Subject-Independent Classification of P300 Event-Related Potentials Using a Small Number of Training Subjects

Article *in* IEEE Transactions on Human-Machine Systems · October 2022

DOI: 10.1109/THMS.2022.3189576

CITATIONS
2

READS
50

3 authors:

Berdakh Abibullaev
Nazarbayev University
65 PUBLICATIONS **743** CITATIONS
SEE PROFILE

Kassymzhomart M Kunanbayev
Korea Advanced Institute of Science and Technology
8 PUBLICATIONS **51** CITATIONS
SEE PROFILE

Amin Zollanvari
Nazarbayev University
90 PUBLICATIONS **838** CITATIONS
SEE PROFILE

Subject-Independent Classification of P300 Event-Related Potentials using a Small Number of Training Subjects

Berdakh Abibullaev, *Senior Member, IEEE*, Kassymzhomart Kunanbayev, and Amin Zollanvari, *Senior Member, IEEE*

***Abstract*—The inter-subject variability present in electroencephalographic (EEG) signals can affect the performance of the brain-computer interface (BCI) systems. Despite the significant progress in the field, the variability in neural data remains one of the most critical challenges in constructing accurate predictive models of human intention. As a result, the majority of the previous studies have focused either on devising subject-specific signal processing and machine learning algorithms, have used some data from a target user to update and calibrate a pre-trained classifier, or have used data collected from a relatively large number of training subjects to construct generic classifiers for new subjects. In this work, we investigate the feasibility of using a relatively small number of training subjects to achieve subject-independent classification of event-related potentials (ERPs) in P300-based BCIs. To this end, we employ convolutional neural networks (CNNs) and propose a leave-one-subject-out cross-validation (LOSO-CV) for model selection; that is to say, for tuning CNN hyperparameters including number of layers, filters, kernel size, and epoch. The utility of the proposed model selection is warranted because LOSO-CV simulates the effect of subject-independent classification within the training data. The entire process of training (including model selection) is validated by applying another LOSO-CV external to the training process. Our empirical results obtained on four publicly available datasets confirm the capability of LOSO-CV model selection with CNN to capture intrinsic ERP features from a small group of subjects to classify observations collected from unseen subjects.**

***Index Terms*—Brain-Computer Interfaces, Event-Related Potentials, Subject-Independent Classification, P300, Deep Learning, Convolutional Neural Networks, Model Selection.**

I. Introduction

BRAIN - Computer Interface (BCI) is a system that measures brain activity of an individual and translates it into commands to communicate and control output devices [1], [2] thereby bypassing a natural communication channel between the central and peripheral nervous system [3], [4]. One potential application of BCIs is for people who have amyotrophic lateral sclerosis (ALS) disorder, which affects neurons responsible for voluntary movements in humans [5]. Patients rapidly lose the ability to move their arms, legs, and face muscles, gradually become unable to communicate [6] while their brain is fully functional. The only means of communication for those patients are using a BCI based speller system [7]–[10]. This technology has attracted extensive attention in the past decade because of its potential in improving the lives of people affected by severe paralysis [7], [11]–[16].

A BCI speller system tries to infer brain activity patterns, as measured by Electroencephalography (EEG), and communicate its output into a computer application such as typing on a screen [4], [8], [9], [17]. The most standard component of EEG is event-related potentials (ERP), which BCI speller uses to infer brain activity patterns using a set of signal processing and machine learning techniques [18]–[21]. The ERPs are attention modulated brain responses time-locked to an external visual/auditory or haptic stimuli events delineated by different spatiotemporal characteristics to background neural activities. The discrepancy lies in the P300 component of ERPs. The P300 wave can be observed when a user attends to a rare stimulus with a positive peak at 300-400 ms after the stimulus event onset [19], [22]. A central task in the design of a BCI speller is to decode the presence of target P300 patterns in the ERP on a single trial basis and translate into appropriate communication commands. However, accurate decoding of ERPs is challenging due to poor signal-to-noise ratio, the presence of non-task relevant neural processes, and enormous variability in EEG data. In particular, the ERP distribution varies between subjects, sessions, and even between trials deteriorating BCI performances [23]. Such variability in data can be characterized in terms of covariate shift problem [24], complex neurophysiology, and neuroanatomy of users [25], or due to a specific experimental setup.

The design of a high-performance BCI system is still an open research problem in the community. The machine learning algorithms play an important role in accurately identifying brain activity patterns and translating them into appropriate commands. In fact, these algorithms have been the key component of BCI research, which aims to extract sensitive signatures of neural signals in which human mental thoughts are best encoded. A wide class of methods have already been utilized in different EEG-based BCI applications, [26]–[31], focusing on problems related to signal processing [13], [32], feature extraction [33], classification [34], system design [35] and clinical applications [36].

Rapid advances in deep learning have already led to a number of predictive models that can potentially address the subject variability in EEG data [37]. For example, Cecotti

B. Abibullaev is with the Department of Robotics and Mechatronics, Nazarbayev University, Kazakhstan (e-mail: berdakh.abibullaev@nu.edu.kz)

K. Kunanbayev and A. Zollanvari are with the Department of Electrical and Computer Engineering, Nazarbayev University, Kazakhstan (e-mail: kassymzhomart.kunanbayev@nu.edu.kz, amin.zollanvari@nu.edu.kz)

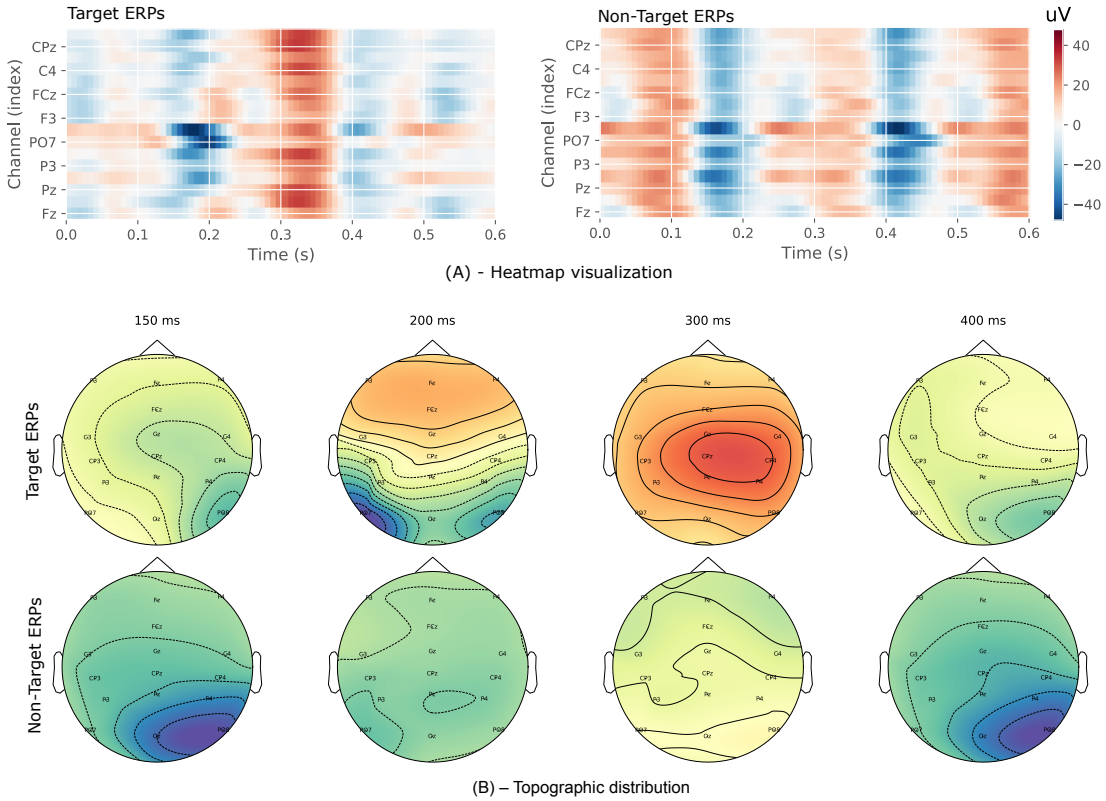

Fig. 1. A time course evolution of grand averaged sixteen channel ERPs from Subject #1 in BNCI14 dataset. The upper panel (A) shows the color-coded representation of amplitudes of grand-averaged target ERPs and non-target ERPs within 600 ms. The lower panel (B) shows color-coded topographic distributions of target and non-target ERPs amplitude values across different time points. Target event-related brain responses patterns is pronounced clearly around 300 ± 50 ms time interval.

et al. introduced a two layers convolutional neural network (CNN) on a P300-ERPs subject-specific task [38]. Yoon *et al.* used CNN's capability to identify the key features of ERPs that distinguish the illiterates of the BCI speller system [39]. Liu *et al.* studied the effect of batch normalization in the input and convolutional layers to alleviate over-fitting in the ERP decoding task [40]. In another study, Maddula *et al.* proposed a three dimensional CNN in conjunction with two-dimensional CNNs to capture the discriminative Spatio-temporal structure of ERPs for an accurate classification [41]. Another recent study introduced a general-purpose architecture called EEGNet [42] for classifying EEG signals acquired from various BCI experiments. The architecture consists of three types of CNN layers, including a conventional layer, a depthwise layer, and a separable layer. Besides, several other neural networks have been proposed in EEG research, the majority of which were CNN-based architectures [37].

Despite these efforts, the inter-subject variability in EEG data remains one of the main challenges in designing a practical BCI speller. As a result, previous studies have focused either on subject-specific contexts [15], [18]–[20], [43], have used some data from a target user to update and calibrate a pre-trained classifier [16], [44]–[46], or have used a relatively large number of training subjects to conduct the subject-independent classification (e.g., 18, 21, and 31 subjects used in [47], [48], and [49], respectively) While the design of subject-specific BCIs may limit its usefulness in real-world applications, having a large number of training subjects to construct subject-independent classifiers may not be feasible either. The focus of the present investigation is warranted because, in many practical scenarios, investigators have only access to a small number of training subjects but are ultimately interested in constructing subject-independent classifiers. For example, in the present investigation, we use four publicly available datasets in which there are less than 10 subjects available within each dataset to train our classifiers.

