

A Unified Novel Neural Network Approach and a Prototype Hardware Implementation for Ultra-Low Power EEG Classification

Antonis Nikitakis, Konstantinos Makantasis^{1b}, Nikolaos Tampouratzis, and Ioannis Papaefstathiou^{1b}

Abstract—This paper introduces a novel electroencephalogram (EEG) data classification scheme together with its implementation in hardware using an innovative approach. The proposed scheme integrates into a single, end-to-end trainable model a spatial filtering technique and a neural network based classifier. The spatial filters, as well as, the coefficients of the neural network classifier are simultaneously estimated during training. By using different time-locked spatial filters, we introduce for the first time the notion of “attention” in EEG processing, which allows for the efficient capturing of the temporal dependencies and/or variability of the EEG sequential data. One of the most important benefits of our approach is that the proposed classifier is able to construct highly discriminative features directly from raw EEG data and, at the same time, to exploit the function approximation properties of neural networks, in order to produce highly accurate classification results. The evaluation of the proposed methodology, using public available EEG datasets, indicates that it outperforms the standard EEG classification approach based on filtering and classification as two separated steps. Moreover, we present a prototype implementation of the proposed scheme in state-of-the-art reconfigurable hardware; our novel implementation outperforms by more than one order of magnitude, in terms of power efficiency, the conventional CPU-based approaches.

Index Terms—CSP, Deep Neural Network, EEG, FPGA.

I. INTRODUCTION

BRAIN Computer Interface (BCI) technology is a real-time communication bridge between users and machines, which

Manuscript received December 12, 2018; revised April 5, 2019; accepted May 1, 2019. Date of publication May 15, 2019; date of current version July 26, 2019. This work was supported by the EuroExa (Grant Agreement 754337) project, funded by the European Union Horizon 2020 Research and Innovation Programme. This paper was recommended by Associate Editor G. Cauwenberghs. (Corresponding author: Konstantinos Makantasis.)

A. Nikitakis is with the Synelixis Solutions S.A., Chalkida 34100, Greece (e-mail: nikitakis@synelixis.com).

K. Makantasis is with the Althexis Solutions Ltd., Nicosia 1066, Cyprus, and also with the Institute of Digital Games, University of Malta, Msida MSD-2080, Malta (e-mail: km@althexis-solutions.com).

N. Tampouratzis is with the Synelixis Solutions S.A., Chalkida 34100, Greece, and also with the School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece (e-mail: ntampouratzis@synelixis.com).

I. Papaefstathiou is with the School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece, and also with the Synelixis Solutions S.A., Chalkida 34100, Greece (e-mail: ygp@ece.auth.gr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TBCAS.2019.2916981

serves as a replacement of normal neuromuscular pathways. The initial purpose of BCIs development was their exploitation in biomedical applications, such as rehabilitation from injuries, or communication tools for patients with locked-in syndromes [1]. The use of BCIs though, can be expanded to healthy users as well serving as training and self-improvement tools for concentration, stress management or act as “hands-free” input to other devices.

Although, BCIs have been successfully exploited in constrained laboratory environments, their exploitation in unconstrained everyday-life environment presents two main challenges. First, the classification algorithm should be robust to dynamically changing operation conditions, and, thus, employ fast adaptation mechanisms. Second, the underlying embedded hardware should be capable of continuously running the algorithm in an on-line fashion, at minimum energy budget. This work aims to address both challenges simultaneously, by proposing a novel method for EEG data classification.

As mentioned before, EEG data classification methods can provide quite accurate results when they operate in constrained environments, such as a laboratory. However, in unconstrained real-life scenarios the EEG data classification results cannot be considered accurate, mainly due to the non-stationary nature of EEG signals, which results in significant temporal variability in the measured response of the neurons [2]–[4]. Such variability has been shown to be related with internal brain processes, such as fatigue and attention, as well as with external ones, such as variability in stimulus properties [5].

In general, the available methods for single-trial EEG signals classification can be divided into two main categories. The first category comprises of classification methods that work directly in time or time-frequency domain, see for example the work in [6]. These methods try to maximize the separation between pattern classes by analyzing directly the amplitude of the recorded signals in an EEG time-series. The second category includes methods that employ some sort of data preprocessing, like spatial filtering, see for example the Common Spatial Pattern (CSP) [7], xDawn [8], wavelet CSP [9], and CSP combined with S transform [10] algorithms. Very recently CSP was also extended for preprocessing tensor objects of arbitrary order [11]. Then, the preprocessed EEG signals are fed into a linear or nonlinear classifier [12], which is responsible for their characterization. The data preprocessing task, usually is being conducted as a supervised learning task, in order to provide a way for enhancing the

separability of the EEG samples belonging to different classes, and, thus, facilitate the classification task.

Methods that belong to the first category can hardly cope with the low signal-to-noise ratio of EEG samples, let alone the non-stationary nature of the recorded signals. On the other hand, the preprocessing and the classification tasks, for methods of the second category, are applied separately in a “black box” fashion, i.e., there is no information flow between the preprocessing and the classification tasks, despite the fact that these are sequential. This poses several problems, such as computational complexity, difficulty in transfer of learning and adaptation from previous training sessions, and, in many cases, a high risk for over-fitting the classification model.

The proposed approach has the merits of the methods that belong to the second category, i.e., enhanced EEG samples separability, while at the same time, it unifies the preprocessing and classification task into a single step, in order to avoid their drawbacks.

