

Received April 25, 2020, accepted May 15, 2020, date of publication May 26, 2020, date of current version June 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997681

# A Brute-Force CNN Model Selection for Accurate Classification of Sensorimotor Rhythms in BCIs

**BERDAKH ABIBULLAEV<sup>1</sup>**, (Member, IEEE), **IRINA DOLZHIKOVA<sup>2</sup>**, (Member, IEEE),  
**AND AMIN ZOLLANVARI<sup>2</sup>**, (Senior Member, IEEE)

<sup>1</sup>Department of Robotics and Mechatronics, School of Engineering and Digital Sciences, Nazarbayev University, 010000 Nur-Sultan, Kazakhstan

<sup>2</sup>Department of Electrical and Computer Engineering, School of Engineering and Digital Sciences, Nazarbayev University, 010000 Nur-Sultan, Kazakhstan

Corresponding author: Amin Zollanvari (amin.zollanvari@nu.edu.kz)

This work was supported in part by the Nazarbayev University Faculty Development Competitive Research Grant under Award SOE2018008.

**ABSTRACT** The ultimate goal of Brain-Computer Interface (BCI) research is to enable individuals to interact with their environment by translating their mental imagery. In this regard, a salient issue is the identification of brain activity patterns that can be used to classify intention. Using Electroencephalographic (EEG) signals as archetypical, this classification problem generally possesses two stages: (i) extracting features from collected EEG waveforms; and (ii) constructing a classifier using extracted features. With the advent of deep learning, however, the former stage is generally absorbed into the latter. Nevertheless, the burden has now shifted from trying a number of feature extraction methods to tuning a large number of hyperparameters and architectures. Among existing deep learning architectures used in BCI, Convolutional Neural Networks (CNN) have become an attractive choice. Most of the existing studies that use these networks are based on well-known architectures such as AlexNet or ResNet, use the domain knowledge to construct the final architecture or have an unclear strategy deployed for model selection. This raises the question as to whether constructing accurate CNN-based classifiers is possible using a principled model selection, with the most straightforward one being the brute-force search or, alternatively, experience and developing high intuition regarding hyperparameters combined with an *ad hoc* approach is the most prudent way to go about designing them. To this end, in this paper, we first define a space of hyperparameters restricted by our computing power. Then we show that an exhaustive search within this limited space of CNN hyperparameters leads to accurate classification of sensorimotor rhythms that arise during motor imagery tasks.

**INDEX TERMS** Brain-computer interfaces, motor imagery, deep learning, convolutional neural network, model selection, brute-force search, sensory-motor rhythms.

## I. INTRODUCTION

Brain-Computer Interface (BCI) research aims to provide alternative channels for communication and control without involving any peripheral nervous system [1]–[3]. This technology is important for people who are affected by motor disabilities such as stroke, paraplegia, and amyotrophic lateral sclerosis [4], [5]. It has already been employed to control external devices such as robotic prostheses/orthoses [6], and exoskeletons for stroke rehabilitation studies [7], [8].

The design of a high-performance BCI system is still an open research problem in the field and requires accurate

The associate editor coordinating the review of this manuscript and approving it for publication was Larbi Bouchir.

decoding of the EEG signals generated by neuro-electrical activities in the brain [9]–[11]. EEG-based BCI systems are classified as exogenous and endogenous, where the former requires an external stimulus to excite specific responses in the brain [12], [13]. Depending on the type of stimulation, the exogenous BCIs use steady-state visual evoked potentials (SSVEPs), or event-related potentials (ERPs), brain signals elicited in response to cognitive or sensory events. The advantages of exogenous BCIs are related to the high information transmission rate with little user training requirements [13]. However, an exogenous BCI system constraints the user to focus on the visual stimuli and, as a result, its usefulness may be limited, especially for severely motor-impaired people [14]. This paper focuses on the endogenous category of

BCI systems, which utilize sensory-motor rhythms (SMRs) for control of external devices, independent of any stimuli. The SMRs represent the modulations of oscillatory activity in EEG induced by motor imagery of limb movement as input features [15], [16].

From a machine learning perspective, there is already a variety of techniques for feature extraction, feature selection, and classification for SMR-based BCI systems that have been developed over the last decade [11], [17]. However, recent BCI studies exhibit a trend of moving towards deep learning methods as they have already shown the state-of-the-art performance in learning from raw EEG data, mitigating the need for laborious preprocessing or extracting handcrafted features [18].

Convolutional neural networks (CNNs) have emerged as powerful deep learning architectures for decoding complex event-related desynchronization and synchronization (ERD/ERS) patterns elicited in SMRs in various BCI applications [18], [19]. Several studies have already demonstrated the efficiency of CNNs in terms of training time (as compared to some of the popular recurrent neural networks) and effectiveness to capture latent features from raw EEG data [20]. For instance, Schirrmester *et al.* [21] evaluated several design choices to develop CNN-based architecture for interpreting the imagined and executed tasks using raw EEG data. Uktveris and Jusas [22] tested 11 different CNN architectures and selected the optimal CNN parameters for four-class motor imagery classification. Sakhavi *et al.* [23] investigated three different convolution scenarios: convolution across time (Channel-wise CNN), convolution across channels (Channel-mixing CNN), and convolution across both time and channels with a two-dimensional kernel (2-D CNN), and demonstrated the effectiveness of CNNs in SMR classification. Dai *et al.* [24] proposed a hybrid-scale CNN architecture that incorporates a data augmentation technique and achieved accurate classification in two different motor imagery datasets. In addition to purely CNN-based architectures, hybrid models incorporating CNN and other deep structures have been used in the context of SMR-based BCIs. For instance, studies in [25]–[27] investigated hybrid CNN and stacked autoencoders (AE) architectures for EEG classification. Likewise, constructing a hybrid CNN and gated recurrent units [28], and CNN with the long-term, short-term memory networks [29] have been studied.

The studies mentioned above exhibit the state-of-the-art performance of deep learning models in learning from raw data, alleviating the need for manual EEG feature engineering. Nevertheless, some studies still explore various EEG data transformation techniques that could enhance deep learning models. For instance, Chaudhary *et al.* [30] applied continuous wavelet transforms (CWT) and short-time-Fourier-transform to EEG data, used the outcome of these transformations as inputs to an AlexNet-based CNN classifier and showed that CWT features yield better accuracy [31]. A similar study [32] applied blind source

separation and CWT to preprocess EEG as input to a CNN classifier [33] and studied effects of adjustments in the convolutional stride and max-pooling size to improve classification accuracy. Further research attempted to identify the best feature extraction, and deep learning-based classification pipeline for the SMR based BCIs [11], [18], [34]. However, a recent study by Lawhern *et al.* proposed a CNN-based architecture called EEGNet, which incorporates a feature extractor and a classifier in a CNN model [35]. Mainly, the EEGNet was inspired by the Filter Bank Common Spatial Patterns (FBCSP) — a celebrated spatial feature extraction method [36], [37]. A comparative analysis between EEGNet and other approaches demonstrated that with a much lower number of parameters, EEGNet could perform as good as or even better than several models in terms of generalization error across various BCI paradigms [35]. Other studies inspired by the FBCSP/CSP methods are [23], [38].

