# EMG-based BCI for PiCar Mobilization

Efe Ertekin
*Computer Engineering Department*
*Mef University*
Istanbul, Türkiye
ertekinef@mef.edu.tr

Burak Bahir Günden
*Computer Engineering Department*
*Mef University*
Istanbul, Türkiye
gundenb@mef.edu.tr

Yasin Yılmaz
*Computer Engineering Department*
*Mef University*
Istanbul, Türkiye
yilmazy@mef.edu.tr

Alperen Sayar
*Tam Finans Ar-Ge*
*Tam Finans*
Istanbul, Türkiye
alperensayar@tamfinans.com.tr

Tuna Çakar
*Computer Engineering Department*
*Mef University*
Istanbul, Türkiye
cakart@mef.edu.tr

Şefik Şuayb Arslan
*Brain and Cognitive Sciences*
*Massachusetts Institute of Technology*
Massachusetts, United States
sarslan@mit.edu

2022 7th International Conference on Computer Science and Engineering (UBMK) | 978-1-6654-7010-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/UBMK55850.2022.9919502

***Abstract*—In this study, the main scope was to develop a brain-computer interface (BCI) with the use of PiCar and EEG/ERP devices. Thus, it is aimed to facilitate the lives of people with certain diseases and disabilities. The ultimate goal of this project has been to direct and control a BCI-based PiCar concerning the signals captured via the EEG/ERP device. With the EEG headset, the EMG signals of the gestures (facial expressions) of the participant were captured. With the collected data, filtering and other preprocessing methods were applied to have noise-free signals. In the preprocessing, the detrending method was used to clean the data set which showed a constantly increasing trend, to a certain range, and zero trends. The denoising (Wavelet Denoising) and outlier detection/elimination methods (OneClassSVM) were used for noise elimination. The SMOTE oversampling method was used for data augmentation. Welch's method was used to get band powers from the signals. With the use of augmented data, several machine learning algorithms were applied such as Support Vector Machine, Logistic Regression, Linear Discriminant Analysis, Random forest Classifier, Gradient Boosting Classifier, Multinomial Naive Bayes, Decision tree, K-Nearest Neighbor, and voting classifier. The developed models were used to predict the direction that is passed as an input to PiCar's API. After that, PiCar was controlled concerning the predicted direction with HTTP GET requests. In this project, the OpenBCI headset and the Brainflow library for EEG/EMG signal obtaining and processing were used. Also, the Tkinter library was used for the Graphical user interface and Django for establishing a server on PiCar's brain which is RaspberryPi.**

***Keywords* —EEG, EMG, Brain Wave Signals, Brain Computer Interface, Machine Learning, PiCar***

## I. Introduction

In recent years, the applications in the artificial intelligence domain have progressed rapidly, and now the developed applications are relatively more capable of creating solutions to real-life problems, impacting everyone's life in many diverse ways. The successful applications of AI to Electroencephalography (EEG), Magnetoencephalography (MEG), and Electromyography (EMG) data create a control pathway of the brain/muscle activities to manage the devices around us, leading to Brain-Computer Interface (BCI) designs. The BCI-enabled devices utilize the electrical activity of the human brain to control wheelchairs[1], drones [2], or robotic arms [3]. The motivation behind this study is to come up with a BCI solution for people who have disabilities and/or diseases. In this project, instant predictions are made by detecting the previously determined facial expressions. To detect facial expressions (EMG-based), an EEG headset is used due to the fact that it can also detect muscle movement in the face. Each of these gestures belongs to one of the 5 directions. These directions are given as forward, right, left, back, and stop. These directions are sent to a drone car named PiCar with an HTTP GET request. Django application installed in PiCar brings mobility to the vehicle according to the direction commands provided. To achieve this, the motors attached to the wheels are activated. A picture of the PiCar is provided in Fig. 1.

The proposed design shall have a plethora of applications ranging from gaming to medical care. The primary audience of the design would be individuals with neural malfunctioning and those with physical disabilities. Additionally, it could be designed to pave the way for those who do not have any known medical conditions and yet would like to have an extra control device. In other words, it will be possible to establish a BCI-based system almost as if an extra limb has been added to an individual.