In this regard, CNN has evidently become an attractive choice (see the aforementioned deep learning studies). This is mainly due to the inherent structure of CNN that exhibits hierarchical feature learning capabilities and is well suited to automatic extraction of spatiotemporal features in EEG. However, to leverage this capacity, estimating the optimal values of CNN hyperparameters is essential—a process referred to as *model selection*. While deep layered CNNs have proven efficient in general, the efficacy of systematic ways for tuning hyperparameters involved in their architectures is left underexplored in EEG literature. By treating the number of layers, filters, kernel size, and epoch as hyperparameters, in this work we propose, for what is believed to be the first time, a *leave-one-subject-out cross-validation (LOSO-CV) model selection* for CNNs used in the context of subject-independent classification. In LOSO-CV error estimation, we

successfully hold out the observations collected from each subject, construct a *surrogate classifier* on observations from the remaining subjects, and test the performance of the constructed surrogate classifiers on data from the held-out subject. The error estimate is then the overall proportion of errors committed on data from all held-out subjects. Although this process is not new in machine learning, to the best of our knowledge, it has not been used for model selection in the context of subject-independent classification of EEG signals. Despite the lack of utility of LOSO-CV for model selection in this context, it is indeed a natural procedure to use because it would simulate the effect of subject-independent classification within the training data and for model selection—recall that the surrogate classifiers are tested on held-out subjects within the training data. In other words, the utility of LOSO-CV model selection aligns the model selection procedure with the objective of the study, which is classifying observations collected from unseen subjects.

In order to validate the utility of the proposed LOSO-CV model selection for subject-independent classification, as in many other studies [50], [51], we simply use a LOSO-CV *external* to the training and model selection (i.e., here external to the inner LOSO-CV model selection). For K number of subjects in a dataset, the effect of the external LOSO-CV is to assess the performance of the entire (inner) LOSO-CV model selection K times (once for each subject whose data is kept unseen for training and model selection) and then take the average to obtain an overall performance estimate. Our results confirm the capability of the utilized LOSO-CV model selection for CNN to capture latent features from a limited group of training subjects to classify data collected from unseen subjects.

The paper is organized as follows. In Section II, we provide a description of (i) publicly available datasets used in our study; (ii) the preprocessing steps applied to our datasets. Section II-C presents the CNN architectures that we use as part of our brute-force model selection procedure. In Section II-E, we discuss the CNN training and model selection procedure, and present the results of applying constructed models in Section III to test data. We summarize our key observations in Section IV, and conclude the paper in Section V.

II. Materials and Methods

A. Dataset Descriptions

In this study, we have used four publicly available ERP datasets. The details of these datasets are described below.

1) BNCI14 dataset [52], [53]: This dataset presents ERP data collected using two different experimental paradigms based on overt and covert attention conditions on P300 Speller. Here we use the data from a paradigm based on overt attention condition, which is a standard experimental paradigm employed in most BCI studies initially proposed by Farwell & Donchin [22]. Ten healthy subjects (10 female, mean age = 26.8 $\pm$ 5.6) were presented with a Farwell & Donchin style [22] 6×6 grid of alphanumeric characters via an LCD monitor. For each participant, EEG data were recorded from four recording sessions using 16 Ag/AgCl sensors (Fz, FCz, Cz, CPz, Pz, Oz, F3, F4, C3, C4, CP3, CP4, P3, P4, PO7, PO8) according to the international 10-10 electrode placement standard. The reference electrode was placed to the participants' linked earlobe, whereas a ground electrode location was attached to the right mastoid. The EEG was recorded at 256 Hz using a g.USBamp amplifier (g.Tec, Austria). High-pass and low-pass filters with cutoff frequencies of 0.1 Hz and 15 Hz, respectively, were used. One trial composed of eight stimulation sequences where each stimulus was intensified for 125 ms, with an inter stimulus interval of 125 ms, resulting in a 250 ms of stimulus-onset asynchrony (SOA). For more information on the experimental protocol, the readers are referred to [53]. Fig 1 shows an instance of the dataset from Subject #1, in particular, the time course evolution of grand averaged target and non-target ERP data. One can observe that target event-related brain response patterns is pronounced clearly around 300 ± 50 ms time interval.

2) BNCI15 dataset [54]: This dataset was collected from ten healthy individuals using a six-choice Farwell & Donchin style ERP paradigm. The EEG data were recorded at a 256 Hz sampling rate using a g.USBamp (24 Bit biosignal amplification unit, g.tec Medical Engineering GmbH, Austria) from eight electrodes ('Fz', 'Cz', 'P3', 'Pz', 'P4', 'PO7','Oz', 'PO8') placed according to the 10-20 international system. The ground electrode was placed on the forehead while the reference electrode was placed on the participants' right earlobe. Each participant completed one session in a single day without removing the EEG cap. Each row and column on the stimuli interface were randomly intensified for 100ms, with an inter-stimulus interval of 60 ms, yielding a 250 ms of SOA. The participants were instructed to attend to the stimulus character he/she is prompted to spell and, at the same time, count the number of times the character is highlighted. The counting was used to help the subject remain focused on the task. For more information on the experimental protocol, the readers are referred to [54]. Hereafter, we refer to this dataset as BNCI15. This dataset was accessed via the MOABB project described in [55].

3) ALS dataset [56]: This dataset was acquired from eight subjects (three women and five men; mean age + 58 $\pm$ 12) with amyotrophic lateral sclerosis (ALS) using the Farwell and Donchin paradigm. The subjects were asked to attend to a 6×6 grid of characters and spell seven predefined words using a BCI speller. Each row and column on the stimuli interface were randomly flashed for 125ms, with an inter-stimulus interval of 125 ms, creating a 250 ms of SOA. All columns and rows were flashed ten times (stimuli repetitions); hence, every single item on the stimuli interface was flashed 20 times. The EEG data were recorded with a sampling rate of 256 Hz using g.MOBILAB, g.tec, g.Ladybird, g.tec, Austria from locations: Fz, Cz, Pz, Oz, P3, P4, PO7, PO8. The subjects' right earlobe was used as a reference channel and the left mastoid as a ground. For more information on the experimental protocol, the readers are referred to [56]. Hereafter, we refer to this dataset as ALS data.

4) EPFL dataset [57]: This dataset was acquired from four disabled and four healthy individuals (all male, age 30 $\pm$ 2.3)

TABLE I
THE SIZE OF EEG DATASET (POOLED ACROSS ALL SUBJECTS WITHIN EACH DATASET) IN TERMS OF TRIALS, CHANNELS AND TIME. THE FIRST DIMENSION OF THE ARRAY CORRESPONDS TO THE NUMBER OF OBSERVATIONS, THE SECOND DIMENSION SHOWS THE NUMBER OF EEG CHANNELS, AND THE THIRD DIMENSION CORRESPONDS TO THE TEMPORAL FEATURES.

Dataset	(trials × channel × time)	Number of Subjects
BNCI14	(17280×16× 600)	10
BNCI15	(18983×16× 600)	10
EPFL	(26608× 32× 600)	8
ALS	(33600× 8× 600)	8

using a six-choice Farwell & Donchin style ERP paradigm at the 2048 Hz sampling rate from 32 electrodes. Each participant completed four experimental sessions completed within two weeks. Furthermore, each session was divided into six runs, with one run for each of the six stimuli images with an approximate duration of one minute. The stimuli consisted of six different images flashed randomly with a 400 ms stimulus interval. The duration of the stimulus was 100 ms, with the inter-stimulus interval equal to 400 ms, yielding 500 ms of SOA. The time between the same target image intensifications is equal to 700 ms. For more information on the experimental protocol, the readers are referred to [57]. Hereafter, we refer to this dataset as EPFL data.

Table I lists the EEG data dimensionality used in terms of total number of subjects, trials, channels, and temporal length (in milliseconds) after segmentation.

B. Preprocessing

The following steps summarize the signal preprocessing methods applied for the extraction of attention modulated ERPs.

1) *Segmentation*: Continuous EEG data were segmented into the target and non-target trials with 600 ms duration after the stimulus event onset markers.
2) *Detrending*: Arbitary offsets in data were removed by subtracting the total mean from each channel. This step ensures the removal of noise in the form of slow drifts, which might occur due to sweating and a poor sensor-to-head contact [58].
3) *Bad trial removal*: EEG data trials were artifact edited using a statistical thresholding procedure to eliminate bad trials associated with egregious movement artifacts. The criteria to detect bad trials was based on calculating mean absolute value per trial and excluding trials with values greater than 3-standard deviations over the median trial.
4) *Bad channel removal:* All channels were analyzed across trials to identify electrodes that are corrupted with excessive noise arising from improper connection to the scalp of a participant. Channels with excessively high power were identified by computing total power for each channel's overall epochs, mean channel power, and variance in channel power. Any channels with power more than three standard deviations were eliminated. These eliminated channels were replaced with a common averaged reference (CAR) channel. The CAR is defined as an averaged channel representing electrical activity measured across all channels.
5) *Spectral Filtering*: EEG data has been bandpass filtered between 0.5-12 Hz range using a Fourier filter [59]. First, the signal was Fourier transformed, and then a weighting is applied to suppress and remove unwanted frequencies outside the frequency of interest range. Finally, the weighted signal wave inverse Fourier transformed to obtain the filtered signal.

C. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [60] are a specific type of Artificial Neural Network (ANN) that is used for processing data with a grid-like topology such as time-series or digital images. Deep learning in general and CNNs, in particular, gained their momentum during the ImageNet competition in 2012 a CNN-based architecture, namely, AlexNet, reported a top-5 accuracy of 86.3%, which was remarkably better than the runner up accuracy.

Each CNN layer convolves the multidimensional array of input by a *kernel*, which is a multidimensional array of parameters (tensors) learned by the learning algorithm. To formalize the idea, let $\mathbf{X}$, $\mathbf{K}$, and $\mathbf{Y}$ denote the 3-D input tensor, 4-D kernel tensor, and 3-D output tensor, respectively.

Let $\mathbf{X}_{r,s,t}$ denote the element in the channel (also known as depth) r, row s and column t of the input $\mathbf{X}$ for $1 \leq r \leq c_{\mathbf{X}}$ (input channels), $1 \leq s \leq h_{\mathbf{X}}$ (input height), and $1 \leq t \leq w_{\mathbf{X}}$ (input width). For our datasets, the first CNN layer has $c_{\mathbf{X}} = 1$ (similar to having a grayscale image), $h_{\mathbf{X}} =$ number of EEG channels from Table I, $w_{\mathbf{X}} = \lfloor 256 \text{ Hz} \times (600) \text{ ms} \rfloor = 153$. We show the elements of $\mathbf{K}$ by $\mathbf{K}_{u,r,s,t}$ for $1 \leq u \leq c_{\mathbf{Y}}$ (output channels), $1 \leq r \leq c_{\mathbf{X}}$, $1 \leq s \leq h_{\mathbf{K}}$ (kernel height), and $1 \leq t \leq w_{\mathbf{K}}$ (kernel width) where $h_{\mathbf{K}} < h_{\mathbf{X}}$ and $w_{\mathbf{K}} < w_{\mathbf{X}}$. This means that $\mathbf{K}$ 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]. Convolving $\mathbf{K}$ across $\mathbf{X}$ (with no padding of the input, no subsampling, nor any bias term) is obtained by computing the output *feature map* $\mathbf{Y}$ with elements $\mathbf{Y}_{u,p,q}$ where $1 \leq u \leq c_{\mathbf{Y}}$, $1 \leq p \leq h_{\mathbf{Y}} = h_{\mathbf{X}} - h_{\mathbf{K}} + 1$, and $1 \leq q \leq w_{\mathbf{Y}} = w_{\mathbf{X}} - w_{\mathbf{K}} + 1$ by

$$\mathbf{Y}_{u,p,q} = \sum_{r,s,t} \mathbf{X}_{r,p+s-1,q+t-1} \mathbf{K}_{u,r,s,t} \,, \tag{1}$$

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, training a CNN means computing $c_{\mathbf{Y}}(c_{\mathbf{X}} h_{\mathbf{K}} w_{\mathbf{K}} + 1)$ parameters for each layer using the back-propagation algorithm [61].

D. Data Upsampling

Assuming a random sampling, when the prevalence (prior probabilities) of a target and non-target classes in a dataset is

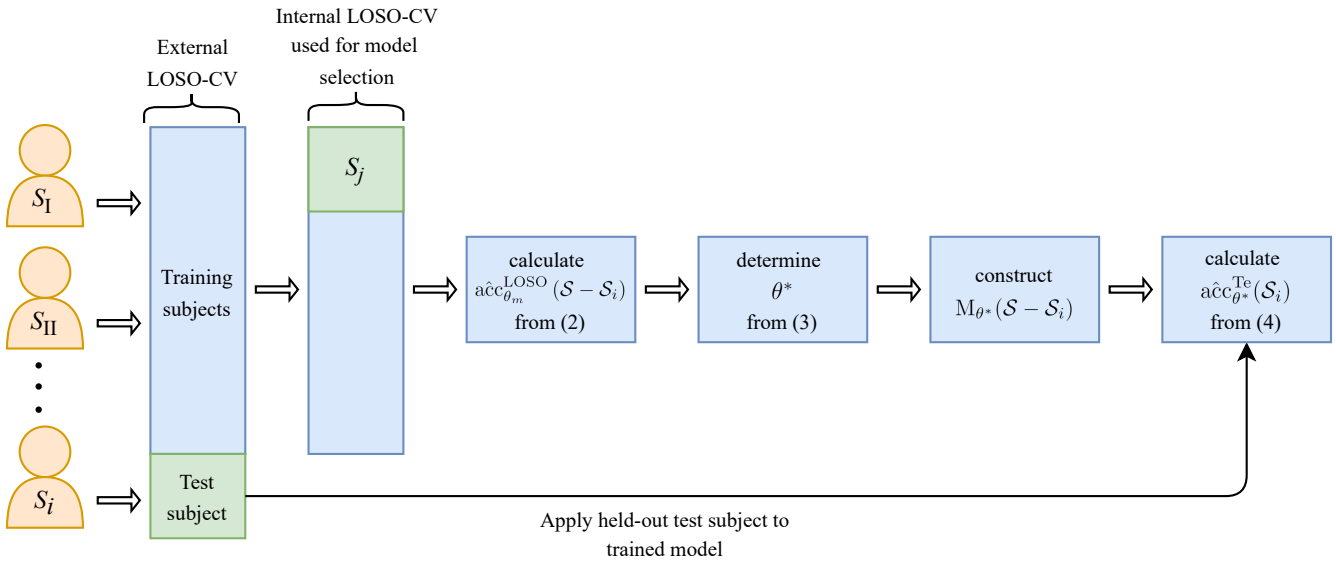

Fig. 2. A schematic representation of the nested leave-one-subject-out cross-validation (LOSO-CV) scheme used for model selection, training, and validation of subject-independent classifiers. Once $\mathcal{S}_i$ is held out as part of the external LOSO-CV, in the internal LOSO-CV, the remaining subjects ($\mathcal{S}_j, j \in S-\mathcal{S}_i$) are held out one at a time. At the end of this internal LOSO-CV, $\hat{\text{acc}}_{\theta_m}^{\text{LOSO}}(\mathcal{S}-\mathcal{S}_i)$ is calculated from (2). Then θ^*, which is the estimate of the optimal combination of hyperparameters on $\mathcal{S}-\mathcal{S}_i$, is obtained from (3) and the performance of $\text{M}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$ is assessed on $\mathcal{S}_i$, which is the held-out subject.

Algorithm 1 Nested LOSO-CV Process

Require: subjects $\mathcal{S}=\bigcup_{i=\text{I}}^{K}\mathcal{S}_i$
$\mathbf{x}_{l,i}, l=1,...,n_i$
$i \in \Omega \triangleq \{\text{I, II, ...}, K\}$, $\mathcal{S}_i=\bigcup_{l=1}^{n_i}\mathbf{x}_{l,i}$
Set of hyperparameters Θ
for each $\mathcal{S}_i \in \mathcal{S}$ **do**
  for each $\theta_m \in \Theta$ **do**
    for each $\mathcal{S}_j \in \mathcal{S}-\mathcal{S}_i$ **do**
      construct $\text{M}_{\theta_m}(\mathcal{S}-\mathcal{S}_i-\mathcal{S}_j)$
    end for
    calculate $\hat{\text{acc}}_{\theta_m}^{\text{LOSO}}(\mathcal{S}-\mathcal{S}_i)$ from (2)
  end for
  determine θ^* from (3)
  construct $\text{M}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$
  calculate $\hat{\text{acc}}_{\theta^*}^{\text{Te}}(\mathcal{S}_i)$ from (4)
end for

different (i.e., class imbalance problem), we may use various data augmentation techniques to improve the classification performance [62], [63]. In this regard, two common techniques are oversampling instances of the minority class or undersampling instances of the majority class. In all our datasets (see Tables S1-S4 in the Supplementary Materials), the number of observations in the majority class (non-target) to minority class (target) is 5 : 1. Based on the conclusion of [63], we oversample the observations (after preprocessing stage) the minority class to adjust the class distribution balance. Let n_{min} and n_{maj} show the number of observations in the minority and majority classes, respectively, and let $\alpha = \frac{n_{\text{maj}}}{n_{\text{min}}}$. In the up-sampling process, we augment the original (imbalanced) dataset, by randomly choosing $n_{\text{min}}(\alpha-1)$ instances of the minority class and replicating those such that the number of observations in the minority class becomes $n_{\text{min}}(\alpha-1)+n_{\text{min}}=n_{\text{maj}}$.

E. Model Selection and Validation

TABLE II
COMBINATIONS OF NUMBER OF LAYERS, FILTER, AND KERNEL SIZE OF CNNS CONSIDERED IN THIS STUDY. THE LEFT COLUMN IS THE ARCHITECTURE IDENTIFIER THAT WILL BE USED IN NEXT TABLES. THESE COMBINATIONS ALONG WITH THE POSSIBLE NUMBER OF EPOCHS (1 TO 100) DEFINE A HYPERPARAMETER SPACE OF SIZE 900 (I.E., $|\Theta|=900$).

Arch. Identifier	$\bigcup_{k=1}^{L}\{c_{\mathbf{Y},k}\}$	$\bigcup_{k=1}^{L}\{h_{\mathbf{K},k}\}$
A.1	{32, 16}	{3, 3}
A.2	{32, 16}	{5, 5}
A.3	{32, 16}	{7, 7}
A.4	{64, 32, 16}	{3, 3, 3}
A.5	{64, 32, 16}	{5, 5, 5}
A.6	{64, 32, 16}	{7, 7, 5}
A.7	{128, 64, 32, 16}	{3, 3, 3, 3}
A.8	{128, 64, 32, 16}	{5, 5, 5, 5}
A.9	{128, 64, 32, 16}	{7, 7, 5, 5}

The datasets that are used in this study were acquired from several subjects as described in Section II-A. Let $\mathbf{x}_{l,i}, l=1,...,n_i$ denote n_i collected observations for subject $i \in \Omega \triangleq \{\text{I, II,...}, K\}$, $\mathcal{S}_i=\bigcup_{l=1}^{n_i}\mathbf{x}_{l,i}$, which is the set of observations collected for subject i, K is the number of subjects in the dataset, and $\mathcal{S}=\bigcup_{i=\text{I}}^{K}\mathcal{S}_i$ denote the set of all observations collected from all subjects in the dataset. In the process of LOSO-CV, we successively hold out $\mathcal{S}_i$, train a classifier on observations collected from remaining subjects (i.e., $\mathcal{S}-\mathcal{S}_i$), and test the performance of the constructed classifier on $\mathcal{S}_i$.