In this study, we are concerned with standard convolutional neural networks. Table 1 summarizes the studies above from the perspective of the hyperparameter selection strategy. In this regard, we distinguish structural hyperparameters (number of layers, filters, kernel size, and the choice of activation functions) from algorithmic hyperparameters (batch size, dropout rate, and signal segmentation and the choice of optimizer and its learning rate).

As seen in Table 1, most of the studies that use the neural networks are based on well-known architectures such as AlexNet or ResNet, use the domain knowledge to construct the final architecture or have an unclear strategy deployed for model selection. This state of affairs can be attributed to a relatively large number and wide range of hyperparameters involved, which together render a principled model selection difficult. This raises the question of whether a principled model selection, such as the brute-force search within a limited space of hyperparameters, could lead to architectures that can accurately classify motor imagery tasks based on collected EEG data. To examine this question, in this investigation, we conduct the first large-scale analysis in which we use EEG waveforms obtained from a number of subjects across different datasets to determine a single architecture via a brute-force model selection. We then examine the classification accuracy of the selected model on a set of subjects from an independent dataset that was unseen during the architecture selection phase.

This work proceeds as follows. Section II-A provides the description of (i) publicly available datasets used in our study; (ii) the preprocessing steps applied to our datasets. Section III presents the CNN architectures that we use as part of our brute-force model selection procedure. In Section IV, we discuss the CNN training and model selection procedures, and present the results of applying constructed models in Section V to independent test data. The key findings and observations are summarized in Section VI and, Section VII provides the brief conclusion of the work.

**TABLE 1. A summary of SMR based BCI studies that use convolutional neural networks and corresponding strategies for hyperparameter selection.**

Study	Structural Hyperparameters	Algorithmic Hyperparameters
[21]	(1) <i>Deep CNN model</i> inspired by architecture of AlexNet [31], 4 convolution-max-pooling blocks + dense softmax classification layer (2) <i>Shallow CNN architecture</i> is inspired by FBCSP, 1 convolution-max-pooling block + dense softmax classification layer (3) <i>hybrid CNN</i> - fusion of deep and shallow CNN after the final layer; (4) <i>Residual CNN</i> (for raw EEG signals) is based on ResNet [39]	Adam optimizer + early stopping method, Dropout + batch normalization
[22]	11 different CNN architectures were tested, finally selected architecture: Convolutional layer (4×4, 16 filters) + RELU layer + max-pooling layer (2×2, stride 2) + fully connected layer + softmax layer	Optimal values of learning rate, batch size, etc. were found via parameter range scanning approach, Initial learning rate=0.01, Batch size=128
[35]	inspired by FBCSP, three types of convolutional layers are used : conventional convolution, depthwise convolution and separable convolution [40]	Adam optimizer with default parameters, Dropout rate: 0.5 (within subject classification), 0.25 (cross-subject classification)
[23]	inspired by FBCSP, size of the kernel and number of convolutional nodes are selected via cross-validation, Channel-wise CNN: 2 convolutional layers+ flatterring+fully connected network, Channel-wise convolution with Channel Mixing: 3 convolutional layers+ flatterring+fully connected network	Adam optimizer with default parameters, dropout probability = 50%
[38]	based on ResNet [39], employs CSP, 1-D CNN	Nadam optimizer with default parameter, Batch size =50, Initial learning rate = 0.001, learning rate decay factor = 0.7
[30]	based on pre-trained AlexNet [31], 5 convolutional layers (+ pooling layers) + 3 fully connected layers	Learning rate = 0.001
[24]	2 convolutional layers + max pooling layer + flatten layer (2D or 1D) + 2 fully connected layers, impact of kernel size across the tested subject was investigated	Stochastic gradient descent optimization method, Dropout probability = 0.8, L2 regularization parameter = 0.01, Initial learning rate = 0.1, learning rate decay is 0.9
[33]	inspired by architecture of LeNet [41], Convolution-2D layer (200 filters of size 3×1) + max pooling- 2D layer (4×4 pool size) + convolution-2D (200 filters of size 2×2) + max pooling- 2D layer (4×4 pool size)+ flattening + 2 fully connected layers	Stochastic gradient descent optimizer, Learning rate = 0.001, L2 regularization parameter = 0.004
[32]	initial design was based on [33] (which in turn was based on LeNet [41]), Convolution-2D layer + max pooling- (4×4 pool size) + convolution-2D + max pooling (3×3 pool size)+ flattening + 2 fully connected layers , analysis of the kernel size was conducted	Dropout =0.4

## II. MATERIALS

### A. DATASET DESCRIPTION

In training and choosing the structure of our constructed CNNs, we have used publicly available EEG data from different sources corresponding to motor-imagery tasks recorded from a number of subjects. In this study, we focus on the decoding of the limb imagery data corresponding to left-hand and right-hand imagery and select data corresponding to those imagery tasks accordingly. Further, we segmented the continuous EEG into a left-hand and right-hand imagination trials of 4–seconds length after the mental imagery onset in all datasets. The detailed description of each dataset is provided below.

#### 1) Weibo2014

This dataset consists of EEG data recorded from ten healthy right-handed subjects (seven females and three males, 23–25 years old). The 64-channel data were acquired using a Neuroscan SynAmps2 amplifier at a sampling rate of 1000 Hz. The data were further band-pass filtered to the range of 0.5–50 Hz and down-sampled to 200 Hz for the subsequent analysis. The electrodes were placed according to the international 10-20 system referenced to the nose and grounded prefrontal lobe. The original study investigated the differences in EEG patterns within simple limb imagery and compound limb motor imagery. A single trial lasted for eight seconds. After two seconds, visual cue in the form of a red circle was shown to participants for a second; subsequently, a cue indicating left-hand or right-hand imagery was shown during which the participants performed kinesthetic motor imagery for about four seconds. The entire experiment

lasted for nine sessions, wherein each session, 60 trials of motor imagery data were collected for each task (for more details, see [42]).

#### 2) PHYSIONET

Physionet Motor/Mental Imagery database has been used. This dataset corresponds to a large-scale motor-imagery EEG data acquired from 109 participants while performing different motor imagery tasks. The EEG data were sampled at 160 Hz from 64-channels using the BCI2000 system according to the international 10-10 electrode placement system. Here, we chose the trials that correspond to right-hand and left-hand motor imagery tasks. Each recorded trial lasted for 4 seconds, which was then followed by 4 seconds rest. The total number of trials acquired for right-hand and left-hand motor imagery tasks was equal to 46 trials for each subject. The dataset from this experiment is publicly available at [43]—readers are referred to [44], [45] for more details.

#### 3) BCI COMPETITION VI - DATASET 2a (BCI-DataSet2A)

This dataset was acquired from nine subjects using a cue-based BCI paradigm. Participants performed four different types of imagery tasks that included imagination of left-hand, right-hand, both feet, and tongue movement. The data were obtained from two different sessions wherein each session, 288 trials were collected for a given imagery task. A session started with a fixation cross on a black screen, and after two seconds, an additional visual cue arrow pointing to the left, right, up, and down was shown for 1.25 seconds. The participants were instructed to perform motor imagery tasks until the fixation cross disappeared on the screen after

six seconds. EEG data were recorded monopolar using 22 Ag/AgCl electrodes according to the 10-20 electrode placement system. The reference electrode was set to the left mastoid while the ground was set to the right mastoid. The data were sampled with 250 Hz and bandpass-filtered between 0.5 Hz and 100 Hz (for more details, see [46]).

