazi, N. Jiang, M. Jochumsen, J. F. Nielsen, K. Dremstrup, and D. Farina, “Detection of movement-related cortical potentials based on subject-independent training,” *Medical & biological engineering & computing*, vol. 51, no. 5, pp. 507–512, 2013.

[14] M. Kaya, M. K. Binli, E. Ozbay, H. Yanar, and Y. Mishchenko, “A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces,” *Scientific data*, vol. 5, p. 180211, 2018.

[15] Y. Wang, P. Berg, and M. Scherg, “Common spatial subspace decomposition applied to analysis of brain responses under multiple task conditions: a simulation study,” *Clinical Neurophysiology*, vol. 110, no. 4, pp. 604–614, 1999.

[16] X. Jiang, G.-B. Bian, and Z. Tian, “Removal of artifacts from eeg signals: a review,” *Sensors*, vol. 19, no. 5, p. 987, 2019.

[17] C. Jutten and J. Karhunen, “Advances in blind source separation (bss) and independent component analysis (ica) for nonlinear mixtures,” *International journal of neural systems*, vol. 14, no. 05, pp. 267–292, 2004.