TABLE III
CROSS-VALIDATION ESTIMATE OF ACCURACY IN PERCENTAGE OBTAINED FROM (2) FOR EACH SET OF TRAINING DATA $\mathcal{S}-\mathcal{S}_i$ AND ALL ARCHITECTURES PRESENTED IN TABLE II. NEXT TO EACH ACCURACY, THE EPOCH THAT LED TO THE HIGHEST ACCURACY GIVEN THE SPECIFIED ARCHITECTURE IS PRESENTED IN PARENTHESES. $\hat{\text{acc}}^{\text{LOSO}}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$ ARE IDENTIFIED BY ACCURACIES SHOWN IN BOLD.

Arch. Identifier	**BCNI14 Data**									
	$\mathcal{S}-\mathcal{S}_{\text{I}}$	$\mathcal{S}-\mathcal{S}_{\text{II}}$	$\mathcal{S}-\mathcal{S}_{\text{III}}$	$\mathcal{S}-\mathcal{S}_{\text{IV}}$	$\mathcal{S}-\mathcal{S}_{\text{V}}$	$\mathcal{S}-\mathcal{S}_{\text{VI}}$	$\mathcal{S}-\mathcal{S}_{\text{VII}}$	$\mathcal{S}-\mathcal{S}_{\text{VIII}}$	$\mathcal{S}-\mathcal{S}_{\text{IX}}$	$\mathcal{S}-\mathcal{S}_{\text{X}}$
A.1	76.7 (83)	75.8 (39)	77.1 (54)	76.0 (99)	75.6 (56)	76.5 (39)	75.8 (45)	77.2 (75)	75.5 (46)	**76.6** (88)
A.2	76.9 (94)	75.9 (65)	77.6 (99)	76.8 (80)	76.0 (70)	**78.3** (97)	76.1 (31)	**77.6** (77)	75.7 (58)	76.1 (81)
A.3	76.8 (98)	76.0 (99)	**77.7** (72)	76.2 (99)	75.7 (56)	76.5 (31)	76.6 (81)	77.1 (53)	**76.0** (62)	76.4 (77)
A.4	**77.0** (77)	**77.0** (85)	77.1 (52)	**76.9** (84)	**76.3** (47)	77.0 (17)	**76.9** (33)	70.0 (72)	75.4 (74)	76.3 (100)
A.5	76.1 (62)	75.9 (90)	76.7 (55)	76.1 (47)	75.8 (93)	77.7 (48)	76.6 (74)	70.0 (14)	75.3 (50)	76.5 (61)
A.6	76.4 (52)	75.8 (67)	77.1 (57)	76.1 (91)	75.3 (24)	76.3 (72)	75.9 (82)	76.4 (92)	75.8 (77)	75.6 (72)
A.7	75.8 (60)	75.3 (60)	76.7 (42)	75.8 (91)	75.9 (19)	76.8 (16)	75.4 (97)	76.8 (94)	75.1 (81)	75.8 (38)
A.8	74.6 (42)	74.4 (91)	75.5 (60)	75.1 (53)	74.6 (19)	75.5 (96)	75.4 (99)	75.0 (58)	74.2 (33)	76.0 (51)
A.9	74.8 (58)	73.6 (22)	75.1 (43)	75.1 (100)	75.3 (91)	75.1 (29)	75.3 (72)	75.5 (26)	73.9 (36)	74.6 (79)
Arch. Identifier	**BNCI15 Data**									
	$\mathcal{S}-\mathcal{S}_{\text{I}}$	$\mathcal{S}-\mathcal{S}_{\text{II}}$	$\mathcal{S}-\mathcal{S}_{\text{III}}$	$\mathcal{S}-\mathcal{S}_{\text{IV}}$	$\mathcal{S}-\mathcal{S}_{\text{V}}$	$\mathcal{S}-\mathcal{S}_{\text{VI}}$	$\mathcal{S}-\mathcal{S}_{\text{VII}}$	$\mathcal{S}-\mathcal{S}_{\text{VIII}}$	$\mathcal{S}-\mathcal{S}_{\text{IX}}$	$\mathcal{S}-\mathcal{S}_{\text{X}}$
A.1	69.93 (63)	62.2 (97)	64.09 (91)	68.67 (88)	**69.03** (94)	**71.22** (98)	65.47 (99)	65.77 (66)	65.97 (71)	69.4 (68)
A.2	70.9 (79)	63.43 (66)	**69.24** (88)	67.33 (79)	68.66 (84)	68.88 (57)	**70.8** (72)	67.63 (56)	70.44 (87)	**72.23** (77)
A.3	72.63 (60)	**68.47** (93)	61.81 (81)	68.33 (68)	63.86 (91)	61.95 (75)	63.57 (89)	**70.39** (67)	64.15 (88)	63.43 (74)
A.4	71.1 (94)	63.17 (61)	66.3 (96)	68.43 (79)	66.37 (55)	67.88 (73)	63.77 (92)	66.08 (59)	**71.22** (70)	65.77 (88)
A.5	**72.73** (58)	67.23 (80)	67.48 (60)	**71.73** (89)	68.91 (91)	64.36 (81)	66.83 (88)	67.28 (60)	62.26 (68)	70.03 (70)
A.6	67.03 (92)	65.37 (55)	68.11 (74)	66.1 (84)	62.53 (97)	63.04 (74)	68.67 (92)	65.52 (89)	66.27 (92)	59.97 (56)
A.7	69.63 (62)	63.43 (68)	67.73 (84)	63.0 (90)	68.34 (95)	62.76 (77)	66.27 (75)	62.79 (89)	63.88 (86)	66.97 (81)
A.8	68.53 (67)	63.03 (86)	66.15 (90)	66.5 (85)	64.1 (73)	62.0 (78)	61.87 (89)	62.8 (58)	67.22 (84)	62.17 (92)
A.9	64.27 (74)	67.93 (83)	64.59 (82)	65.57 (77)	61.7 (93)	65.64 (61)	66.57 (71)	66.69 (97)	63.49 (66)	64.87 (91)

In this regard, once subject i is held out, the entire process of model selection and construction is conducted on $\mathcal{S}-\mathcal{S}_i$. For model selection *per se*, we perform another round of (i.e., internal) LOSO-CV as described next.

Let $\Theta=\{\theta_m, m=1,2,...,B\}$ denote the set of possible combination of hyperparameters; that is to say, each θ_m is a possible combination of number of layers, filter, and kernel size, epoch number, and B is the total number of combinations considered in the process of model selection. Furthermore, let $\text{M}_\theta(\mathcal{D})$ denote a model constructed by applying a classification rule with a fix set of hyperparemeters θ on data $\mathcal{D}$. During the internal LOSO-CV, we successively set aside $\mathcal{S}_j \in \mathcal{S}-\mathcal{S}_i, j \in \Omega-\{i\}$, and for each θ_m, we construct $\text{M}_{\theta_m}(\mathcal{S}-\mathcal{S}_i-\mathcal{S}_j)$, and then use this model to classify observations in $\mathcal{S}_j$. Given a fixed θ_m, the accuracy of the corresponding model constructed on $\mathcal{S}-\mathcal{S}_i$ $\big($i.e., $\text{M}_{\theta_m}(\mathcal{S}-\mathcal{S}_i)\big)$ used in a subject-independent context is estimated by the overall proportion of correctly classified observations by surrogate classifiers $\text{M}_{\theta_m}(\mathcal{S}-\mathcal{S}_i-\mathcal{S}_j)$ on $\mathcal{S}_j, \forall j$; that is to say, the LOSO-CV estimate of the accuracy of $\text{M}_{\theta_m}(\mathcal{S}-\mathcal{S}_i)$ is:

$$\hat{\text{acc}}^{\text{LOSO}}_{\theta_m}(\mathcal{S}-\mathcal{S}_i)=\frac{1}{|\mathcal{S}-\mathcal{S}_i|}\sum_{j\in\Omega-\{i\}}\sum_{\mathbf{x}_{l,j}} I_{\{y_{l,j}=\hat{y}_{l,j}\}}\,, \quad (2)$$

where $|S|$ is the cardinality of set S, $I_{\{A\}}$ is 1 when A is true, $y_{l,j}$ is the true label of $\mathbf{x}_{l,j}$, and $\hat{y}_{l,j}$ denotes the classified label of $\mathbf{x}_{l,j}$ by $\text{M}_{\theta_m}(\mathcal{S}-\mathcal{S}_i-\mathcal{S}_j)$, respectively. Then the estimated optimal combination of hyperparameters using $\mathcal{S}-\mathcal{S}_i$ training data denoted as θ^* is the one with the highest estimated accuracy; that is to say,

$$\theta^*=\underset{\theta_m}{\text{argmax}}\ \hat{\text{acc}}^{\text{LOSO}}_{\theta_m}(\mathcal{S}-\mathcal{S}_i)\,. \quad (3)$$

Under general assumptions on sufficiency of the sample size in $\mathcal{S}-\mathcal{S}_i$, $\hat{\text{acc}}^{\text{LOSO}}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$ provides a reasonable estimate of the performance of the $\text{M}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$, which is the model constructed using θ^* on the full $\mathcal{S}-\mathcal{S}_i$. Nevertheless, the preferred method of validating $\text{M}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$ is to apply and test it on an entirely independent dataset. Here, observations in $\mathcal{S}_i$ serve as an independent test data. Therefore, once $\text{M}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$ is constructed, it is applied to $\mathcal{S}_i$ to obtain an estimate of the accuracy of the model on test data denoted as $\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)$. In particular,

$$\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)=\frac{1}{|\mathcal{S}_i|}\sum_{\mathbf{x}_{l,i}} I_{\{y_{l,i}=\hat{y}_{l,i}\}}\,, \quad (4)$$

where $\hat{y}_{l,i}$ is the classified label of $\mathbf{x}_{l,i}$ by $\text{M}_{\theta^*}(\mathcal{S}-\mathcal{S}_i)$. Fig. 2 and Algorithm 1 represent schematic and algorithmic representations of the proposed nested LOSO-CV process, respectively.