#### 4) BCI COMPETITION VI - DATASET 2b (BCI-DataSet2B)

This dataset consists of EEG data recorded from nine healthy, right-handed participants. EEG data were sampled with 250 Hz using three channels (C3, Cz, and C4) and bandpass-filtered between 0.5 Hz and 100 Hz. Each subject participated in five data acquisition sessions, where the first two sessions were conducted without feedback, and the last three sessions with feedback. The electrode Fz was used as EEG ground. Participants performed two-class motor imagery of the left-hand and right-hand movements using a cue-based paradigm. The total number of data samples recorded from each session for each motor imagery task was equal to 120 trials. Each trial in a session started with a fixation cross and an additional short acoustic warning tone. A few seconds later, a visual cue arrow was shown for 1.25 seconds, after which the participants were instructed to perform motor imagery tasks over four seconds. There was a short break (1.5 seconds) between each consecutive trial. The session with feedback used a smiley on a screen at the beginning of each trial ( $t = 0$ ). The visual cue was shown between  $t = 3$  to  $t = 7.5$  seconds, and subjects performed the motor imagery tasks (for more details, see [47]).

### B. EEG PRE-PROCESSING

The deep learning research has shown enhanced performance in learning from raw EEG data, mitigating the need for preprocessing or handcrafted features [19], [21], [34]. Here, we apply a minimal preprocessing to all four datasets described in the previous section, and refer readers to [21], [35] for more details regarding the preprocessing steps and usefulness of deep learning models employed. In this regard, the EEG waveforms were high-pass filtered above 4 Hz using a fourth-order Butterworth IIR filter. The high-pass filter with 4 Hz cut-off frequency was used to suppress electro-oculographic artifacts that arose due to eye movement dominant between 0.1 and 4 Hz band in EEG. Other than that, and as it was suggested by [21], we did not apply low-pass filtering to leave the raw EEG data intact.

Further, the continuous EEG was segmented into a left-hand and right-hand imagination trials with a four-second length following the motor imagery onset. Subsequently, EEG data trials were artifact corrected by applying a statistical threshold to exclude: (i) bad EEG trials correlated with egregious movement noise; and (ii) channels that are noisy because of possible poor connection to the scalp of a participant. Bad trials were identified by calculating the mean absolute value per trial and eliminating trials with values higher than three standard deviations over the mean trial.

All the methods described here have been implemented in the MNE Python environment [48].

The minimal preprocessing steps adopted allows the constructed CNNs to learn discriminative features automatically from a nearly set of raw EEG data without the imposition of prior knowledge through excessive preprocessing.

### III. CLASSIFICATION RULES

The key question in this study is whether using standard convolutional neural networks within a systematic model selection would possibly lead to a comparable classification accuracy for motor imagery tasks as the current state-of-the-art deep learning architectures, which are partially inspired by domain knowledge. Therefore, we first briefly describe the main building block of standard CNNs, which is the convolutional operation. We then briefly describe the EEGNet architecture [35], which serves as the benchmark for comparison as it has shown the state-of-the-art performance in various EEG classification tasks. Hereafter, we refer to any specific CNN that is constructed as part of the systematic model selection process as *ConvNet*.

#### A. CONVOLUTIONAL LAYER

As the name suggests, a convolutional layer convolves an input tensor by another tensor parameters tuned by the learning algorithm (*kernel*). Let  $\mathbf{X}$  denote the 3-D input tensor with elements  $\mathbf{X}_{i,j,k}$  where  $i$ ,  $j$ , and  $k$  denote the channel (also known as depth), row, and column, respectively. Let us assume  $1 \leq i \leq c_{\mathbf{X}}$  (input channels),  $1 \leq j \leq h_{\mathbf{X}}$  (input height), and  $1 \leq k \leq w_{\mathbf{X}}$  (input width). For our data, the first convolutional layer has  $c_{\mathbf{X}} = 1$  (similar to a grayscale image),  $h_{\mathbf{X}} = (\text{number of EEG channels})$ ,  $w_{\mathbf{X}} = [80 \text{ Hz} \times (4000) \text{ ms}] = 320$ .

The 4-D kernel tensor is denoted by  $\mathbf{K}$  with elements  $\mathbf{K}_{l,i,j,k}$  for  $1 \leq l \leq c_{\mathbf{Y}}$  (output channels),  $1 \leq i \leq c_{\mathbf{X}}$ ,  $1 \leq j \leq h_{\mathbf{K}}$  (kernel height), and  $1 \leq k \leq w_{\mathbf{K}}$  (kernel width) where  $h_{\mathbf{K}} < h_{\mathbf{X}}$  and  $w_{\mathbf{K}} < w_{\mathbf{X}}$ . Therefore,  $\mathbf{K}$  contains  $c_{\mathbf{Y}}$  kernels (filters) of size  $c_{\mathbf{X}}$  [same as input]  $\times$   $h_{\mathbf{K}}$  [smaller than input]  $\times$   $w_{\mathbf{K}}$  [smaller than input]. Convoluting  $\mathbf{K}$  across  $\mathbf{X}$  (with no padding/subsampling/bias terms) is obtained by computing the 3-D output tensor (*feature map*)  $\mathbf{Y}$  with elements  $\mathbf{Y}_{l,m,n}$  where  $1 \leq l \leq c_{\mathbf{Y}}$ ,  $1 \leq m \leq h_{\mathbf{Y}} = h_{\mathbf{X}} - h_{\mathbf{K}} + 1$ , and  $1 \leq n \leq w_{\mathbf{Y}} = w_{\mathbf{X}} - w_{\mathbf{K}} + 1$  as

$$\mathbf{Y}_{l,m,n} = \sum_{i,j,k} \mathbf{X}_{i,m+j-1,n+k-1} \mathbf{K}_{l,i,j,k}, \quad (1)$$

where the summation is over all valid indices. When there are multiple convolutional layers in one network, the output feature map of one layer serves as the input to the next layer. Assuming one bias term for each kernel, in training a CNN  $c_{\mathbf{Y}}(c_{\mathbf{X}}h_{\mathbf{K}}w_{\mathbf{K}} + 1)$  parameters are learned for each layer using the back-propagation algorithm [49].

#### B. EEGNet: A BENCHMARK FOR COMPARISON

As a benchmark for comparing the performance of constructed ConvNets presented in the next section,

**TABLE 2.** The EEGNet model architecture used for decoding subject-specific EEG data.

Index	Layers
1	EEG Input
2	Conv2d (Temporal)
3	BatchNorm2d
4	Conv2d (Spatial)
5	BatchNorm2d
6	ELU Activation
7	AvgPool2d
8	Dropout
9	Conv2d (sep_depth)
10	Conv2d (sep_point)
11	BatchNorm2d
12	ELU Activation
13	AvgPool2d
14	Dropout
15	Conv2d (Classifier)
16	Linear

**TABLE 3.** Total number of observations for each pooled datasets collected as part of the Training and Validation.

Data Type	Number of Subjects	Training Set	Validation Set
WEIBO2014	10	1264	316
PhysioNet	109	3713	928
BCI Data-2a	9	2074	518
BCI Data-2b	9	5217	1303

we consider a well-known deep learning architecture known as EEGNet [35].