As initially mentioned, this project aims to improve brain-controlled prostheses by combining electroencephalography

(EEG) signals using facial expressions. The motivation of the project is to improve the efficiency and performance of brain-computer interfaces. There are two main paradigms in the related domain such as BCI motor-imagery based (MI-BCI) and steady-state visual evoked potential (SSVEP-BCI). In MI-BCI, future research may be restricted by the length of its training and the diversity of its users. The accuracy of MI-BCI is nearly 90% after several months. In SSVEP-BCI, stimulators are used. The most common stimulator is a flashing light pattern. A prolonged stimulation period can easily result in epileptic seizures. Moreover, the accuracy of SSVEP-BCI is 85%.

The control scheme of myoelectric prosthesis has a part called pattern recognition. The main duty of pattern recognition is decoding input data and sending outputs to the myoelectric prosthesis for making the intended gesture. Pattern recognition has a classifier. The classifier is generally calibrated and trained with a supervised learning framework. Since the aim of the project has to develop a better framework for the classifiers, the three-phase identification framework is developed. According to obtained bio-signals (EMG - EEG), the framework is capable of self-learning. The self-learning framework has higher accuracy in learning patterns and it reduces the lag-time than the supervised learning framework.

In an empirical study in the related academic literature, a system was proposed to control prosthetic fingers with the adoption of BCI to help people disabled people suffering from motor mobility impairment. The study focused on the multi-class classification problem. The previous EEG studies extracted from the same area of the brain(same electrodes) for finger movements at a hand. But for the same motor imagery task, the activity of the brain in the left and right hemispheres is different. So the study states that all electrodes need to be considered. This led the researchers to use a model (which is statistical) to identify the imagery task of each finger for each subject. During the study, they focused on five motor imagery tasks which are pinky finger, ring finger, middle finger, index finger, and the thumb. For the experiments a g.HIamp from g.tec, an 80-channel amplifier was used for the creation of the

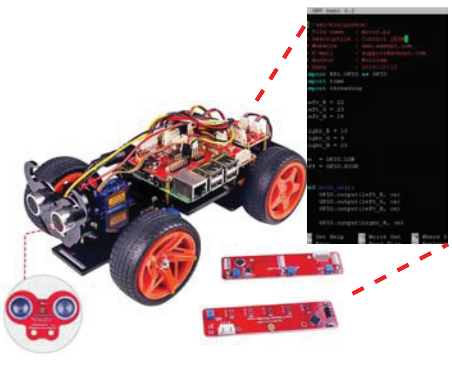

Fig. 1. PiCar is a drone car built on the Raspberry Pi platform which is hooked up with our EEG equipment.

data set. The data was processed by removing artifacts; it helped to increase the signal-to-noise ratio of the EEG signals [4]. For feature extraction of the finger movements, the Constraint Satisfaction Problem (CSP) algorithm was applied. For the experiment, they developed an SPI to guide subjects in each session.

After that, a filter block was used to eliminate artifacts. To allow subtraction of signals in each electrode for each time point from the average signal and allow calculation of average signal at all electrodes, a technique called the common average referencing technique was applied. This technique helps to discriminate between positive and negative peaks in the EEG signals [4]. It is known that for the selection of relevant electrodes, previous studies extract features from a predefined set of electrodes to detect the movements. For feature extraction, CSP spatial filters are used, and for the classification, the Linear Discriminant Analysis (LDA) algorithm is used.

The rest of the paper is organized as follows. In Section II, all the equipment and software required for the development of the system and all the mathematical formulas and tools required for their optimization will be given. After these are detailed, the results obtained with different algorithms will be shared and discussed in Section III. Finally, we conclude the paper in Section IV with a few potential directions.

## II. Materials and Methods

Multiple software and hardware combinations were used in this project. In addition to the software used, different functions and algorithms were utilized to pre-process the collected data and prepare it for further inference. We start with the hardware components of the system in the next two subsections. The rest of the subsections provide the details of the programming languages, libraries, algorithms, and machine learning models used.