A. Related Work

Most of the works that target the classification of EEG patterns are heavily relied on the exploitation of the CSP algorithm. This algorithm is a feature extraction method that automatically constructs and applies optimized spatial filters on EEG samples in order to increase between-class separability in a binary classification task [13], [14]. This is achieved by exploiting the Karhunen-Loeve expansion [15] to suppress the temporal dimension of EEG samples. Specifically, the CSP algorithm projects the temporal dimension of the EEG samples into a one-dimensional space, such that the variance of the power of the first class is maximized, while the variance of the power of the second class is minimized [13]. Although, the original CSP algorithm targets binary classification problems, variations of this algorithm have been proposed that address multiple class classification problems [16], [17].

Due to the successful application of the CSP algorithm for preprocessing EEG data, various methods have been proposed for increasing the robustness of CSP [18]–[20]. This kind of methods utilize prior knowledge (e.g., information from other subjects) and introduce assumptions (e.g., neighboring neurons tend to have similar activations and, thus, neighboring electrodes should provide similar recordings) via regularization techniques, in order to achieve robust filtering. Regularization techniques based on stationary subspace analysis have also been proposed for coping with the non-stationary nature of EEG patterns [21], [22]. The work in [23] proposes an extension of the original CSP algorithm. This method embeds time delays in the EEG signal, in order to take into account not only the spatial information, but also the temporal, through the autocorrelation of EEG channels.

All the aforementioned approaches utilize one average covariance matrix for the samples of each class, in order to maximize the between-class separability. However, due to the non-stationary nature of the EEG data these covariance matrices could change over time [22] (e.g., between training and

testing sessions), and this could result to ineffective spatial filters, which, in turn, may lead to inaccurate classification results.

The objective of CSP-based approaches is the maximization of separability among samples that belong to two different classes [24] by maximizing the average power of one class, and, at the same time, minimize the average power of the other (a formal definition of this criterion is presented in Section II-B). This objective is not directly related to the classification accuracy. To overcome this limitation, the authors of [25] integrate into a single unified model the CSP algorithm and a logistic regression classifier. Based on their approach the spatial filters, as well as, the coefficients of the logistic regression model are estimated simultaneously. Thus, spatial filters are optimized for the specific classification problem at hand. However, the logistic regression model can only produce linear decision boundaries in the input space, and, thus, it may not be able to accurately capture the statistical relationship between the input EEG signals and the desired classification output.

B. Our Contribution

Our work is motivated by the work presented in [25], which involves a logistic regression model and CSP filters which are trained at once. Our work extends the approach in [25], by proposing a deep neural net architecture, which is extended both in width and in depth, in order to increase its representation and discrimination power.

Already mentioned that the CSP algorithm maximizes the separability among samples that belong to different classes. For doing this, one average covariance matrix for each one of the available classes, given a training set, is calculated. One of the major flaws of the CSP algorithm stems exactly from this fact. As EEG is a non-stationary signal its covariance matrix changes over time (e.g., between training and testing sessions). As a result, by representing all of the EEG epochs of a class in a given training set by only one average covariance matrix can result in inaccurate spatial filtering [22]. Our work builds upon this idea, by proposing a novel neural network architecture. The proposed model aims to overcome the aforementioned drawback by utilizing a number of different CSP filters spread across the temporal domain, and then, by employing an attention mechanism, similar to those used in state-of-the-art natural language processing models [26]. The attention mechanism decides how to organize the CSP filters, and what level of significance to assign to them in a completely data-driven way. Since attention mechanisms allow for a selective assignment of the hidden state of the model at different points, we train simultaneously multiple CSP filters producing a series of hidden state representations for overlapping time sequences.

Moreover, our approach aims to overcome one more limitation of CSP, which is related to the employed fitness function. The computation of the spatial filters is based on the maximization of Rayleigh quotient, which is very sensitive to outliers and noise [27]. In contrast, to CSP, our approach derives the spatial filters by directly minimizing the misclassification error. Therefore, the derivation of the spatial filters avoids the optimisation

of the Rayleigh quotient, which is only indirectly related to the classification task, and, as mentioned before, is very sensitive to outliers.

Finally, motivated by the recent advances in hardware that permit machine learning applications to run in portable devices, this work presents a reference implementation of the proposed novel neural network architecture in reconfigurable hardware. Through this implementation we are able to demonstrate that the proposed approach can be efficiently accelerated using custom hardware achieving an outstanding efficiency in terms of computation per watt, enabling future BCI applications to be seamlessly supported by portable embedded devices. To summarize, we propose a novel modular and extensible EEG classifier that unifies numerous CSP approaches efficiently integrated within a single end-to-end trainable neural network.

The main contributions of the proposed methodology are the following;

- 1) integration of both time and spatial domain approaches in a single scheme,
- 2) introduction of a novel end-to-end trainable neural network architecture,
- 3) introduction of an innovative spatio-spectral filtering approach utilizing time-delayed inputs,
- 4) support, for the first time, of adaptability and transfer of knowledge between sessions and/or subjects by using multiple CSP filters with an attention mechanism, and
- 5) a very efficient prototype implementation in state-of-the-art reconfigurable hardware using an High Level Synthesis (HLS) approach.

The rest of the paper is organized as follows; Section II briefly describes the idea of attention mechanisms in neural networks, as well as, the modification of these mechanisms in order to be applied on the problem of EEG data classification. It further describes in more detail the CSP technique and the proposed learning architecture methodology. Section III describes in detail the training algorithm along with the associated derivatives in the backward graph. Finally in IV we present the evaluation results of the proposed scheme in public available datasets, as well as, our prototype hardware implementation.