The EEGNet architecture is composed of three convolutional layers with the following characteristics: the first convolutional layer uses a temporal convolution to learn the parameters of frequency filters. The second convolutional layer uses a depthwise convolution designed to learn frequency-specific spatial filters. The third convolutional layer uses a depth-wise convolution, which is used to learn a temporal summary for each feature map individually, followed by a pointwise convolution, which is used to learn an optimal combination of feature maps. The details of the network architecture can be found in Table 2 where the convolutional operations, batch normalization, the exponential linear unit activation function, the average pooling operation, and the fully connected linear layer are identified by Conv2d, BatchNorm2d, ELU, AvgPool2d, and linear, respectively. We re-implemented EEGNet models in PyTorch using the source code-shared by the authors in [35].

#### IV. TRAINING AND MODEL SELECTION

Due to the physical and mental burden on human subjects in motor imagery experiments, it is usually challenging to collect a large sample for each subject. Generally, a limited number of EEG training observations can be acquired in one experimental session, which itself usually lasts an hour. Such an experiment should be repeated over weeks and months to

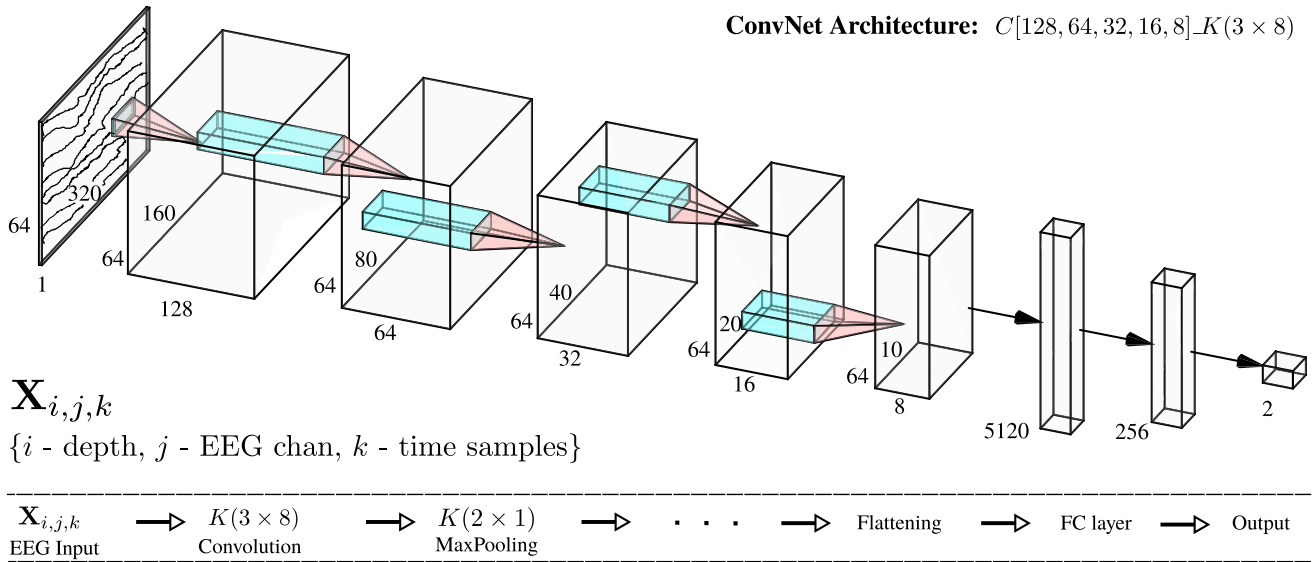
collect a large sample, which is generally required to train deep neural networks. Table 3 shows the total number of observations (i.e., signal segments of 4000 ms; as described in Sections II-A and II-B) that were used in training and validation sets (used for model selection described next in this section) across both right-hand and left-hand motor imagery classes for all datasets used in our study (see Section II-A for description of these datasets). For each dataset, we pooled the observations collected for all subjects within the dataset, 80% of this pooled sample was randomly set aside for training, and the rest was left for model selection. To increase the sample size, one may segment the collected waveforms to a series of shorter time intervals [21], [50]. In this regard, we generated multiple training observations by subsampling the original complete trial with a fixed time window of length  $t = 1500$  ms with a 50% overlap along the time axis. Subsequently, we trained ConvNets on multiple time-window segments obtained from the completed trials.

The following assumptions were made to define the possible search space of hyperparameters: (i) the maximum number of examined epochs for all models was set to 150 (no early stopping); (ii) the mini-batch gradient descent (batch size of 64) with Adam optimizer with a learning rate of 0.001 and a decay of 0.0001 was examined [51]; (iii) the loss function was cross-entropy [52]; (iv) the maximum number of convolutional layers that were examined was 6 with a single fully connected layer; (v) a dropout layer with the retention probability of  $p = 0.5$  was used prior to the fully connected layer [53]; (vi) assuming  $h_{\mathbf{K},j}$  and  $w_{\mathbf{K},j}$  denote the height and the width of the kernel used in layer  $j$ , respectively, where  $j = 1, \dots, L$  with  $L$  being the total number of layers used in the architecture ( $2 \leq L \leq 6$ ), in training ConvNets, we used a common rectangular shape kernel for all layers such that  $h_{\mathbf{K},j} = h_{\mathbf{K}} = 3, \forall j$ , and  $w_{\mathbf{K},j} = w_{\mathbf{K}} = 8 * \tau, \forall j$  where  $\tau \in [1, 3, 5]$  was introduced to define convolutional filters that cover different temporal features of EEG<sup>1</sup>; and (vii) assuming  $c_{\mathbf{Y},j}$  denotes the number of output channels in layer  $j$ , we considered both an increasing and decreasing pattern for  $c_{\mathbf{Y},j}$  with  $c_{\mathbf{Y},j} = 2^{2+j}$  and  $c_{\mathbf{Y},j} = 2^{L+3-j}$ .

The above set of assumptions sets the cardinality of hyperparameter space to  $4500 = 150$  (epoch)  $\times$   $10$  (patterns of  $c_{\mathbf{Y},j}$ )  $\times$   $3$  (possible values of  $w_{\mathbf{K}}$ )—the latter two hyperparameters define 30 possible ConvNet architectures. In addition, all ConvNets are based on a stack of standard convolutional layers; that is to say, each layer includes a convolutional operation (Section III-A), batch normalization, the rectified linear unit activation function, and the max-pooling operation. Using the defined hyperparameter space, we deployed a two-stage systematic model selection.

In the first stage, we used the validation set within each dataset to estimate the optimal set of algorithmic hyperparameters within the defined hyperparameter space

<sup>1</sup>Given  $f_s = 80$  Hz sampling rate, the temporal window of  $t = 100$  ms would be covered by a kernel width of  $w_{\mathbf{K}} = 8$ , and  $t = 300$  ms,  $t = 500$  ms are covered by kernel widths of  $w_{\mathbf{K}} = 24$ ,  $w_{\mathbf{K}} = 40$ , respectively



**FIGURE 1.** The architecture of ConvNet<sub>opt</sub> for decoding event-related synchronization and desynchronization in EEG signals for motor imagery BCIs. The input to the ConvNet<sub>opt</sub> model is represented as 3-D tensor,  $\mathbf{X}_{i,j,k}$  where  $\{i = 1, j \in [64 \text{ or } 22 \text{ or } 3], k = 320\}$ . The convolution is performed with a kernel size of  $K(3 \times 8)$ .