F. The Hyperparameter Space

Assume L is the number of convolutional layer used in the model. Based on our prior experience and computational

TABLE IV
CROSS-VALIDATION ESTIMATE OF ACCURACY IN PERCENTAGE OBTAINED FROM (2) FOR EACH SET OF TRAINING DATA $\mathcal{S} - \mathcal{S}_i$ AND ALL ARCHITECTURES PRESENTED IN TABLE II. NEXT TO EACH ACCURACY, THE EPOCH THAT LED TO THE HIGHEST ACCURACY GIVEN THE SPECIFIED ARCHITECTURE IS PRESENTED IN PARENTHESES. $\hat{\text{ACC}}^{\text{LOSO}}_{\theta^*}(\mathcal{S} - \mathcal{S}_i)$ ARE IDENTIFIED BY ACCURACIES SHOWN IN BOLD.

Arch. Identifier	**ALS Data**							
	$\mathcal{S} - \mathcal{S}_{\text{I}}$	$\mathcal{S} - \mathcal{S}_{\text{II}}$	$\mathcal{S} - \mathcal{S}_{\text{III}}$	$\mathcal{S} - \mathcal{S}_{\text{IV}}$	$\mathcal{S} - \mathcal{S}_{\text{V}}$	$\mathcal{S} - \mathcal{S}_{\text{VI}}$	$\mathcal{S} - \mathcal{S}_{\text{VII}}$	$\mathcal{S} - \mathcal{S}_{\text{VIII}}$
A.1	73.89 (60)	75.2 (96)	70.99 (86)	74.07 (93)	73.94 (68)	75.04 (81)	**76.4** (84)	**76.56** (56)
A.2	**76.0** (91)	**75.76** (92)	70.51 (97)	74.89 (97)	73.64 (74)	75.01 (98)	75.13 (92)	76.01 (86)
A.3	71.49 (81)	75.16 (99)	70.91 (73)	**75.63** (81)	74.06 (99)	74.59 (97)	75.29 (86)	74.41 (67)
A.4	72.64 (76)	75.6 (71)	72.74 (68)	74.77 (73)	74.49 (96)	73.84 (59)	73.83 (89)	74.64 (66)
A.5	73.36 (98)	75.0 (74)	73.71 (81)	74.49 (94)	74.8 (99)	75.59 (70)	73.71 (79)	74.77 (89)
A.6	73.31 (72)	74.26 (86)	**73.71** (66)	73.21 (96)	**75.14** (65)	74.84 (63)	74.59 (58)	75.87 (74)
A.7	70.83 (79)	73.3 (79)	71.33 (84)	73.33 (87)	72.06 (66)	74.11 (96)	74.83 (68)	75.06 (97)
A.8	73.1 (97)	73.7 (56)	71.29 (86)	75.04 (90)	72.0 (76)	72.7 (92)	73.24 (55)	73.59 (97)
A.9	68.29 (56)	74.83 (72)	72.71 (81)	75.1 (61)	73.66 (83)	**76.13** (96)	73.59 (83)	76.01 (55)
Arch. Identifier	**EPFL Data**							
	$\mathcal{S} - \mathcal{S}_{\text{I}}$	$\mathcal{S} - \mathcal{S}_{\text{II}}$	$\mathcal{S} - \mathcal{S}_{\text{III}}$	$\mathcal{S} - \mathcal{S}_{\text{IV}}$	$\mathcal{S} - \mathcal{S}_{\text{V}}$	$\mathcal{S} - \mathcal{S}_{\text{VI}}$	$\mathcal{S} - \mathcal{S}_{\text{VII}}$	$\mathcal{S} - \mathcal{S}_{\text{VIII}}$
A.1	57.0 (94)	**70.87** (86)	58.73 (56)	64.64 (75)	59.17 (77)	**67.87** (75)	51.75 (84)	**67.35** (96)
A.2	61.13 (89)	62.22 (80)	**63.71** (84)	65.2 (96)	53.2 (89)	57.93 (57)	58.05 (78)	63.76 (66)
A.3	62.8 (81)	60.73 (91)	61.42 (72)	63.02 (82)	**62.1** (58)	50.07 (64)	**62.11** (92)	59.4 (80)
A.4	**67.82** (62)	70.55 (86)	61.55 (66)	**68.35** (84)	56.18 (77)	54.67 (72)	54.31 (94)	61.62 (93)
A.5	62.98 (96)	53.78 (95)	53.0 (92)	60.51 (91)	50.86 (99)	50.0 (66)	49.98 (90)	58.22 (58)
A.6	59.02 (98)	59.07 (81)	50.04 (64)	60.96 (98)	53.05 (97)	52.47 (66)	53.36 (77)	53.55 (85)
A.7	66.71 (66)	59.2 (96)	55.04 (56)	65.11 (70)	55.98 (97)	57.4 (63)	60.98 (79)	56.29 (81)
A.8	62.33 (98)	55.0 (82)	49.93 (99)	59.33 (97)	58.65 (67)	54.84 (97)	60.24 (78)	53.89 (62)
A.9	53.45 (92)	53.85 (85)	49.58 (68)	64.25 (73)	54.27 (85)	51.4 (79)	53.04 (57)	52.65 (94)

capacity, we made the following assumptions to restrict the possible search space for candidate models (i.e., to determine $|\Theta|$): (i) $L = 2, 3, 4$; (ii) the number of output channels (depth of the feature maps) for the k^{th} convolutional layer ($k = 1, ..., L$) is $c_{\mathbf{Y},k} = 2^{4+L-k}$ (Supplementary Material Section II provides a detail discussion on the proposed pattern for the number of output channels); (iii) at each layer k, the height and width of kernels denoted by $h_{\mathbf{K},k}$ and $w_{\mathbf{K},k}$, respectively, are equal (i.e., square shape kernels); (iv) the output of the last convolutional layer is flattened (vectorized) and is then treated as the input to two consecutive fully connected layers with ReLU activation function after the first fully connected layer of 256 neurons followed by an output layer of 2 neurons for each class; (v) after each convolutional layer, there is a ReLU function followed by a max pooling layer with filter size 1×2 and no stride; (vi) the maximum number of epochs for training is 100; (vii) the batch size is 32; (viii) the dropout rate is set to 10% [64]; (ix) the loss functions is Cross-entropy [65]; and (x) the optimizer is Adam with a learning rate of 0.001 and a decay of 0.0001 [66]. With these assumptions,$|\Theta| \equiv B = 900$ (100 epochs $\times$ 9 combination of layer, filter, and kernel size presented in Table II). Detailed specification about the structure of these 9 architectures are presented in Tables S6-S14 in Supplementary Materials.

G. Comparison with other Deep Learning Models

As a benchmark for comparison, we consider two well-known CNN-based deep learning architectures known as the EEGNet [42] and ShallowConvNet [67] that are trained using the common strategy of "pooling" data from training subjects and performing model selection on a held-out proportion of the pooled data. In other words, 80% of the pooled data was used as training and 20% as validation sets. The network training hyperparameters such as the learning rate, optimizer, batch size, and epochs were the same as described in Section II-F. These models have shown the state-of-the-art performance in various EEG decoding tasks [37], [68]. The EEGNet architecture is based on three convolutional layers with the following properties: (i) the first layer uses a temporal convolution; (ii) the second layer uses a depthwise convolution; and (iii) the third convolutional layer uses the combination of a depthwise convolution and a pointwise convolution. Table VI lists specification of the EEGNet architecture 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. The output size of each operation in this table is shown in the third column as a 3-tuple $[c_{\mathbf{Y}}, h_{\mathbf{Y}}, w_{\mathbf{Y}}]$. The ShallowConvNet architecture consists of two convolutional layers. The first

layer uses temporal convolutions to learn temporal features of EEG, while the second layer tries to learn spatial patterns using convolution. The ShallowConvNet utilizes squaring nonlinearity, an average pooling layer, and a log nonlinearity between layers. Table VII lists the details of the network architecture. The ShallowConvNet architecture tries to replicate the Filter Bank Common Spatial Patterns (FBCSP) [69], [70] structure which has won several BCI competitions, including various ERP-based classification tasks, as reported in [42].

The entire workflow was implemented in PyTorch (version 1.8.1) deep learning environment with CUDAlibrary version 10.2 on a Linux workstation with Intel Corei9-9900K, (3.6 GHz) processor, 32 GB of RAM, and NvidiaGeForce RTX 2080 Ti (RAM = 11GB, CUDA Cores: 4352).

III. Results

The estimated cross-validation accuracies are listed in Table III (BNCI14, BNCI15 datasets) and Table IV (ALS, EPFL datasets). These results were obtained using the proposed nested LOSO-CV strategy summarized in Algorithm 1. In these tables, the epoch that led to the highest accuracy (in percentage) given the specified architecture is presented in parentheses next to each accuracy. Furthermore, the bold entries in these tables show the cross-validation accuracies of the estimated optimal combination of hyperparameters θ^* on $\mathcal{S} - \mathcal{S}_i$ training data.