II. PROPOSED METHODOLOGY

Before proceeding to the detailed description of our methodology, we give some notes on the notation used throughout this paper; matrices are denoted with bold uppercase letters, vectors with bold lowercase letters, and scalars with lowercase letters.

A. Attention in Neural Networks

According to [26], the following definition states what an attention mechanism is.

Definition 1: Given a model which preserves a hidden state \mathbf{h}_j at each time step j (K in total) of its input sequence I_K , an attention mechanism is able to compute a “context” vector \mathbf{v} , which is representative for the whole input sequence, and can be defined as the weighted sum of the K hidden states given by

$$\mathbf{v} = \sum_{j=1}^K a_j \mathbf{h}_j. \quad (1)$$

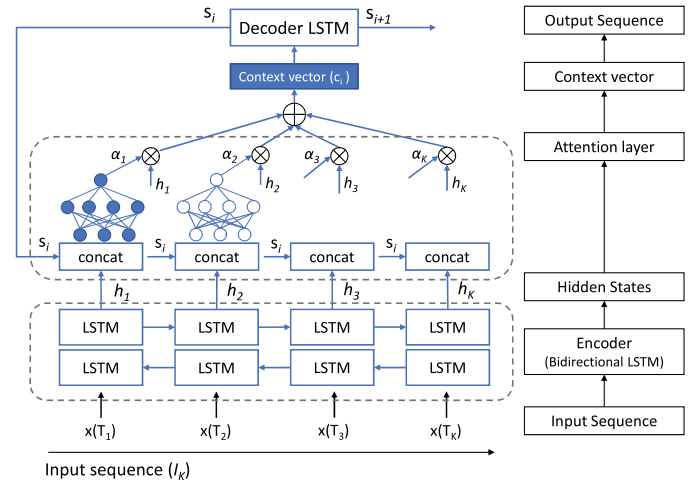


Fig. 1. A typical seq2seq architecture with attention. The encoder and decoder are usually utilized with RNN/LSTM networks.

In relation (1), variable K stands for the total number of time steps in the input sequence I_K , and a_j is a weight computed at each time step j given the state \mathbf{h}_j .

If the attention mechanism produces a sequence S_i in a number of steps (e.g., in the case of *seq2seq* models [28] in Fig. 1), instead of producing a single output, then the weights, and, as a result, the context vector depends on the current output step i of the output sequence and it is given by the following formula

$$\mathbf{v}_i = \sum_{j=1}^K a_{ij} \mathbf{h}_j. \quad (2)$$

Such models are comprised of an encoder (i.e., a bidirectional Recurrent Neural Network (RNN)) and a decoder (i.e., unidirectional RNN) bound together with an attention mechanism as depicted in Fig. 1. In such cases, for the same input sequence I_K , the attention mechanism “weights” differently the hidden states \mathbf{h}_j based on the time step i of the output sequence S_i . In practice the context vectors \mathbf{v}_i are generated from a feed forward neural network which takes as input the concatenation of \mathbf{h}_j and the current state or the output S_i . The parameters a_{ij} are estimated during the training phase of the network.

Our work presents a novel approach that utilizes, *for the very first time*, this notion of attention for classifying EEG data. The classification model proposed in this work has a single output and not a sequence of outputs. Therefore, by using a straightforward simplification to the above attention mechanism, a single vector \mathbf{v} is produced from each EEG epoch sequence using the internal hidden states \mathbf{h}_j produced by a sequence encoder. The proposed sequence encoder is formed by multiple CSP modules each one taking as input a different segment of the EEG epoch (K segments in total). The parameters of CSP modules are estimated during the training phase of the network. The proposed attention architecture, in its most general configuration, is depicted in Fig. 2. In this configuration, and, in order to simplify the design, the possible dependencies in the input sequence are not directly taken into account. However, if such a feature is needed an RNN can be added on top of the CSP modules, so

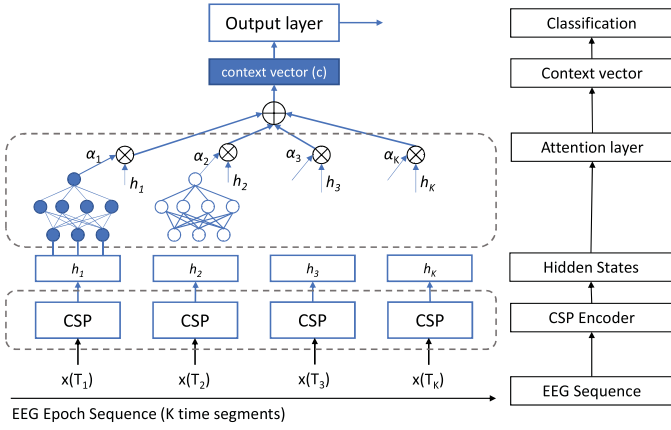


Fig. 2. Our Neural Network architecture with utilizing attention and a sequence encoder using CSP. The whole network is end-to-end differentiable.

as to perform this task. Moreover, despite the fact that our implementation is relatively simple (it does not utilize an RNN), it is able to cover certain input dependencies due to the overlap between the subsequent inputs fed into the CSP modules.

B. The CSP Technique

For the sake of clarity and completeness, this subsection briefly describes the CSP technique, which is necessary for accurately describing our proposed methodology.