### A. OpenBCI Headset

The headset with the electrodes is the main entity that enables the instantaneous collection of EEG and EMG signals. A cap standard with 16 channels with a sampling rate of 125 Hz was used in our experiments. The same cap also has models with different numbers of channels and sampling rates. This data collection device, which was obtained by combining two motherboards named Cyton + Daisy, works with 4 AA batteries.

### B. PiCar

In our study, we utilize a toy known as PiCar, which is a drone car built on the Raspberry Pi platform. There is an additional Django application shared by the manufacturer for web-based control/access and management. Thus, by making changes to this application, movements in the desired direction can be achieved.

### C. Python Libraries

The data was collected using already available python libraries optimized for use for these systems. The most popular of these libraries, namely the MNE and the BrainFlow, were both used in this project. The MNE is a software package that facilitates the examination and analysis of brain signals [5]. First of all, the analysis was carried out using the BrainFlow framework, and similar steps were carried out with the MNE to observe the consistency of the data collected using both software. Thus, we have determined whether the collected data is corrupted or not, which may arise from the use of different Python libraries. In addition to these, the library named Tkinter was used in the development of the interface required to collect the data. Basic libraries such as Pandas and NumPy are also used to efficiently work with arrays and help develop efficient processing of the collected data.

### D. Data Collection

For the data collection part, we developed an interface using Python Tkinter and BrainFlow libraries and used this to perform data collection instantly (online data streaming). With this interface, you can choose how much data block you want to collect. Thus, you can choose how many minutes the user will stay in a data collection session. To achieve this, GIF animations are displayed on the screen in 5 different directions (forward, right, down, left, and middle). These GIF animations consist of small videos that show you how to move your face/gaze. There is a 0.5-second break in transitions of each direction display. The GIF animations are played on the screen for 5 seconds. A block is completed by displaying and collecting data for all directions. This equates to approximately a total of 30 seconds of experimentation time.

During the data collection phase, it was necessary to ensure that the data passed through the filters accurately. Different methods were proposed for this. First of all, with detrending, the uptrend in the data was eliminated, so it became more logical to comment on the signals. After that, it was filtered by the 50Hz band-stop filter to remove the jitters caused by the city electricity. Following this phase, the range of 1Hz-46Hz was extracted with an IIR Chebyshev-type 1 band-pass filter. The last remaining data went through Wavelet-based denoising processes. Thus, we have cleared the data within a certain range as described. More specifically, we have applied the following methods to the collected data.

*1) Detrending:* Since it is difficult to operate on data that rises regularly, with detrend, the rise (trend) of this data is balanced and the operations are performed on this data[6].

*2) Band-stop Filtering:* A Band-stop filter is a filter that passes most frequencies unaltered but attenuates those in a specific range to very low levels. The standard frequency in Europe's electricity grid (50Hz) is filtered out.

*3) Band-pass Filtering:* With the band-pass filter, incoming frequencies in a certain range are allowed to pass. Frequencies outside this range are not allowed and the data is placed in a range [7].

*4) Wavelet Denoising:* The wavelet transform leads to a sparse representation of many real-world signals and images, which is the core notion behind wavelet denoising, or wavelet thresholding. The wavelet transform focuses signal and image characteristics into a small number of large-magnitude wavelet coefficients [8]. Small wavelet coefficients are often noisy, and you can "shrink" or delete them without harming the signal or image quality. After thresholding, the coefficients was used the inverse wavelet transform to rebuild the data. Now data is denoised after the inverse transformation.

### E. Machine Learning

Different machine learning techniques and algorithms are utilized. The main purpose of using them is to organize the obtained data and to provide more successful results during the model development.

*1) Feature Engineering:* Frequency-based information such as harmonics was obtained by segmenting the frequency spectrum of the collected raw data. This methodology mostly resembles the known filter-based approaches used in the past for target detection in SSVEP-based BCI systems. Eight different frequencies were formed for further analysis: Alpha low (7 - 10 Hz), alpha high (10 - 13 Hz), beta low (14 - 22 Hz), beta high (22 - 30 Hz), gamma low ( 31 - 38) Hz), gamma high (38 - 45 Hz), theta low (4 - 5.5 Hz), and theta high (5.5 - 7 Hz) bands [9]. Note that the main difference with respect to previous filter banks of SSVEP is that we use narrowband bandpass filtering for Brainwaves.