Table V shows the estimated test and cross-validation accuracies for the models with the optimal hyperparameter set assessed on independent held-out subject data. In this table, θ^* is represented as a tuple with elements being an architecture identified from Tables III and IV, and an epoch number. Also note that the cross-validation accuracies match accuracies given in bold in Tables III and IV. Therefore, for each $\mathcal{S} - \mathcal{S}_i$ set of training data, one can determine θ^* by taking the architecture and the epoch number associated with the accuracy identified in bold for that set of training data. For the ease of readers, these optimal combinations are represented for each dataset as one row in Table V.

The test accuracy $\big($i.e., $\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)\big)$ of each constructed model on unseen held-out subject data is presented in every third rows of Table V for each dataset. In Table V, we also show the test performance of the EEGNet and ShallowConvNet architectures as a benchmark for comparison with the performance of the trained CNN using θ^*.

These results show that the models selected via LOSO-CV approach outperforms both EEGNet and ShallowConvNet in subject-independent decoding of P300 waveforms in almost all datasets in terms of the average test accuracies (in terms of Information Transfer Rate (ITR), which are presented in Supplementary Materials Section IV, the same conclusion is reached). These differences in test performances of these models are also visualized in the box plots shown in Fig. 3.

We also note that the majority of CNN models selected through the cross-validation have simpler architecture, as shown in Tables III and IV—in most cases, the better models have two or three convolutional layers. Such simpler models that perform better are generally expected since the model selection is conducted on a small number of training subjects. To explore the potential utility of more complex CNNs, one could increase the training sample size and perform the systematic model selection using the proposed LOSO-CV method. Having more data could guard against overfitting and possibly lead to a better generalization on un-seen data.

IV. Discussion

The enormous variability present in EEG data impairs the performance of most BCI speller systems. In particular, ERP data distribution can vary for a subject within a single session, between sessions, between online and offline settings, or from trial-to-trial (i.e., intra-subject variability). These variabilities in data can cause the previously functioning BCIs to fail outright. Moreover, the inter-subject variability in EEG data still remains one of the main challenges in designing practical BCIs. As a result, many studies have focused on subject-specific applications, with the results of subject-independent prediction being much scarcer in the literature [18], [37], which is what concerns us here in this study.

The focus of our research to construct subject-agnostic (independent) classifiers for BCIs is warranted because having such classifiers in place could potentially prevent the time-consuming calibration time that is typically required for new BCI users. Nevertheless, machine learning techniques that could achieve this goal are categorized as methods for calibration time reduction or methods for calibration time suppression (or zero-calibration methods) [71]. Our study here is an effort towards constructing *zero-calibration classifiers*; that is to say, to use a set of training subjects to construct classifiers that are able to immediately classify observations collected on unseen test subjects with no further tuning. In this regard, the process of nested LOSO-CV is utilized for model selection and validation. In particular, the internal LOSO-CV is used for CNN model selection (hyperparameter tuning) and the external LOSO-CV is used for assessing the entire process of training and model selection. As can be seen in Table V, the performances of constructed classifiers manifest the capability of the utilized systematic LOSO-CV model selection with CNN to construct zero-calibration BCIs with reasonable performances.

Despite the fact that the recent advances in deep learning have introduced a plethora of neural network architectures in the BCI field [37], [68], [72], we note that the majority of the studies still focus on subject-specific classification or use part of data from a target user to update a pre-trained classifier (see Introduction). Moreover, almost all deep learning studies in the BCI field use conventional model training and selection routines—for example, pooling data from a number of subjects (or sessions) to construct a classifier, which is used/updated for new users (or in new sessions). Among other investigated neural network architectures, the EEGNet and ShallowConvNet have shown remarkable performances and have been used as a benchmark in many BCI studies. In this regard, we chose these two well-known CNN-based deep learning architectures for comparison (see Section II-G.) We have shown that conventional CNN architectures optimized

TABLE V

THE ESTIMATED CROSS-VALIDATION AND TEST ACCURACIES (IN PERCENTAGE) FOR THE CNN MODELS CONSTRUCTED ON $\mathcal{S} - \mathcal{S}_i$ WITH OPTIMAL HYPERPARAMETER SET θ^* ON FOUR DIFFERENT DATASETS. THE CROSS-VALIDATION ACCURACIES MATCH ACCURACIES GIVEN IN BOLD IN TABLES III, IV. THE θ^* IS REPRESENTED AS A TUPLE WITH AN ARCHITECTURE IDENTIFIED FROM TABLE II AND AN EPOCH NUMBER. THE TEST PERFORMANCE OF THE EEGNET AND SHALLOWCONVNET ARCHITECTURES ARE SHOWN AS A BENCHMARK COMPARISON WITH $\hat{\text{acc}}^{\text{TE}}_{\theta^*}(\mathcal{S}_i)$ FOR EACH SEPARATE DATA.

Data	Models	$\mathcal{S}_{\text{I}}$	$\mathcal{S}_{\text{II}}$	$\mathcal{S}_{\text{III}}$	$\mathcal{S}_{\text{IV}}$	$\mathcal{S}_{\text{V}}$	$\mathcal{S}_{\text{VI}}$	$\mathcal{S}_{\text{VII}}$	$\mathcal{S}_{\text{VIII}}$	$\mathcal{S}_{\text{IX}}$	$\mathcal{S}_{\text{X}}$	Average
BNCI14	θ^*	(A.4, 77)	(A.4, 85)	(A.3, 72)	(A.4, 84)	(A.4, 47)	(A.2, 97)	(A.4, 33)	(A.2, 77)	(A.3, 62)	(A.1, 88)	-
	$\hat{\text{acc}}^{\text{LOSO}}_{\theta^*}(\mathcal{S} - \mathcal{S}_i)$	77.00	77.00	77.70	76.90	76.30	78.30	76.90	77.60	76.00	76.60	-
	$\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)$	67.80	77.90	75.60	81.90	75.10	71.70	73.40	73.80	82.30	78.50	**75.80**
	EEGNet	65.96	78.60	74.01	67.87	64.81	54.18	71.16	65.83	69.67	73.30	68.54
	ShallowConvNet	72.04	76.59	71.02	71.97	81.58	69.83	60.96	67.18	83.11	80.02	73.43
BNCI15	θ^*	(A.5, 63)	(A.3, 86)	(A.2, 89)	(A.5, 57)	(A.1, 86)	(A.1, 62)	(A.2, 93)	(A.3, 58)	(A.4, 56)	(A.2, 80)	-
	$\hat{\text{acc}}^{\text{LOSO}}_{\theta^*}(\mathcal{S} - \mathcal{S}_i)$	72.73	68.47	69.24	71.73	69.03	71.22	70.8	70.39	71.22	72.23	-
	$\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)$	57.36	66.04	52.65	65.03	59.48	54.02	64.53	65.12	58.58	67.19	**61.00**
	EEGNet	49.98	52.95	49.09	54.48	51.55	49.44	48.27	49.09	50.73	50.91	50.65
	ShallowConvNet	50.02	49.98	49.09	51.01	54.29	49.09	50.33	50.91	50.94	51.07	50.67
ALS	θ^*	(A.2, 82)	(A.2, 87)	(A.5, 55)	(A.6, 85)	(A.3, 63)	(A.6, 62)	(A.9, 91)	(A.1, 97)	-	-	-
	$\hat{\text{acc}}^{\text{LOSO}}_{\theta^*}(\mathcal{S} - \mathcal{S}_i)$	76.0	75.76	73.71	73.71	75.63	75.14	76.13	76.4	-	-	-
	$\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)$	68.47	67.24	72.64	69.73	71.22	65.94	70.15	70.41	-	-	**69.47**
	EEGNet	56.86	64.89	57.42	52.62	64.42	59.31	72.69	51.02	-	-	59.9
	ShallowConvNet	67.76	63.35	76.04	68.0	60.79	67.42	70.14	57.09	-	-	66.33
EPFL	θ^*	(A.4, 63)	(A.1, 61)	(A.2, 69)	(A.4, 65)	(A.3, 79)	(A.1, 87)	(A.3, 66)	(A.1, 55)	-	-	-
	$\hat{\text{acc}}^{\text{LOSO}}_{\theta^*}(\mathcal{S} - \mathcal{S}_i)$	67.82	70.87	63.71	68.35	62.1	67.87	62.11	67.35	-	-	-
	$\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)$	66.85	61.92	61.18	47.60	59.18	61.57	58.57	58.57	-	-	**59.43**
	EEGNet	50.74	49.68	49.7	49.85	50.11	61.53	49.57	49.56	-	-	51.34
	ShallowConvNet	51.37	52.36	56.6	57.54	64.15	63.32	64.27	51.14	-	-	57.59