EEG recordings are organized in sessions, and each session consists of several epochs. The objective of an EEG classification scheme is to classify epochs into a given number of classes. The CSP algorithm is a widely used technique for preprocessing EEG signals, in order to increase the separability between different pattern classes. Each epoch in a session should be a zero average signal, and can be represented by a matrix

$$\mathbf{X}_{c,i} \in \mathbb{R}^{N \times T}, \quad (3)$$

where superscript c denotes the class of the epoch, subscript i is the epoch number belonging to class c , N is the total number of EEG channels, and T stands for the number of samples captured by each one of the EEG channels during an epoch. In binary classification problems, the CSP technique increases the between-class separability by trying to maximize the average power of one class and, at the same time, minimize the average power of the other. The average power of class c can be calculated as follows:

$$\bar{P}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} \mathbf{w}^\top \mathbf{X}_{c,i} \mathbf{X}_{c,i}^\top \mathbf{w}, \quad (4)$$

where vector $\mathbf{w} \in \mathbb{R}^N$ projects $\mathbf{X}_{c,i}$ into an T -dimensional space and n_c is the number of epochs that belong to the c -th class. Let

$$\mathbf{R}_{c,i} = \frac{\mathbf{X}_{c,i} \mathbf{X}_{c,i}^\top}{\text{tr}(\mathbf{X}_{c,i} \mathbf{X}_{c,i}^\top)} \text{ and } \bar{\mathbf{R}}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} \mathbf{R}_{c,i} \quad (5)$$

be the covariance matrix of epoch $\mathbf{X}_{c,i}$ and the average covariance matrix of class c , respectively. Then relation (4) can be rewritten as

$$\bar{P}_c = \mathbf{w}^\top \bar{\mathbf{R}}_c \mathbf{w}.$$

Thus, the CSP technique increases the between-class separability through the estimation of the projecting vector \mathbf{w}^* by solving the following problem:

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \frac{\mathbf{w}^\top \bar{\mathbf{R}}_1 \mathbf{w}}{\mathbf{w}^\top \bar{\mathbf{R}}_2 \mathbf{w}}. \quad (6)$$

Problem (6) is a standard eigenvalue problem, and \mathbf{w}^* is equal to the eigenvector that corresponds to the largest eigenvalue of $\bar{\mathbf{R}}_2^{-1} \bar{\mathbf{R}}_1$.

The CSP filter, \mathbf{W} , is a matrix, which is constructed by using $M = 2m$, ($M \leq N$), eigenvectors corresponding to the m largest and m smallest eigenvalues of $\bar{\mathbf{R}}_2^{-1} \bar{\mathbf{R}}_1$. Assuming that the eigenvectors of $\bar{\mathbf{R}}_2^{-1} \bar{\mathbf{R}}_1$ have been sorted in order of decreasing eigenvalues, the CSP filter matrix is defined as

$$\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_m, \mathbf{w}_{N-m+1}, \dots, \mathbf{w}_N]. \quad (7)$$

Having estimated the CSP filter \mathbf{W} , each epoch is spatially filtered by

$$\mathbf{Y}_{c,i} = \mathbf{W} \mathbf{X}_{c,i}, \quad (8)$$

and, finally, it is represented by a feature vector $\mathbf{f}_{c,i}$ of the following form;

$$\mathbf{f}_{c,i} = \log \left[\frac{\text{var}(\mathbf{Y}_{c,i}^1)}{\sum_{j=1}^M \text{var}(\mathbf{Y}_{c,i}^j)} \cdots \frac{\text{var}(\mathbf{Y}_{c,i}^M)}{\sum_{j=1}^M \text{var}(\mathbf{Y}_{c,i}^j)} \right], \quad (9)$$

where $\mathbf{Y}_{c,i}^j$ stands for the j -th row of $\mathbf{Y}_{c,i}$. Typically, features $\mathbf{f}_{c,i}$ are used to train an EEG classifier.

C. The Proposed Unified Neural Network Architecture

The overall architecture of the novel classification scheme, proposed in this work, is presented in Fig. 3. It consists of two key components; i) the CSP encoder layer, and ii) the attention layer that has the form of a fully connected layer. Both of those layers have trainable parameters, which are estimated during the end-to-end training phase of the network. The CSP encoder layer has multiple decoupled CSP modules, i.e., they do not share parameters, that treat different segments of the input EEG epoch sequence (with a certain overlap) as depicted in Fig. 2. This implies that, during the training phase, the CSP encoder layer learns to map the input EEG sequence to multiple feature vectors (i.e., the hidden states of the network). Then, the model automatically weighs the produced hidden states in the attention layer, in order to compute a representative context vector of the EEG patterns. Since the parameters of all layers are being simultaneously estimated during the training phase by minimizing classification error, we argue that the constructed context features can be considered as the most discriminative representation of EEG epochs with respect to the classification task at hand.