(here, the epoch size only), given all structural hyperparameters fixed. In this regard, for each dataset and each combination of structural hyperparameters, we trained a ConvNet and recorded its accuracy at the epoch size, which led to the highest accuracy of the validation set within that dataset. In the second stage, the optimal combination of structural hyperparameters was estimated as the one that led to the highest average accuracy across all datasets used for model selection. In order to maximize the number of subjects that were used in the model selection stage, we used three datasets with the highest number of subjects for model selection (Physionet, Weibo2014, and BCI-DataSet2A) and set aside BCI-DataSet2B for testing—BCI-DataSet2A and BCI-DataSet2B have the same number of subjects and the assignment was done arbitrarily. All this process was conducted using a Linux workstation with Intel Core i9-9900K, (3.6 GHz) processor, 32 GB of RAM, and Nvidia GeForce RTX 2080 Ti (RAM = 11GB, CUDA Cores: 4352). The entire workflow was implemented in Pytorch deep learning environment.

## V. RESULTS

In this section, we present the results of the systematic model selection described in Section IV. Table 4 shows the classification accuracies achieved at the epoch that led to the highest accuracy on the validation set within each dataset used for model selection (Physionet, Weibo2014, and BCI-DataSet2A)—this was the first stage of model selection as described in Section IV. In this table,  $C[c_{Y,1}, \dots, c_{Y,L}]$  and  $\mathbf{K}(h_{\mathbf{K}} \times w_{\mathbf{K}})$  are used to represent the number of output channels in each layer and the rectangular shape of the kernels that was in common across all layers, respectively. The rightmost column in this table is used to estimate the

optimal combination of structural hyperparameters within the defined hyperparameter space (i.e., the second stage of the model selection). The ConvNet structure that led to the highest average accuracy across all three datasets used in model selection is identified in bold in the first column. Hereafter, we refer to this structure ( $C[128, 64, 32, 16, 8]_K(3 \times 8)$ ) as ConvNet<sub>opt</sub>. Fig. 1 shows a schematic representation of ConvNet<sub>opt</sub> architecture.

Having estimated the structural hyperparameters, we now compare the performance of ConvNet<sub>opt</sub> with the EEGNet on a set of subjects within a dataset that was entirely unseen during the model selection (i.e., BCI-DataSet2B). Note that the aforementioned model selection led to a fixed set of structural hyperparameters (i.e., the structure of ConvNet<sub>opt</sub>), but the algorithmic hyperparameter (here, the epoch size) is left undetermined. Although using a similar strategy we could have estimated all hyperparameters including the epoch size, we left it to retraining process of ConvNet<sub>opt</sub> on each new data to estimate the optimal epoch size (similar to EEGNet in which we only use the structure, and for each new dataset, the weights and the epoch size are estimated). As a result, to compare the performance of ConvNet<sub>opt</sub> and EEGNet on BCI-DataSet2B, we first randomly split the observations for each subject to training (70%), validation (15%), and test (15%) sets and use the validation set to estimate the optimal epoch size for each model.

Tables 5 and 6 show the subject-specific classification accuracy achieved using the EEGNet and the ConvNet<sub>opt</sub> on training, validation, and test sets for each subject within BCI-DataSet2B. Of particular importance in these tables is the classification accuracies achieved on the test data for each subject. Fig. 2 shows a subject-specific comparison between these accuracies. As seen in this figure, there is not a

**TABLE 4. Training and Validation Accuracies (in percentage) of ConvNet models for pooled sample on three different datasets indicated by Physionet, BCI Competition VI - Dataset 2a (BCI-DataSet2A) and Weibo-2014. The best model in terms of achieving the highest average accuracy on the train and validation sets are shown in bold.**

ConvNet Architectures	PhysioNet		BCI-DataSet2A		Weibo-2014		Average	
	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc
C[16, 8]_K(3 × 8)	73.20	71.77	99.95	73.02	96.42	60.47	89.85	68.42
C[16, 8]_K(3 × 24)	96.94	72.56	93.40	70.99	100	62.79	96.78	68.78
C[16, 8]_K(3 × 40)	94.58	73.02	95.84	67.14	81.75	63.79	90.72	67.98
C[32, 16, 8]_K(3 × 8)	86.78	75.28	100	71.20	96.08	64.78	94.29	70.42
C[32, 16, 8]_K(3 × 24)	75.30	74.72	72.07	72.21	98.83	70.43	82.07	72.45
C[32, 16, 8]_K(3 × 40)	96.43	74.94	98.37	72.21	97.58	69.10	97.46	72.09
C[64, 32, 16, 8]_K(3 × 8)	80.71	74.83	100	75.25	100	70.76	93.57	73.62
C[64, 32, 16, 8]_K(3 × 24)	87.92	75.51	100	74.44	96.92	72.09	94.94	74.02
C[64, 32, 16, 8]_K(3 × 40)	82.61	76.19	100	72.41	98.58	69.44	93.73	72.68
<b>C[128, 64, 32, 16, 8]_K(3 × 8)</b>	99.49	76.76	100	<b>75.66</b>	99.92	71.43	<b>99.80</b>	<b>74.62</b>
C[128, 64, 32, 16, 8]_K(3 × 24)	97.50	76.30	97.97	75.05	97.83	67.44	97.77	72.93
C[128, 64, 32, 16, 8]_K(3 × 40)	82.87	76.08	100	74.65	97.67	70.10	93.51	73.61
C[256, 128, 64, 32, 16, 8]_K(3 × 8)	80.88	73.81	86.74	71.40	67.47	66.89	78.37	70.70
C[256, 128, 64, 32, 16, 8]_K(3 × 24)	82.50	74.49	98.58	75.46	82.83	65.01	87.97	71.65
C[256, 128, 64, 32, 16, 8]_K(3 × 40)	78.70	75.74	100	72.41	84.53	64.78	87.74	70.98
<hr/>								
C[8, 16]_K(3 × 8)	75.07	71.54	100	69.37	82.08	60.13	85.72	67.02
C[8, 16]_K(3 × 24)	87.44	72.68	97.61	73.63	98.58	63.12	94.54	69.81
C[8, 16]_K(3 × 40)	87.52	73.47	98.78	67.75	75.50	62.13	87.27	67.78
C[8, 16, 32]_K(3 × 8)	75.38	73.70	98.22	74.65	96.75	62.13	90.12	70.16
C[8, 16, 32]_K(3 × 24)	76.40	73.47	73.54	73.43	98.17	66.78	82.70	71.22
C[8, 16, 32]_K(3 × 40)	78.45	74.15	83.09	73.83	92.08	65.45	84.54	71.14
C[8, 16, 32, 64]_K(3 × 8)	77.60	74.38	98.63	74.04	96.17	67.44	90.80	71.95
C[8, 16, 32, 64]_K(3 × 24)	94.95	75.40	77.86	71.60	100	<b>74.75</b>	90.94	73.92
C[8, 16, 32, 64]_K(3 × 40)	75.58	<b>77.21</b>	95.33	71.81	89.08	64.78	86.66	71.27
C[8, 16, 32, 64, 128]_K(3 × 8)	79.78	75.85	96.14	72.82	100	69.44	91.97	72.70
C[8, 16, 32, 64, 128]_K(3 × 24)	81.28	75.96	100	72.41	79.00	67.77	86.76	72.05
C[8, 16, 32, 64, 128]_K(3 × 40)	93.62	75.96	97.00	66.73	99.17	70.76	96.60	71.15
C[8, 16, 32, 64, 128, 256]_K(3 × 8)	80.37	73.70	97.60	66.19	98.83	63.34	92.27	67.74
C[8, 16, 32, 64, 128, 256]_K(3 × 24)	75.50	73.92	66.43	66.33	65.81	62.68	69.24	67.64
C[8, 16, 32, 64, 128, 256]_K(3 × 40)	81.76	73.92	67.19	64.71	69.14	64.45	72.70	67.69