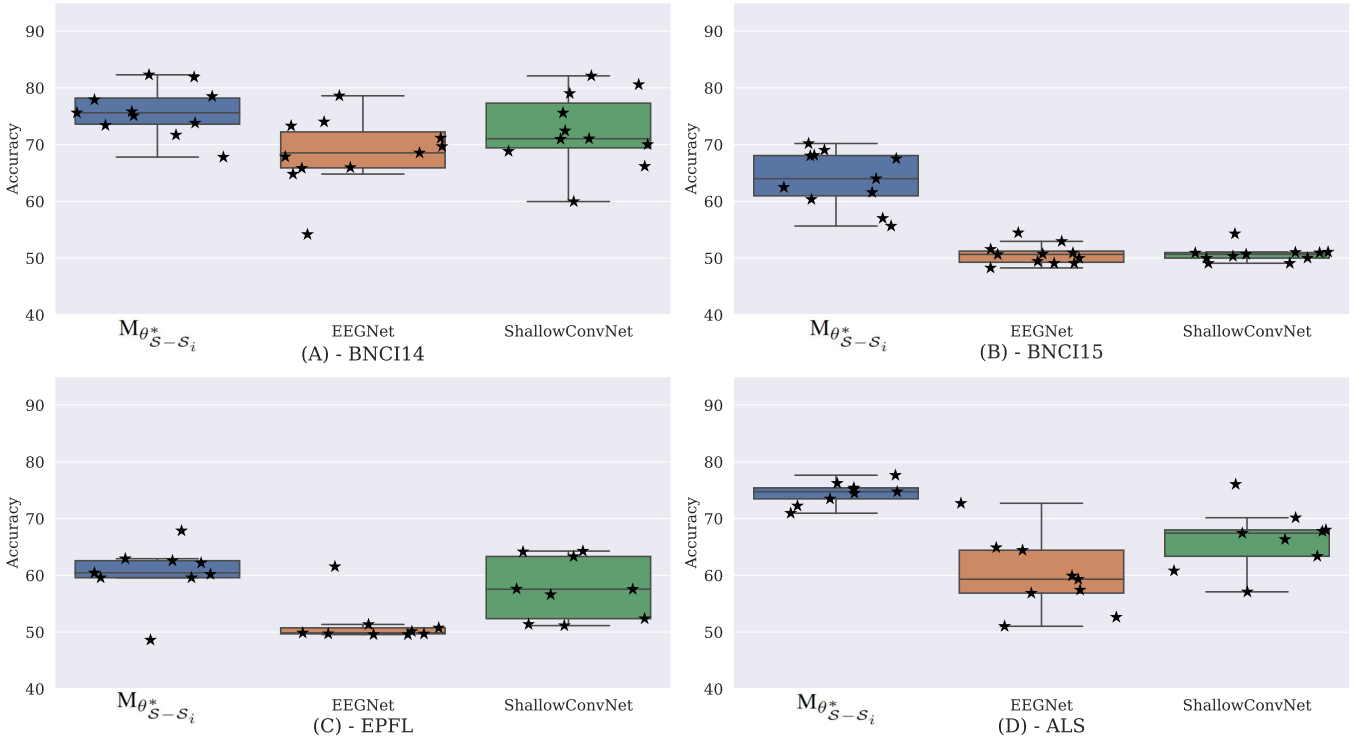

Fig. 3. The boxplot view of the estimated $\hat{\text{acc}}^{\text{Te}}_{\theta^*}(\mathcal{S}_i)$ accuracies of $\text{M}_{\theta^*}(\mathcal{S} - \mathcal{S}_i)$, EEGNet, and ShallowConvNet models on held-out subjects from four different datasets denoted as A) BNCI14 data (10 - subjects), B) BNCI15 data (10 - subjects), C) EPFL data (8 - subjects), and D) ALS data (8 - subjects). The strip plot corresponds to the performances of each model on held-out test subjects data.

(i.e., tuning hyperparameters) via a LOSO-CV model selection conducted over a limited space of hyperparameters can potentially outperform these state-of-the-art models in the subject independent classification of ERP data across all four datasets used in our study (see Fig. 3).

V. Conclusions

This investigation shows that given a small number of subjects available for training, performing a systematic model selection via the leave-one-subject out cross-validation (LOSO-CV) and using standard convolutional neural networks (CNNs) could potentially lead to relatively accurate classifiers for subject-independent P300-based BCI applications. Standard CNN has been used as the model of choice because its high efficiency in terms of training time and effectiveness in various problems have collectively made it as one of the most attractive deep learning architectures today. Using four datasets with a limited number of subjects, we used LOSO-CV for both model selection (hyperparameter tuning), and assessment. In the external loop of the LOSO-CV, we successively hold-out one subject from training, apply model selection and construction on the remaining subjects and test the accuracy of the constructed classifier on the held-out subject. The effect of the inner LOSO-CV is to simulate the subject-independent context within the training data (at most nine subjects in our datasets) and estimate the best set of hyperparameters to be used in training the CNN architecture.

Appendix: The structure of EEGNet and ShallowConvNet

This section provides the details of EEGNet architecture in Table VI, and the ShalloConvNet architecture in Table VII. These models were used as benchmark for comparison.

TABLE VI
Specification of EEGNet architecture used for decoding subject-independent P300 data.

Index	Operation	Output Size	Kernel
1	Conv2d (temporal)	[8, 16, 77]	(1, 64)
2	BatchNorm2d	[8, 16, 77]	
3	Conv2d (depthwise)	[16, 1, 77]	(16, 1)
4	BatchNorm2d	[16, 1, 77]	
5	ELU	[16, 1, 77]	
6	AvgPool2d	[16, 1, 19]	
7	Dropout	[16, 1, 19]	
8	Conv2d (seperable)	[16, 1, 20]	(1, 1)
9	BatchNorm2d	[16, 1, 20]	
10	ELU	[16, 1, 20]	
11	AvgPool2d	[16, 1, 2]	
12	Dropout	[16, 1, 2]	(p = 0.25)
13	Linear	[2]	

TABLE VII
The details of the ShallowConvNet architecture used for decoding subject-independent P300 data.

Index	Operation	Output Size	Kernel
1	Conv2d (temporal)	[15, 52, 16]	(25, 1)
2	Conv2d (spatial)	[20, 52, 1]	(1, 16)
3	BatchNorm2d	[20, 52, 1]	
4	Squaring	[20, 52, 1]	
5	AvgPool2d	[20, 3, 1]	(30, 1)
6	Log	[20, 3, 1]	
7	Dropout	[20, 3, 1]	(p = 0.25)
8	Conv2d	[2, 1, 1]	(3, 1)
9	Linear	[2]	

References

[1] 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 *et al.*, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–375, May 2012.

[2] A. Venkatakrishnan, G. E. Francisco, and J. L. Contreras-Vidal, "Applications of brain–machine interface systems in stroke recovery and rehabilitation," *Current Physical Medicine and Rehabilitation Reports*, vol. 2, no. 2, pp. 93–105, Jun. 2014.

[3] Z. Lin, C. Zhang, Y. Zeng, L. Tong, and B. Yan, "A novel p300 bci speller based on the triple rsvp paradigm," *Scientific reports*, vol. 8, no. 1, pp. 1–9, 2018.

[4] B. He, S. Gao, H. Yuan, and J. R. Wolpaw, "Brain–computer interfaces," in *Neural Engineering*. Springer, 2013, pp. 87–151.

[5] M. C. Kiernan, S. Vucic, B. C. Cheah, M. R. Turner, A. Eisen, O. Hardiman, J. R. Burrell, and M. C. Zoing, "Amyotrophic lateral sclerosis," *The Lancet*, vol. 377, no. 9769, pp. 942–955, 2011.

[6] J. Morris, "Amyotrophic lateral sclerosis (ALS) and related motor neuron diseases: an overview," *The Neurodiagnostic Journal*, vol. 55, no. 3, pp. 180–194, 2015.

[7] U. Chaudhary, N. Birbaumer, and A. Ramos-Murguialday, "Brain–computer interfaces in the completely locked-in state and chronic stroke," *Progress in Brain Research*, vol. 228, pp. 131–161, 2016.

[8] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, Jun. 2002.

[9] A. Vallabhaneni, T. Wang, and B. He, "Brain– computer interface," in *Neural Engineering*. Springer, 2005, pp. 85–121.

[10] N. Birbaumer and L. G. Cohen, "Brain–computer interfaces: communication and restoration of movement in paralysis," *The Journal of physiology*, vol. 579, no. 3, pp. 621–636, Jan. 2007.

[11] J. Wolpaw and E. W. Wolpaw, *Brain-computer interfaces: principles and practice*. OUP USA, 2012.

[12] K.-R. Müller, M. Krauledat, G. Dornhege, G. Curio, and B. Blankertz, "Machine learning techniques for brain-computer interfaces," *Biomedical Technologies*, vol. 49, no. 1, pp. 11–22, Dec. 2004.

[13] 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," *Journal of Neural engineering*, vol. 4, no. 2, p. R32, Mar. 2007.

[14] N. Kaongoen, M. Yu, and S. Jo, "Two-factor authentication system using p300 response to a sequence of human photographs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[15] J. Jin, Z. Chen, R. Xu, Y. Miao, X. Wang, and T.-P. Jung, "Developing a novel tactile p300 brain-computer interface with a cheeks-stim paradigm," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 9, pp. 2585–2593, 2020.

[16] J. Jin, S. Li, I. Daly, Y. Miao, C. Liu, X. Wang, and A. Cichocki, "The study of generic model set for reducing calibration time in p300-based brain–computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 3–12, Jan. 2020.

[17] Z. Gao, W. Dang, M. Liu, W. Guo, K. Ma, and G. Chen, "Classification of eeg signals on vep-based bci systems with broad learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–9, 2020.

[18] 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," *Journal of neural engineering*, vol. 15, no. 3, p. 031005, 2018.

[19] R. Fazel-Rezai and W. Ahmad, *P300-based brain-computer interface paradigm design*. INTECH Open Access Publisher, 2011.

[20] B. Abibullaev and A. Zollanvari, "Learning discriminative spatiospectral features of ERPs for accurate brain-computer interfaces," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 5, pp. 2009–2020, 2019.

[21] B. Abibullaev and A. Zollanvari, "A systematic deep learning model selection for p300-based brain-computer interfaces," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2021.

[22] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.

[23] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components–a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[24] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009.

[25] L. Roijendijk, "Variability and nonstationarity in brain computer interfaces," Unpublished master's thesis, Radboud University Nijmegen, 2009.

[26] A. Kübler, "Quo vadis P300 BCI," in *Brain-Computer Interface (BCI), 2017 5th International Winter Conference on*. IEEE, 2017, pp. 36–39.

[27] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr, "Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection," *IEEE Trans. on Biomed. Eng.*, vol. 55, no. 2, pp. 512–518, Feb. 2008.