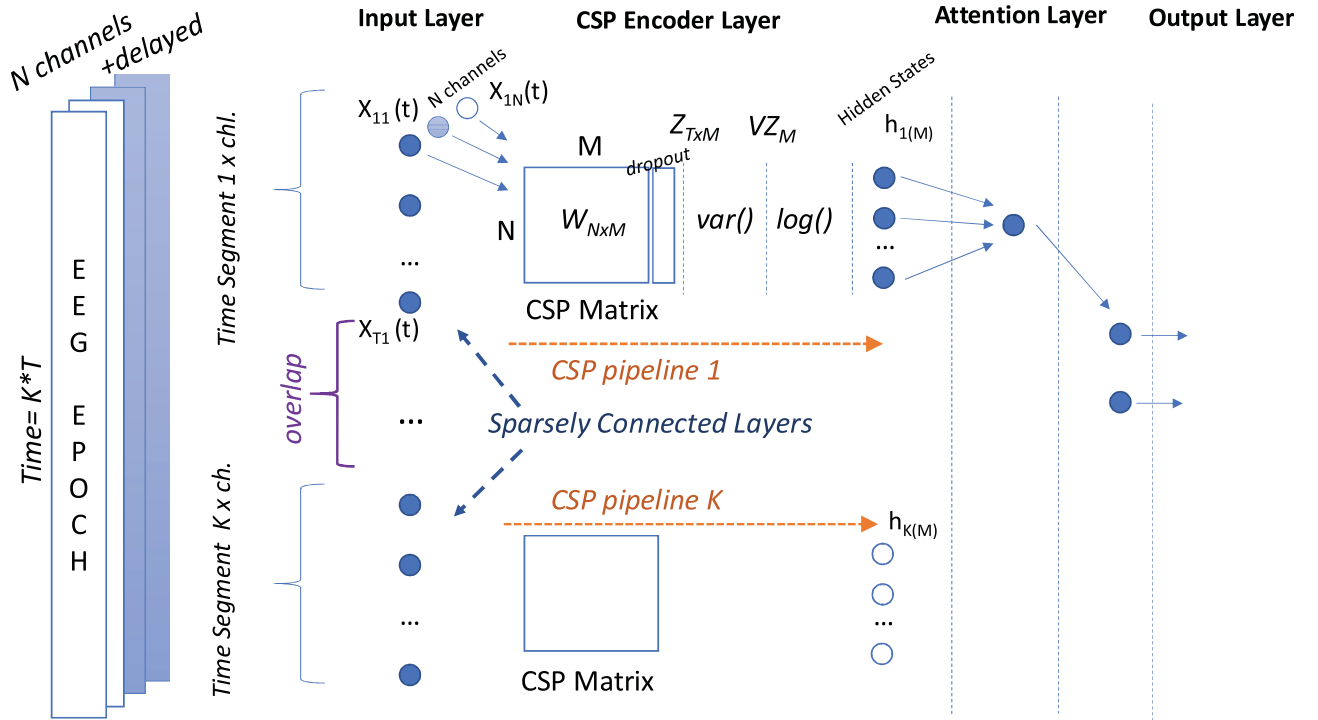


Fig. 3. The overall architecture of the proposed scheme.

The ultimate objective of the proposed learning scheme is to accurately classify EEG epochs. Each EEG epoch can be represented as a matrix, whose each row corresponds to a sequence of measurements that were recorded by a specific electrode (channel) and spans a specific time interval [see relation (3)]. The fact that the statistics of the EEG signals may vary over time, can significantly deteriorate the performance of a conventional classifier. In order to address this problem, we split each EEG epoch in different *segments* along the temporal dimension, constructing a sequence of sequences; each of them is treated by a different CSP module. This segmentation process is similar to applying a sliding kernel filtering approach. If the stride equals the size of the filter, then no overlap between the segments occurs. However, in order to guarantee that any event related potentials are not interrupted across different segments, a hyperparameter is introduced to allow and, at the same time, control the percentage of overlap among subsequent segments.

Moreover, *time delays*, can be introduced in the input EEG sequence in order to embed the idea of spatio-temporal filtering [23] into the proposed model in an indirect way. This can be achieved by concatenating the time delayed EEG epochs along the channels' dimension without affecting the overall architecture of the model.

Having split the EEG epoch into segments, each one of the segments is fed as an input into a CSP pipeline (see Fig. 3). Subsequently, each one of the CSP pipelines output features, as given by (9), which form separate internal hidden states. It should be noted here that the proposed architecture is end-to-end trainable, and, thus each $Y_{c,i}$ [see relation (9)] is being estimated

during the training phase of the network. The output of the i -th CSP pipeline is an M -dimensional representation of the i -th input segment [see relations (7) and (9)]. It should be stressed that the M -dimensional representations of the input segments can be computed independently, something that emphasizes the inherent *parallelism* of the proposed methodology.

Finally, the M -dimensional representations of the segments are fed into the attention layer, which, in its general form as depicted in Fig. 2, consists of multiple instances of a neural network, i.e., it shares parameters among instances. The attention layer, firstly, computes interdependently the parameters of each hidden state, and, then, in a subsequent layer all the hidden states are weighted to form a single context vector. The current implementation of the attention layer is simplified. Instead of using multiple instances of a shared neural network, a fully connected layer is used. This simplified configuration is not affecting the accuracy of the proposed model, as shown in the results in Section IV. The final layer is the output layer, which actually is responsible for the estimation of the decision boundary given a number of available classes and the context vector produced by the attention layer.