*2) Outlier Detection and Elimination:* Outlier detection was performed on 8 different data types obtained using SVMOneClass with the following parameters: (kernel='rbf', nu=0.2). "nu" is the bound of the fraction of support vectors for lower and fraction of training errors for upper. Outliers detected here were excluded from the data [10].

*3) Oversampling:* Oversampling was performed with SMOTE [11]. After the outlier detection and elimination process was applied to the data, the data set became unbalanced. For example, while we have the same number of data for each direction, we now have less "left" because more outliers are detected in those going left. The oversampling method was used to balance the data set. This helped us to obtain more accurate and logical results during the model development.

*4) Algorithms:* Support Vector Machine, Logistic Regression, Linear Discriminant Analysis, Random forest Classifier, Gradient Boosting Classifier, Multinomial Naive Bayes, Decision tree, K-Nearest Neighbor, and voting classifier algorithms were tested. The results for each are presented in the next section.

### F. Fine-Tuning

Sklearn's GridSearch method is used to optimize the model parameters and perform fine-tuning of the hyper-parameters. There are different parameters of the models to be trained in which the parameter space is discretized. With this method, it is possible to give certain limits to these values and find the best result within those limits by running the local exhaustive search. This search, which explores the given parameter space, uses a lot of computational/memory resources and can take long time to find the optimal solution.

TABLE I
SUPPORT VECTOR MACHINES (SVM) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 0.89 | 0.76 | 0.82 | 33 |
| **Right** | 0.90 | 0.82 | 0.86 | 34 |
| **Backward** | 0.96 | 0.74 | 0.84 | 35 |
| **Left** | 0.77 | 0.71 | 0.74 | 34 |
| **Stop** | 0.60 | 0.94 | 0.73 | 33 |
| **Accuracy** | | | 0.79 | 169 |
| **Macro Avg.** | 0.83 | 0.79 | 0.80 | 169 |
| **Weighted Avg.** | 0.83 | 0.79 | 0.80 | 169 |

TABLE II
LOGISTIC REGRESSION (LR) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 1 | 0.91 | 0.95 | 33 |
| **Right** | 0.97 | 0.91 | 0.94 | 34 |
| **Backward** | 0.94 | 0.89 | 0.91 | 35 |
| **Left** | 0.81 | 0.88 | 0.85 | 34 |
| **Stop** | 0.73 | 0.82 | 0.77 | 33 |
| **Accuracy** | | | 0.88 | 169 |
| **Macro Avg.** | 0.89 | 0.88 | 0.88 | 169 |
| **Weighted Avg.** | 0.89 | 0.88 | 0.88 | 169 |

## III. RESULTS & DISCUSSION

In this section, we provide the accuracies of various ML techniques on the data we have collected. We also report precision, recall, and F1-score values for a better comparison. Our results are tabulated and presented in Tables I-IX for each technique used. We have observed an average accuracy of 80%-88% for models that are trained during the progression timeline of the study. However, the acquired findings indicated that the most balanced results were obtained using Logistic Regression. We concluded that Logistic Regression creates a more suitable model for this system. More specifically the system has reached the level of weighted accuracy up to 88% with the Logistic Regression model. Although there are other algorithms with similar accuracy and performance ratios, our observation during our tests demonstrated that it works quite well. For instance, the KNN model development also has performed relatively fine and it may be possible to use it while developing this system. It may also be possible to achieve higher results with the use of a voting classifier. We also note that through numerical evaluations it is possible to achieve higher levels of performance by using a longer duration of training sessions in which the participants will be enabled to do the given tasks several times until they can provide a clean data set. Although such multi-trial sessions will make the experiments more tiring, they may be required for a strong training of the algorithms.