[28] S. Ghosh-Dastidar and H. Adeli, "A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection," *Neural Networks*, vol. 22, no. 10, pp. 1419–1431, Dec. 2009.

[29] L. Chisci, A. Mavino, G. Perferi, M. Sciandrone, C. Anile, G. Colicchio, and F. Fuggetta, "Real-time epileptic seizure prediction using AR models and support vector machines," *IEEE Trans. on Biomed. Eng.*, vol. 57, no. 5, pp. 1124–1132, May 2010.

[30] U. R. Acharya, S. V. Sree, S. Chattopadhyay, W. Yu, and P. C. A. Ang, "Application of recurrence quantification analysis for the automated identification of epileptic EEG signals," *International journal of neural systems*, vol. 21, no. 03, pp. 199–211, Jun. 2011.

[31] U. R. Acharya, S. V. Sree, P. C. A. Ang, R. Yanti, and J. S. Suri, "Application of non-linear and wavelet based features for the automated identification of epileptic EEG signals," *International journal of neural systems*, vol. 22, no. 02, Apr. 2012.

[32] D. P. Subha, P. K. Joseph, R. Acharya, and C. M. Lim, "EEG signal analysis: A survey," *Journal of medical systems*, vol. 34, no. 2, pp. 195–212, Apr. 2010.

[33] M. A. Rahman, W. Ma, D. Tran, and J. Campbell, "A comprehensive survey of the feature extraction methods in the EEG research," in *Algorithms and Architectures for Parallel Processing*. Springer, 2012, pp. 274–283.

[34] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, B. Arnaldi *et al.*, "A review of classification algorithms for EEG-based brain–computer interfaces," *Journal of neural engineering*, vol. 4, Jun. 2007.

[35] S. Mason, A. Bashashati, M. Fatourechi, K. Navarro, and G. Birch, "A comprehensive survey of brain interface technology designs," *Annals of biomedical engineering*, vol. 35, no. 2, pp. 137–169, Nov. 2007.

[36] S. Machado, F. Araújo, F. Paes, B. Velasques, M. Cunha, H. Budde, L. F. Basile, R. Anghinah, O. Arias-Carrión, M. Cagy *et al.*, "EEG-based brain-computer interfaces: an overview of basic concepts and clinical applications in neurorehabilitation," *Reviews in the Neurosciences*, vol. 21, no. 6, pp. 451–468, Dec. 2010.

[37] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: a systematic review," *Journal of neural engineering*, vol. 16, no. 5, p. 051001, 2019.

[38] H. Cecotti and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 433–445, 2011.

[39] J. Yoon, J. Lee, and M. Whang, "Spatial and time domain feature of ERP speller system extracted via convolutional neural network," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[40] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on batch normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

[41] R. Maddula, J. Stivers, M. Mousavi, S. Ravindran, and V. de Sa, "Deep recurrent convolutional neural networks for classifying P300 BCI signals," in *Proceedings of the Graz BCI Conference*, 2017.

[42] 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," *Journal of neural engineering*, vol. 15, no. 5, 2018.

[43] S. Kundu and S. Ari, "MsCNN: A deep learning framework forp300-based brain-computer interface speller," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 1, pp. 86–93, 2020.

[44] H. L. J. Hou, Y. Li and S. Wang, "Improving the p300-based brain-computer interface with transfer learning," in *In Proceedings of the 2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2017, pp. 485–488.

[45] F. D. J. Adair, A. Brownlee and G. Ochoa, "Evolving training sets for improved transfer learning in brain computer interface," in *In International Workshop on Machine Learning, Optimization, and Big Data; Springer: Cham, Switzerland*, 2017, pp. 186–197.

[46] A. Onishi and S. Nakagawa, "Comparison of classifiers for the transfer learning of affective auditory p300-based bcis," in *In Proceedings of the 2019 41st Annual International Conference of the IEEE Engi-neering in Medicine and Biology Society (EMBC)*, 2019, pp. 6766–6769.

[47] K. Vo, T. Pham, D. N. Nguyen, H. H. Kha, and E. Dutkiewicz, "Subject-independent erp-based brain-computer interfaces," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 26, no. 4, April 2018.

[48] P.-J. Kindermans, M. Tangermann, K.-R. Müller, and B. Schrauwen, "Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller," *Journal of Neural Engineering*, vol. 11, no. 3, p. 035005, may 2014.

[49] N. R. Waytowich, V. J. Lawhern, A. W. Bohannon, K. R. Ball, and B. J. Lance, "Spectral transfer learning using information geometry for a user-independent brain-computer interface," *Frontiers in Neuroscience (Online)*, vol. 10, 9 2016.

[50] O.-Y. Kwon, M.-H. Lee, C. Guan, and S.-W. Lee, "Subject-independent brain?computer interfaces based on deep convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 3839–3852, 2019.

[51] K. Zhang, N. Robinson, S.-W. Lee, and C. Guan, "Adaptive transfer learning for eeg motor imagery classification with deep convolutional neural network," *Neural networks : the official journal of the International Neural Network Society*, vol. 136, pp. 1–10, April 2021.

[52] P. Aricò, F. Aloise, F. Schettini, S. Salinari, D. Mattia, and F. Cincotti, "Influence of p300 latency jitter on event related potential-based brain–computer interface performance," *Journal of Neural Engineering*, vol. 11, no. 3, p. 035008, May 2014. [Online]. Available: https://doi.org/10.1088/1741-2560/11/3/035008

[53] F. Aloise, P. Aricò, F. Schettini, A. Riccio, S. Salinari, and *et al.*, "A covert attention P300-based brain–computer interface: Geospell," *Ergonomics*, vol. 55, no. 5, pp. 538–551, 2012.

[54] C. Guger, S. Daban, E. Sellers, C. Holzner, G. Krausz, R. Carabalona, F. Gramatica, and G. Edlinger, "How many people are able to control a P300-based brain–computer interface (BCI)?" *Neuroscience letters*, vol. 462, no. 1, pp. 94–98, 2009.

[55] V. Jayaram and A. Barachant, "MOABB: trustworthy algorithm benchmarking for BCIs," *Journal of neural engineering*, vol. 15, no. 6, p. 066011, 2018.

[56] A. Riccio, L. Simione, F. Schettini, A. Pizzimenti, M. Inghilleri, and *et al.*, "Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis," *Frontiers in human neuroscience*, vol. 7, p. 732, 2013.

[57] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, "An efficient P300-based brain–computer interface for disabled subjects," *Journal of Neuroscience methods*, vol. 167, no. 1, pp. 115–125, 2008.

[58] D. Talsma and M. G. Woldorff, "6 methods for the estimation and removal of artifacts and overlap," *Event-related potentials: A methods handbook*, p. 115, 2005.

[59] J. Farquhar and N. J. Hill, "Interactions between pre-processing and classification methods for event-related-potential classification," *Neuroinformatics*, vol. 11, no. 2, pp. 175–192, 2013.

[60] Y. LeCun, "Generalization and network design strategies," *Technical Report, CRG-TR-89-4*, 1989.

[61] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining Practical Machine Learning Tools and Techniques*, 4th ed. Elsevier, 2017.

[62] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowledge and Data Engineering*, vol. 21, pp. 1263–1284, 2009.
[63] M. Buda, A. Maki, and M. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks : the official journal of the International Neural Network Society*, vol. 106, pp. 249–259, 2018.
[64] 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, Jan. 2014.
[65] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." *CoRR*, vol. abs/1412.6980, 2014.
[67] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.
[68] A. Craik, Y. He, and J. L. P. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *Journal of neural engineering*, 2019.
[69] 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 in neuroscience*, vol. 6, p. 39, 2012.
[70] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) in brain-computer interface," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 2390–2397.
[71] F. Lotte, "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces," *Proceedings of the IEEE*, vol. 103, pp. 871–890, 2015.
[72] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. McAlpine, and Y. Zhang, "A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers," *Journal of Neural Engineering*, vol. 18, no. 3, p. 031002, mar 2021. [Online]. Available: https://doi.org/10.1088/1741-2552/abc902

Berdakh Abibullaev (M' 12, SM' 21) received his 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 Daegu-Gyeongbuk Institute of Science and Technology (2010-2013) and at Samsung Medical Center, Seoul, South Korea (2013-2014). In 2014, he received the National Institute of Health postdoctoral research fellowship II to join a multi-institutional research project between the University of Houston Brain-Machine Interface Systems Team and Texas Medical Center in developing neural interfaces for rehabilitation in post-stroke patients. He is currently an Assistant Professor at Robotics Department, Nazarbayev University, Kazakhstan. His current research focuses on signal processing and machine learning algorithms for the inference problems of Brain-Computer Interfaces, and brain data analytics. He is an associate editor of IEEE Access, and PeerJ Computer Science.

Kassymzhomart Kunanbayev received his bachelor degree in Electrical and Electronic Engineering from Nazarbayev University, Nur-Sultan, Kazakhstan in 2020. He worked as a research assistant in the Data Analytics Lab at Nazarbayev University (2019-2020). He is currently a graduate student at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. His research interests include deep learning and biomedical engineering with a focus on brain imaging.

Amin Zollanvari (M' 10, SM' 19) received B.Sc. and M.Sc. degrees in electrical engineering from Shiraz University, Iran, and Ph.D. degree in electrical engineering from Texas A&M University, College Station TX, in 2010. He held a postdoctoral position at Harvard Medical School and Brigham and Women's Hospital, Boston MA (2010-2012), and then joined the Department of Statistics at Texas A&M University as an Assistant Research Scientist (2012-2014). Since 2015, he has been with Nazarbayev University, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. His research interests include machine learning, statistical signal processing, and biomedical informatics. He is a Senior Member of the IEEE and an Associate Editor of IEEE Access.