To summarize, the proposed novel EEG classification scheme receives as an input a sequence of EEG epochs (or an EEG epoch plus the corresponding delays as different channels) split into segments. Each one of the segments is fed into a separate CSP pipeline, which, in turn, encodes an M -dimensional hidden representation of its input. Then, all hidden representations are fed to a fully connected layer (or multiple fully connected layers with shared weights in its fully blown version), which is

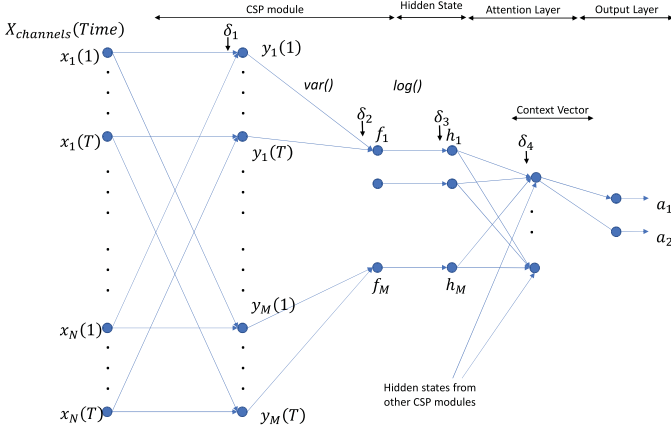


Fig. 4. The architecture of the proposed network.

responsible to weight the hidden states into a context vector. Finally the output layer takes the role of a linear classifier which decides the label of the incoming input sequence. In order to better understand the flow of information through the proposed classification scheme, Fig. 4 presents the architecture of the network when only one segment of the sequence is used as input. A very important feature of our innovative approach is that all CSP encoder pipelines, the attention layer and the output layer, as shown in Fig. 3, are natively embedded into a single and unified neural network architecture, which can be trained end-to-end.

III. TRAINING ALGORITHM

The problem that is being addressed in this work is a classification problem. Therefore, in order to estimate the parameters of the proposed neural network architecture through a training process, the cross entropy loss is minimized with respect to neural network's weights. Firstly the derivative of the error with respect to the network's weights is computed using the backpropagation algorithm, and then, network's weights are updated towards the negative direction of the derivative using a gradient based optimization algorithm.

In the following subsection we introduce the mathematics for the forward pass of information through the network, and then, based on the forward pass, we derive the formulas for computing the derivative of the error with respect the network's weights using the backpropagation algorithm.

A. Forward Pass

The input to the network is an EEG epoch split into segments [see Section II-B]. Each segment is a matrix with $N \times T$ elements, where N is the number of channels and T is the number of samples captured by a specific channel during a predefined time interval. Each one of the segments passes through a CSP pipeline [see Fig. 4], which filters the segment along the spatial dimension (i.e., EEG channels). The filtered segment is a matrix with $M \times T$ elements, where $M < N$. This filtering is realized by multiplying the input with appropriate weights.

Specifically, if we denote as $\mathbf{x}^{(t)} = [x_1(t) \ x_2(t) \ \dots \ x_N(t)]$ the vector that contains the t -th samples of each channel, then the output of CSP pipelines with respect to $\mathbf{x}^{(t)}$ is given as

$$\mathbf{y}^{(t)} = \langle \mathbf{W}_{CSP}, \mathbf{x}^{(t)} \rangle, \quad (10)$$

where $\mathbf{W}_{CSP} \in \mathbb{R}^{M \times N}$ are the weights of CSP pipelines that treat the EEG signals captured at time t , while

$$\mathbf{y}^{(t)} = [y_1(t), y_2(t), \dots, y_M(t)] \in \mathbb{R}^M. \quad (11)$$

Aggregating together $\mathbf{y}^{(t)}$ for $t = 1, 2, \dots, T$ the matrix

$$\mathbf{Y} = [\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(T)}] \in \mathbb{R}^{M \times T} \quad (12)$$

can be formed. Let us denote as

$$\mathbf{y}_m := [y_m(1), y_m(2), \dots, y_m(T)] \quad (13)$$

the m -th row of matrix \mathbf{Y} . Then, following the original CSP algorithm, the neural network computes the variance of each row \mathbf{y}_m to form the following vector:

$$\mathbf{f} = [f_1, f_2, \dots, f_M], \quad (14)$$

where $f_m = \text{var}(\mathbf{y}_m)$ is the variance of the m -th row of matrix \mathbf{Y} . At the next step, and according to relation (9), the network produced the CSP-like feature vector

$$\mathbf{h} = \log(\mathbf{f}). \quad (15)$$

In relation (15) the $\log(\cdot)$ operation is performed element wise. After the computation of the CSP-like feature vector \mathbf{h} , the information is propagated through the layers of the network towards the output layer, in the same way as when fully connected feed forward neural networks are used.

B. Backward Pass

As mentioned before, the parameters of the proposed neural network (i.e., network's weights) are estimated by minimizing the cross entropy loss function with respect to network's weights. For doing this, the partial derivative of the loss function with respect to network's weights is computed using the backpropagation algorithm. It should be stressed that via the backpropagation algorithm the partial derivatives of all network's weights, both in the CSP pipelines and the fully connected layers, are computed.

As the spatial filtering is occurred in the spatio-temporal domain within a complete EEG epoch, the parameters should be updated after a complete EEG epoch is fed to the network, when stochastic gradient descent is utilized, or a batch of an EEG epochs, when batch gradient decent is used. In other words for our specific EEG classification problem, each training example (i.e., EEG epoch) consists of $T(\text{samples}) \times N(\text{channels})$ and we need at least one such example in order to perform the back propagation task.

The error function is calculated according to the network output and the class label(s) of the EEG epochs presented in the network for the current example (batch of examples). All the weights are initialized randomly before the training process.

The partial derivatives of the loss function with respect to the weights of the fully connected layers of the proposed neural

network can be computed via a typical application of the back-propagation algorithm. Therefore, what needs to be described is the computation of the partial derivatives of the loss function with respect to the weights of the CSP pipeline of the proposed architecture.