Evaluation of the techniques is not limited to our computer simulations. We have also created an online processing system through transferring the classification and control signals obtained to the PiCar instantly, we showed that a wheelchair with the necessary equipment is able to move and execute the control commands with this system. Thanks to PiCar mobilization, it has been possible to realize such as system in an online setting and have the system come to life with real world experience.

TABLE III
LINEAR DISCRIMINANT ANALYSIS (LDA) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 0.77 | 0.52 | 0.62 | 33 |
| **Right** | 0.67 | 0.85 | 0.75 | 34 |
| **Backward** | 0.79 | 0.63 | 0.70 | 35 |
| **Left** | 0.60 | 0.53 | 0.56 | 34 |
| **Stop** | 0.63 | 0.88 | 0.73 | 33 |
| **Accuracy** | | | 0.68 | 169 |
| **Macro Avg.** | 0.69 | 0.68 | 0.67 | 169 |
| **Weighted Avg.** | 0.69 | 0.68 | 0.67 | 169 |

TABLE IV
RANDOM FOREST CLASSIFIER (RFC) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 0.87 | 0.82 | 0.84 | 33 |
| **Right** | 0.83 | 0.85 | 0.84 | 34 |
| **Backward** | 0.96 | 0.71 | 0.82 | 35 |
| **Left** | 0.76 | 0.65 | 0.70 | 34 |
| **Stop** | 0.62 | 0.91 | 0.74 | 33 |
| **Accuracy** | | | 0.79 | 169 |
| **Macro Avg.** | 0.81 | 0.79 | 0.79 | 169 |
| **Weighted Avg.** | 0.81 | 0.79 | 0.79 | 169 |

TABLE V
GRADIENT BOOSTING CLASSIFIER (GBC) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 0.97 | 0.88 | 0.92 | 33 |
| **Right** | 0.91 | 0.91 | 0.91 | 34 |
| **Backward** | 1 | 0.63 | 0.77 | 35 |
| **Left** | 0.72 | 0.76 | 0.74 | 34 |
| **Stop** | 0.62 | 0.88 | 0.73 | 33 |
| **Accuracy** | | | 0.81 | 169 |
| **Macro Avg.** | 0.84 | 0.81 | 0.81 | 169 |
| **Weighted Avg.** | 0.85 | 0.81 | 0.81 | 169 |

TABLE VI
MULTINOMIAL NAIVE BAYES CLASSIFIER (MNB) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 1 | 0.88 | 0.94 | 33 |
| **Right** | 0.97 | 0.91 | 0.94 | 34 |
| **Backward** | 0.96 | 0.77 | 0.86 | 35 |
| **Left** | 0.84 | 0.76 | 0.80 | 34 |
| **Stop** | 0.61 | 0.91 | 0.73 | 33 |
| **Accuracy** | | | 0.85 | 169 |
| **Macro Avg.** | 0.88 | 0.85 | 0.85 | 169 |
| **Weighted Avg.** | 0.88 | 0.85 | 0.85 | 169 |

The very promise and main objective of this project were to develop a basic control system to support disadvantaged people with limited physical abilities and help them live more comfortably. The idea is to address some of the problems that they are facing every day with plausible solutions. In addition, note that the same idea can be utilized for any healthy individual to provide additional abilities such as having an extra arm or a leg. On the other hand, we would like to highlight the required training for the proposed system. Just like it takes time to excel in driving a car through experience with the

TABLE VII
DECISION TREE CLASSIFIER (DTC) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 0.96 | 0.79 | 0.87 | 33 |
| **Right** | 0.89 | 0.94 | 0.91 | 34 |
| **Backward** | 0.86 | 0.69 | 0.76 | 35 |
| **Left** | 0.76 | 0.82 | 0.79 | 34 |
| **Stop** | 0.68 | 0.85 | 0.76 | 33 |
| **Accuracy** | | | 0.82 | 169 |
| **Macro Avg.** | 0.83 | 0.82 | 0.82 | 169 |
| **Weighted Avg.** | 0.83 | 0.82 | 0.82 | 169 |