significant difference between the classification accuracies achieved by ConvNet<sub>opt</sub> and EEGNet—a two-sided Wilcoxon rank-sum test does not reject the null hypothesis of having a difference between these two sets of accuracies ( $p = 0.51$ ). In other words, the classification accuracies achieved by ConvNet<sub>opt</sub> and EEGNet are comparable on BCI-DataSet2B.

Fig. 3 shows the total training time of the selected ConvNet<sub>opt</sub> architecture on all four datasets. The longer training time was naturally associated with the larger datasets (see Table 3). For instance, the BCI-DataSet2B took 5352 seconds, which is the longest duration among other datasets, elapsed to complete the training of the model.

## VI. DISCUSSION

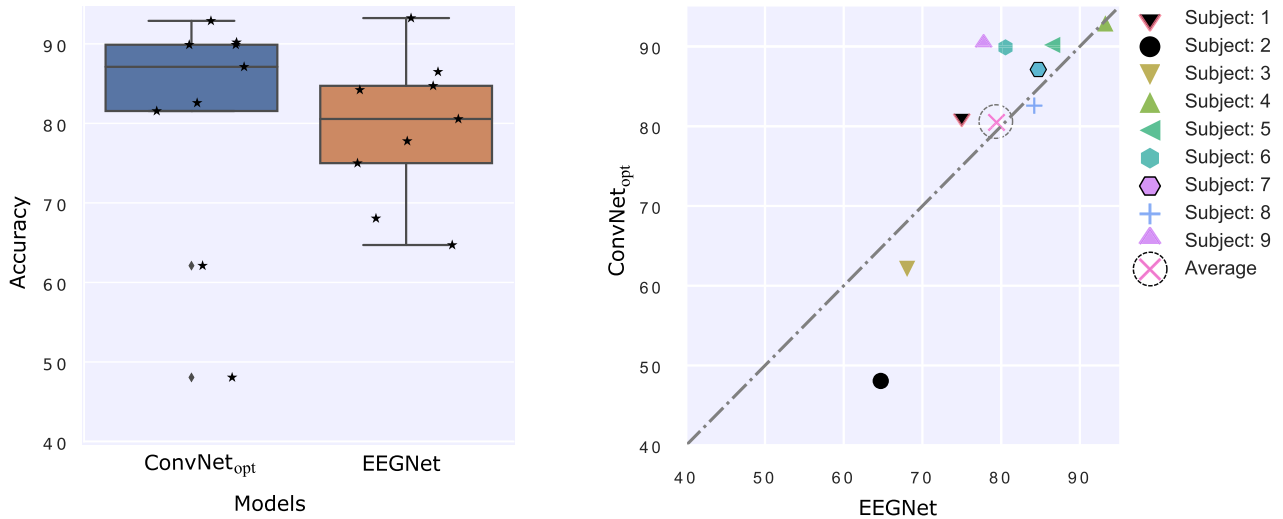
Deep learning technologies have offered unprecedented opportunities to construct remarkably accurate classifiers by integrating the process of feature extraction into the classifier training. However, this integration process comes at the price of tuning a large number of (algorithmic and structural) hyperparameters. This has partially led many studies to rely on existing well-known architectures such as AlexNet or ResNet, use the domain knowledge to construct

**TABLE 5. EEGNet subject-specific performance on the independent BCI Competition VI - Dataset 2b dataset (BCI-DataSet2B) that has not been used in model selection.**

Subjects	Train Acc	Val Acc	Test Acc	Epoch
Subject: 1	73.17	71.54	75.00	96
Subject: 2	69.53	64.23	64.71	104
Subject: 3	52.32	62.31	68.06	12
Subject: 4	96.24	97.76	93.24	23
Subject: 5	86.28	93.28	86.49	59
Subject: 6	88.61	83.08	80.56	109
Subject: 7	86.29	85.38	84.72	55
Subject: 8	87.57	85.40	84.21	51
Subject: 9	89.38	82.31	77.78	97
<b>Average</b>	81.04	80.59	79.42	

the final architecture, or have an unclear *ad hoc* strategy deployed for model selection. This raises the question of whether training accurate deep learning models using a principled model selection is possible, or, alternatively, experiencing and developing high intuition regarding the collected BCI data is the most prudent way to go about tuning the hyperparameters.

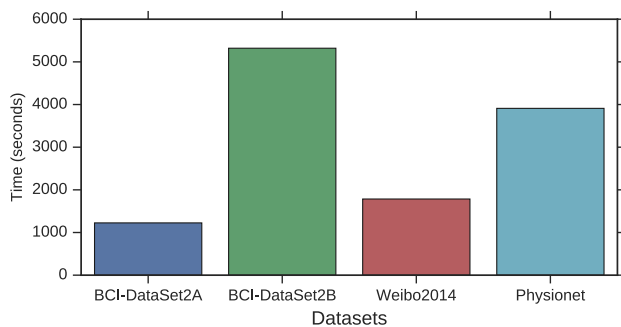
To address this question, in this work, we sought to show the efficacy of using standard convolutional neural networks within a systematic model selection for the classification



**FIGURE 2.** Classification accuracy (%) of EEGNet and ConvNet<sub>opt</sub> achieved on test data that was set aside for each subject within BCI-DataSet2B. The left pane shows the boxplot for the model performances on nine subjects where each star represents the subject-specific accuracy. The right pane shows pairwise comparison between EEGNet and ConvNet<sub>opt</sub> for each subject as well as the average accuracy obtained across all subjects.

**TABLE 6.** ConvNet<sub>opt</sub> (C[128, 64, 32, 16, 8]\_K(3 × 8)) subject-specific performance on the independent BCI Competition VI - Dataset 2b dataset (BCI-DataSet2B) that has not been used in model selection.