Let us denote as δ_4 the error with respect to the loss function and the weights of first fully connected layer of the proposed architecture [see Fig. 4], and as δ_3 and δ_2 the error with respect to the loss function, and the $\log(\cdot)$ and $\text{var}(\cdot)$ operations, respectively, of the CSP pipeline of the network. Finally, let us denote as $\delta_{1,t}$ the error with respect to the loss function and the weights of the input $\mathbf{x}^{(t)}$ [see relations (10) and (11)]. Then,

$$\delta_3 = \langle \mathbf{W}, \delta_4 \rangle \in \mathbb{R}^{M \times 1}, \quad (16)$$

where \mathbf{W} stand for the weights that connect the output of the CSP pipeline to the first fully connected layer. Regarding δ_2 , and $\delta_{1,t}$, we have that

$$\delta_2 = \delta_3 \odot \mathbf{f}^{-1} \in \mathbb{R}^{M \times 1} \quad (17)$$

and

$$\delta_{1,m} = [\mathbf{I}^T \delta_{2,m}] \odot \left(\frac{2}{T|\mathbf{W}_m|} \mathbf{y}_m \right) \in \mathbb{R}^{T \times 1}. \quad (18)$$

In relation (18) \mathbf{I} is a vector with T ones, $\delta_{2,m}$ is the m -th element of δ_2 , and \mathbf{W}_m are the weights that transform the input into \mathbf{y}_m . The operator “ \odot ” represents the element wise multiplication. Finally,

$$\delta_1 = [\delta_{1,1}, \delta_{1,2}, \dots, \delta_{1,M}] \in \mathbb{R}^{T \times M}. \quad (19)$$

Having estimated the quantity δ_1 the partial derivative of the loss function with respect to the \mathbf{W}_{CSP} parameters is given by

$$\frac{\partial C}{\partial \mathbf{W}_{CSP}} = \langle \delta_1, \mathbf{X} \rangle, \quad (20)$$

where C is the loss function and $\mathbf{X} = [\mathbf{x}^{(1)} \ \mathbf{x}^{(2)} \ \dots \ \mathbf{x}^{(T)}]^\top$.

After the computation of loss function partial derivatives with respect to the network weights, a gradient based optimization algorithm can be used for minimizing the loss by updating the network’s weights towards the negative direction of the derivatives.

IV. EVALUATION RESULTS

In this section we investigate the performance of the proposed methodology in terms of EEG signal classification accuracy. We also present a prototype hardware implementation of our proposed scheme and evaluate its efficiency in terms of processing speed and energy consumption.

A. Dataset Description and Experimental Setting

For the evaluation of the proposed scheme, we use the public available and widely used Cichockis¹ benchmark. This dataset contains several EEG data sets that were recorded from healthy subjects. The cue-based BCI paradigm consisted of trials in two different setups of motor imagery tasks. The first set (LH/RH)

TABLE I
DESCRIPTION OF THE CHICHOCKI’S DATASET

Session ID	Subject	Class	Channels	Duration	Trials
A6chan2LRs1	A	LH/RH	6	3 sec	130
A6chan2LRs2					134
B6chan2LR	B	LH/RH	6	4 sec	162
C6chan2LRs1	C	LH/RH	6		3 sec
C6chan2LRs2					158
C6chan2LRs3					48
C6chan2LRs4					120
C6chan2LRs5					90
D5chan2LR	D	LH/RH	5	4 sec	80
E5chan2LR	E	LH/RH	5	4 sec	48
F6chan2LR	F	LH/RH	6	4 sec	80
G6chan2LR	G	LH/RH	6	4 sec	120
H6chan2LR	H	LH/RH	6	3 sec	150
A6chan2LF	A	LH/RF	6	3 sec	150
C6chan2LFs1	C	LH/F	6	3 sec	330
C6chan2LFs2					180
C6chan2LFs3					102

considers the imagination of movement of the left hand (LH) or right hand (RH). The second set (LH/F) the imagination of left hand (LH) or both feet (F). Apparently both tasks with regard to machine learning, fall in the same category of binary classification. Table I provides the detail information of data sets which are used in our experiments. It should be mentioned that the sampling rate for all data sets is 256 Hz.

The aforementioned data sets are used for evaluating the classification accuracy of the proposed scheme both when the training and the testing data are coming from the same recording session (single-session evaluation), and when the training and the testing data are coming from multiple sessions, recorded at different times (between-sessions evaluation). We focus though in the multiple sessions scenario as the proposed architecture is the first introduced which tries to handle the time variability (non-stationary process) of the EEG signal between different sessions.

All experiments are conducted following a 5-fold cross-validation approach, and the classification accuracy of our methodology is compared against a logistic regression classifier that is fed with EEG features produced via the CSP algorithm (LRCSP). The LRCSP is the most commonly used approach for analysing and classifying EEG signals, and, in this work, it serves as a baseline benchmark. Despite the fact that our methodology permits the design of deep networks, we decide to use a network without any fully connected hidden layers. Instead, we design and utilize a network with one hidden layer, that corresponding to the CSP pipelines. This way, our network produces linear decision boundaries in the space of the CSP-like features produced by the CSP pipelines, and, thus, we can conduct a fair comparison against LRCSP, which is also linear in the space of CSP features. Moreover, for all the experiments we used the same hyperparameters for the learning algorithms. Specifically, we use Stochastic Gradient Descent optimizer for learning the network weights. We set the initial learning rate equal to 0.15 and utilize a learning rate drop strategy by reducing the learning rate 20% every 100 training epochs, while the training process does not employ any stopping criteria; instead, it terminates after 1000 training epochs. We use 5% dropout in the input layer

¹<http://www.bsp.brain.riken.jp/~qibin/homepage/Datasets.html>

(i.e., the CSP layer), while also implemented a set of 5 different time delays (i.e., delays = 2, 4, 8, 12, 16) in the input channels that simulates the method of [23]. Time delays are concatenated in the channel dimension which gives as a total number of 36 channels (5 delays for each one of the 6 channels plus original 6 channels). Finally, the overlap percentage of EEG segments is set to 50%. The values for the aforementioned parameters were selected via the employment of cross-validation combined with a grid-search approach.