TABLE VIII
K-NEAREST NEIGHBORS (KNN) TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 0.97 | 0.88 | 0.92 | 33 |
| **Right** | 0.91 | 0.94 | 0.93 | 34 |
| **Backward** | 0.96 | 0.74 | 0.84 | 35 |
| **Left** | 0.88 | 0.88 | 0.88 | 34 |
| **Stop** | 0.70 | 0.91 | 0.79 | 33 |
| **Accuracy** | | | 0.87 | 169 |
| **Macro Avg.** | 0.88 | 0.87 | 0.87 | 169 |
| **Weighted Avg.** | 0.89 | 0.87 | 0.87 | 169 |

TABLE IX
VOTING CLASSIFIER TEST PREDICTION RESULTS

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Forward** | 0.94 | 0.88 | 0.91 | 33 |
| **Right** | 0.97 | 0.91 | 0.94 | 34 |
| **Backward** | 0.96 | 0.77 | 0.86 | 35 |
| **Left** | 0.87 | 0.79 | 0.83 | 34 |
| **Stop** | 0.64 | 0.91 | 0.75 | 33 |
| **Accuracy** | | | 0.85 | 169 |
| **Macro Avg.** | 0.88 | 0.85 | 0.86 | 169 |
| **Weighted Avg.** | 0.88 | 0.85 | 0.86 | 169 |

whole driving process, it is clear that the proposed control and the accuracy of the system can be further improved through establishing experience with a higher level of practice.

## IV. CONCLUSION

A genuine control mechanism is proposed to detect different facial expressions using the OpenBCI system and enable possible use cases for different purposes such as supplying commands to a robotic manipulator for enhanced management. In our experimental evaluations using different machine learning techniques with hyper-parameter optimizations, the trained Logistic Regression model secured the best results with the top 88% accuracy. Based on our preliminary results, with accuracies approaching 90%, the proposed system can be used to enhance the lives of people in need adequately well. Moreover, as mentioned in the main text, the performance of such BCI systems might be improved with more data and experience. Further improvements can be obtained through fine-tuning for subject-specific experiments. The current team will be working on such a model that could be fed with higher amounts of data, investigate the effect of experience and eventually be able to learn from the incorrect classification outputs.

## REFERENCES


[1] H. Yousif, N. Abdul Rahim, M. Juhairi, A. Safar, and A. Zakaria, "An intelligent emg controlled wheelchair – a review," *International Academy of Engineering and Medical Research, 2016*, 01 2016.

[2] M. McCoggle, S. Wilson, A. Rivera, R. Alba-Flores, and V. Soloiu, "Biosensors based controller for small unmanned aerial vehicle navigation," in *SoutheastCon 2022*, 2022, pp. 380–384.

[3] H. Shedeed, M. Issa, and S. El-Sayed, "Brain eeg signal processing for controlling a robotic arm," 11 2013, pp. 152–157.

[4] S. Gannouni, K. Belwafi, H. Aboalsamh, B. Alebdi, Y. Almassad, Z. AlSamhan, and H. Alobaedallah, "Eeg-based bci system to control prosthesis's finger movements," 07 2020.

[5] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "Mne software for processing meg and eeg data," *NeuroImage*, vol. 86, pp. 446–460, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811913010501

[6] A. Cheveigné and D. Arzounian, "Robust detrending, rereferencing, outlier detection, and inpainting for multichannel data," *NeuroImage*, vol. 172, 02 2018.

[7] P. Chum, S. Park, K. Ko, and K.-B. Sim, "Orthonormal polynomial based optimal eeg feature extraction for motor imagery brain-computer interface," *Journal of Korean Institute of Intelligent Systems*, vol. 22, 12 2012.

[8] N. Sobahi, "Denoising of emg signals based on wavelet transform," *Asian Transactions on Engineering*, vol. 1, 01 2011.

[9] J.-S. Lin and C.-H. Shieh, "An ssvep-based bci system and its applications," *International Journal of Advanced Computer Science and Applications*, vol. 5, 10 2014.

[10] S. Saba-Sadiya, E. Chantland, T. Alhanai, T. Liu, and M. Ghassemi, "Unsupervised eeg artifact detection and correction," *Frontiers in Digital Health*, vol. 2, p. 608920, 01 2021.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, p. 321–357, jun 2002.