Subjects	Train Acc	Val Acc	Test Acc	Epoch
Subject: 1	99.63	83.94	80.56	99
Subject: 2	86.63	59.23	47.06	127
Subject: 3	73.86	61.31	61.11	133
Subject: 4	100	98.58	91.89	143
Subject: 5	100	93.62	89.19	140
Subject: 6	87.02	88.32	88.89	95
Subject: 7	80.44	85.40	86.11	94
Subject: 8	83.54	88.28	81.58	105
Subject: 9	81.90	90.51	88.89	99
<b>Average</b>	<b>88.11</b>	<b>83.24</b>	<b>79.47</b>	



**FIGURE 3.** Total training time of ConvNet<sub>opt</sub> on four datasets obtained from a total of 150 epochs using Intel Core i9-9900K (3.6 GHz) processor, 32 GB of RAM, and Nvidia GeForce RTX 2080 Ti (RAM = 11GB, CUDA Cores: 4352).

of sensorimotor rhythms that arise during motor imagery tasks. In this regard, we used a two-stage systematic model selection. In the first stage, the validation set within each dataset was used to estimate the optimal set of algorithmic hyperparameters within the defined hyperparameter space given all structural hyperparameters fixed. The validation set used in this stage was a portion of observations randomly

set aside from the pooled sample of subjects within each dataset—pooled design *per se*) was a feasible solution for reducing the extensive calibration time of BCI systems for individual subjects [54]. In the second stage, the optimal combination of structural hyperparameters was estimated as the one that led to the highest average accuracy across all datasets used for model selection.

The aforementioned model selection led to a fixed set of structural hyperparameters. However, the only variable algorithmic hyperparameter (i.e., the epoch size) was left free and is estimated in the training process of the selected architecture on new data (similar to training the EEGNet architecture, which was used as the benchmark for comparison)—training the EEGNet and the ConvNet<sub>opt</sub> architectures for nine additional subjects from BCI-DataSet2B that was entirely unseen during the model selection led to a set of (statistically) comparable accuracies with EEGNet. Moreover, ConvNet<sub>opt</sub> is invariant on the spatial dimensionality of the input data and is able to capture discriminative spatial features for accurate classification. Going back to the main question of this study, this observation shows the possibility of using standard CNNs within a systematic brute-force model selection to achieve comparable classification accuracy as the state-of-the-art deep learning architectures used in the classification of motor imagery tasks.

In other words, our study compares the use of prior knowledge versus data in the context of model selection. Naturally, if “good” prior knowledge about the nature of data and a mechanism for encoding this knowledge into the structure of a classification rule is available, we may expect training highly accurate predictive models; however, in the absence of such prior knowledge or encoding mechanism, we may look into conducting a data-driven brute-force model selection as a viable option. Nevertheless, the performance of the selected structure, which is the outcome of this brute-force

model selection, depends on the pre-specified hyperparameters space. Here, we showed that in so far as classification of sensorimotor rhythms is concerned, a pre-specified space that was restricted by our computing power and defined based on common values of hyperparameters can lead to accurate classifiers.

The systematic model selection strategy used here is not, of course, the only way to go about estimating the hyperparameters; for example, various other performance metrics or network pruning techniques could be used. Depending on computing power capacity, one may even define a much larger hyperparameter space or may use some suboptimal search strategies (as opposed to the exhaustive search used here). Furthermore, one may estimate both the algorithmic and structural hyperparameters in model selection. Nonetheless, we believe that the results obtained here would suffice to verify the efficacy of conducting a brute-force CNN model selection within a limited hyperparameter space.

## VII. CONCLUSION

In this work, we examined whether a brute-force search with a limited space of hyperparameters for standard convolutional neural networks (CNNs) would possibly lead to a comparable classification accuracy as the state-of-the-art deep learning architectures for classification of motor imagery tasks. In this regard, we conducted the first large-scale analysis in which we use EEG data collected from 128 subjects across three different datasets to determine the single architecture that achieved the highest average accuracy on pooled-sample validation sets across these datasets. Retraining the identified architecture on a set of nine additional subjects within another dataset that was entirely unseen during the model selection stage led to a comparable set of classification accuracies as the EEGNet. This result of our investigation does not undermine the capacity and efficacy of the EEGNet, which was used as the benchmark for comparison. EEGNet has shown the state-of-the-art performance in various EEG classification tasks across various BCI paradigms; however, its structure is partially inspired by the domain knowledge, which is, of course, not possessed by non-experts. The findings of current study on systematic CNN model selection may serve as guide to the deep learning practitioners with minimal domain knowledge to propose robust BCI models in the classification of motor imagery tasks.

## REFERENCES

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–791, 2002.
- [2] A. Vallabhaneni, T. Wang, and B. He, "Brain-computer interface," in *Neural Engineering*. Boston, MA, USA: Springer, 2005, pp. 85–121.
- [3] N. Birbaumer and L. G. Cohen, "Brain-computer interfaces: Communication and restoration of movement in paralysis," *J. Physiol.*, vol. 579, no. 3, pp. 621–636, Mar. 2007.
- [4] S. Machado, F. Araújo, F. Paes, B. Velasques, M. Cunha, H. Budde, L. F. Basile, R. Anghinah, O. Arias-Carrión, M. Cagy, R. Piedade, T. A. de Graaf, A. T. Sack, and P. Ribeiro, "EEG-based brain-computer interfaces: An overview of basic concepts and clinical applications in neurorehabilitation," *Rev. Neurosci.*, vol. 21, no. 6, pp. 451–468, Jan. 2010.
- [5] A. Kilicarslan, S. Prasad, R. G. Grossman, and J. L. Contreras-Vidal, "High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 5606–5609.
- [6] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–375, May 2012.
- [7] N. A. Bhagat, A. Venkatakrisnan, B. Abibullaev, E. J. Artz, N. Yozbatiran, A. A. Blank, J. French, C. Karmonik, R. G. Grossman, M. K. O'Malley, G. E. Francisco, and J. L. Contreras-Vidal, "Design and optimization of an EEG-based brain machine interface (BMI) to an upper-limb exoskeleton for stroke survivors," *Frontiers Neurosci.*, vol. 10, p. 122, Mar. 2016.
- [8] A. Venkatakrisnan, G. E. Francisco, and J. L. Contreras-Vidal, "Applications of brain-machine interface systems in stroke recovery and rehabilitation," *Current Phys. Med. Rehabil. Rep.*, vol. 2, no. 2, pp. 93–105, Jun. 2014.
- [9] M. Gerven, J. Farquhar, R. Schaefer, R. Vlek, J. Geuze, and, "The brain-computer interface cycle," *J. Neural Eng.*, vol. 6, no. 4, 2009, Art. no. 041001.
- [10] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 4, p. R1, Jun. 2007.
- [11] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain-computer interfaces: A 10 year update," *J. Neural Eng.*, vol. 15, no. 3, Jun. 2018, Art. no. 031005.
- [12] J. Wolpaw and E. W. Wolpaw, *Brain-Computer Interfaces: Principles and Practice*. Oxford, U.K.: OUP, 2012.
- [13] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.
- [14] U. Chaudhary, N. Birbaumer, and A. Ramos-Murguialday, "Brain-computer interfaces in the completely locked-in state and chronic stroke," *Prog. Brain Res.*, vol. 228, pp. 131–161, 2016.
- [15] G. Pfurtscheller, "Spatiotemporal erd/ers patterns during voluntary movement and motor imagery," *Supplements Clin. Neurophysiol.*, vol. 53, pp. 196–198, Jan. 2000.
- [16] G. Pfurtscheller and C. Neuper, "Future prospects of erd/ers in the context of brain-computer interface (bci) developments," *Prog. Brain Res.*, vol. 159, pp. 433–437, Jan. 2006.
- [17] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch, "A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals," *J. Neural Eng.*, vol. 4, no. 2, pp. R32–R57, Mar. 2007.
- [18] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *J. Neural Eng.*, vol. 16, no. 5, Aug. 2019, Art. no. 051001.
- [19] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning based brain computer interface: Recent advances and new frontiers," 2019, *arXiv:1905.04149*. [Online]. Available: <http://arxiv.org/abs/1905.04149>
- [20] S. Aggarwal and N. Chugh, "Signal processing techniques for motor imagery brain computer interface: A review," *Array*, vols. 1–2, Jan. 2019, Art. no. 100003.
- [21] R. T. Schirmermeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggenberger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Nov. 2017.
- [22] T. Uktveris and V. Jusas, "Application of convolutional neural networks to four-class motor imagery classification problem," *Inf. Technol. Control*, vol. 46, no. 2, pp. 260–273, Jun. 2017.
- [23] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain-computer interface using convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5619–5629, Nov. 2018.
- [24] G. Dai, J. Zhou, J. Huang, and N. Wang, "HS-CNN: A CNN with hybrid convolution scale for EEG motor imagery classification," *J. Neural Eng.*, vol. 17, no. 1, Jan. 2020, Art. no. 016025.
- [25] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *J. Neural Eng.*, vol. 14, no. 1, Feb. 2017, Art. no. 016003.

- [26] X. Tang, J. Yang, and H. Wan, "A hybrid SAE and CNN classifier for motor imagery eeg classification," in *Proc. Comput. Sci. On-Line Conf.* Cham, Switzerland: Springer, 2018, pp. 265–278.
- [27] M. Dai, D. Zheng, R. Na, S. Wang, and S. Zhang, "EEG classification of motor imagery using a novel deep learning framework," *Sensors*, vol. 19, no. 3, p. 551, Jan. 2019.
- [28] W. Qiao and X. Bi, "Deep spatial-temporal neural network for classification of EEG-based motor imagery," in *Proc. Int. Conf. Artif. Intell. Comput. Sci. AICS*, 2019, pp. 265–272.
- [29] R. Zhang, Q. Zong, L. Dou, and X. Zhao, "A novel hybrid deep learning scheme for four-class motor imagery classification," *J. Neural Eng.*, vol. 16, no. 6, Oct. 2019, Art. no. 066004.
- [30] S. Chaudhary, S. Taran, V. Bajaj, and A. Sengur, "Convolutional neural network based approach towards motor imagery tasks EEG signals classification," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4494–4500, Jun. 2019.
- [31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [32] C. J. Ortiz-Echeverri, S. Salazar-Colores, J. Rodríguez-Reséndiz, and R. A. Gómez-Loenzo, "A new approach for motor imagery classification based on sorted blind source separation, continuous wavelet transform, and convolutional neural network," *Sensors*, vol. 19, no. 20, p. 4541, Oct. 2019.
- [33] Z. Zhang, F. Duan, J. Sole-Casals, J. Dinares-Ferran, A. Cichocki, Z. Yang, and Z. Sun, "A novel deep learning approach with data augmentation to classify motor imagery signals," *IEEE Access*, vol. 7, pp. 15945–15954, 2019.
- [34] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *J. Neural Eng.*, vol. 16, no. 3, Jun. 2019, Art. no. 031001.
- [35] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, Oct. 2018, Art. no. 056013.
- [36] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Frontiers Neurosci.*, vol. 6, p. 39, Mar. 2012.
- [37] K. Keng Ang, Z. Yang Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) in brain-computer interface," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 2390–2397.
- [38] X. Zhu, P. Li, C. Li, D. Yao, R. Zhang, and P. Xu, "Separated channel convolutional neural network to realize the training free motor imagery BCI systems," *Biomed. Signal Process. Control*, vol. 49, pp. 396–403, Mar. 2019.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [40] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1251–1258.
- [41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [42] W. Yi, S. Qiu, K. Wang, H. Qi, L. Zhang, P. Zhou, F. He, and D. Ming, "Evaluation of EEG oscillatory patterns and cognitive process during simple and compound limb motor imagery," *PLoS ONE*, vol. 9, no. 12, Dec. 2014, Art. no. e114853.
- [43] *Physionet Motor/Mental Imagery (MMI) Database*. Accessed: Dec. 18, 2019. [Online]. Available: <https://archive.physionet.org/pn4/eegmmidb>
- [44] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.
- [45] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: A general-purpose brain-computer interface (BCI) system," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004.
- [46] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz, "Review of the BCI competition IV," *Frontiers Neurosci.*, vol. 6, p. 55, Jul. 2012.
- [47] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, "Brain–computer communication: Motivation, aim, and impact of exploring a virtual apartment," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 4, pp. 473–482, Dec. 2007.
- [48] A. Gramfort, "MEG and EEG data analysis with MNE-Python," *Frontiers Neurosci.*, vol. 7, p. 267, Dec. 2013.
- [49] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining Practical Machine Learning Tools and Techniques*, 4th ed. Amsterdam, The Netherlands: Elsevier, 2017.
- [50] S. M. Kay, "Fundamentals of statistical signal processing," in *Practical Algorithm Development*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2013.
- [51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*, [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [54] F. Lotte, "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces," *Proc. IEEE*, vol. 103, no. 6, pp. 871–890, Jun. 2015.



**BERDAKH ABIBULLAEV** (Member, IEEE) received the M.Sc. and Ph.D. degrees in electronic engineering from Yeungnam University, South Korea, in 2006 and 2010, respectively. He held research scientist positions at the Daegu Gyeongbuk Institute of Science and Technology, from 2010 to 2013, and the Samsung Medical Center, Seoul, South Korea, from 2013 to 2014. He is currently an Assistant Professor with the Robotics Department, Nazarbayev University, Kazakhstan.

His research interests include machine learning algorithms, neural signal processing, and brain–computer/machine interfaces. In 2014, he received the National Institute of Health Postdoctoral Research Fellowship II to join a multiinstitutional research project between the University of Houston BMI Systems Team and the Texas Medical Center in developing neural interfaces for rehabilitation.



**IRINA DOLZHIKOVA** (Member, IEEE) received the bachelor's degree (Hons.) and the master's degree in electrical and electronic engineering from Nazarbayev University, Kazakhstan, in 2016 and 2018, respectively. She is currently pursuing the Ph.D. degree in machine learning with the focus on the EEG-based brain–computer interfaces. She was a Teaching Assistant with Nazarbayev University, from 2017 to 2018.



**AMIN ZOLLANVARI** (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Shiraz University, Iran, and the Ph.D. degree in electrical engineering from Texas A&M University at College Station, College Station, TX, USA, in 2010. He held a postdoctoral position at the Harvard Medical School and the Brigham and Women's Hospital, Boston, MA, USA, from 2010 to 2012. He was an Assistant Research Scientist with the Department of Statistics, Texas A&M University, from 2012 to 2014. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Nazarbayev University, Kazakhstan. His research interests include machine learning, statistical signal processing, and biomedical informatics.