Before proceeding to the experimental validation of the proposed model, we have to give some information regarding the number of its trainable parameters. The number of parameters of the proposed network is equal to

$$K \cdot M \cdot N + K \cdot M \cdot H_1 + \sum_{i=1}^{L-1} (H_i + 1) \cdot H_{i+1} + H_L + 1 \quad (21)$$

where K the number of EEG segments, M the number of employed CSP pipelines (filters), and N the number of EEG channels (plus the delayed channels). Variable H_i stands for the number of neurons in the i -th fully connected hidden layer. As mentioned before, in our experiments we used a network with one hidden layer, that of the CSP pipelines, although our methodology is general enough to support the design of deeper architectures. Without any fully connected hidden layers, and by setting the number of overlapping segments (parameter K) equal to 2 and the number of CSP filters (parameter M) equal to 4, the number of parameters of our network is 297. The number of trainable parameters is relatively small due to the sparse connectivity property of the CSP pipelines. Since, the number of parameters is comparable to the number of available samples for training (see Table I), our network is able to avoid overfitting. In addition, the very strong inductive bias introduced via the CSP pipeline, which applies a very specific transformation on the input, shields further our network against overfitting. In general, the introduction of bias in a learning model reduces model's variance making it less prone to overfitting, see [29] Ch.5 and Ch.13.2.

B. Single-Session Evaluation

In this subsection, we evaluate the performance of the proposed scheme in the case where the training and the testing sets are coming from the same session. Our proposed network is parameterized by K , that is the number of overlapping segments, and M , that is the number of CSP filters. Therefore, we conduct two different sets of experiments in order to evaluate its performance with respect to these two parameters.

In the first set of experiments, we keep parameter K fixed ($K = 2$) and evaluate the performance of the proposed approach using different number of CSP filters (M parameter). This way we can investigate the effect of parameter M on the performance of the proposed methodology, and also to compare the classification accuracy of the proposed method against LRCSP. As mentioned before, for all the experiments we followed a 5-fold cross validation setting. At this point it has to be mentioned, that for each one of the experiments we utilize the same number of filters both for our approach and for LRCSP. The results of this

evaluation are presented in Table II. For each one of the models, the average classification accuracy and the standard deviation across the 5 folds are reported. As it can be seen, our proposed methodology consistently outperforms the LRCSP methodology when more than 1 filters are used, despite the fact that both methodologies utilize the same number of filters. The only exception is the fifth session i.e., 5Dchan2LR. For this session more than 2 filters are required in order for our approach to outperform LRCSP. In the case of $M = 1$ the number of trainable parameters employed by our approach is very small. Due to this fact our method seems to not be able to fit the data, and, thus, the LRCSP method performs better.

Besides the average classification accuracy, Table II also presents the standard deviation of the models. The standard deviation can provide an indication of the statistical significance of the results. In order, however, to be more rigorous, we also conducted t-tests to compare the performance of the models in Table II. Via the t-tests we check whether or not the null hypothesis that the two approaches, our method and the LRCSP, perform the same is valid. For the experiments in Table II we used four different sets of parameters for each approach, which implies that in total there are four different models of our proposed network and four different LRCSP models. For each set of parameters we conducted a t-test to check whether the corresponding models perform the same over all subjects. The p-values for the null hypothesis to hold for the models parameterized by one, two, four and six filters are 0.89, 0.05, 0.01, and 0.02 respectively. Based on these t-tests, we can safely reject the null hypothesis for the three out of the four cases, and conclude that the performance of our approach is statistically superior than the performance of the LRCSP model.

In the second sets of experiments, we keep fixed the parameter M ($M = 2$), and evaluate the performance of our approach using different number of overlapping time segments, that is parameter K . In these experiments, we do not perform any comparison against LRCSP, since this method does not employ multiple overlapping time segments, and, thus, its performance is not affected by this parameter. Again, we follow a 5-fold cross-validation approach, in order to evaluate the classification accuracy of our methodology. The results of these experiments are summarized in Table III. For $K = 2$ and $K = 3$ our method achieves the highest classification accuracy for all subjects. For values of K larger than 2 the performance slightly decreases. This implies that two or three overlapping time windows are adequate enough for capturing the non-stationarities of the EEG signals.

C. Between-Sessions Evaluation

In this set of experiments we evaluate the classification accuracy of the models when the training and the testing data are coming from different sessions. In other words, we use a number of sessions to train the network and a single session to test its accuracy. We repeat the same procedure after changing the training and test sessions like in a k-Fold cross validation scenario. The average classification accuracy is computed after all the possible iterations (e.g., for 3 sessions we have 3 possible

TABLE II
AVERAGE CLASSIFICATION ACCURACY AND STANDARD DEVIATION OF THE PROPOSED MODEL AND LRCSP WHEN DIFFERENT NUMBER OF CSP FILTERS ARE USED

Session	M=1		M=2		M=4		M=6	
	Our	LRCSP	Our	LRCSP	Our	LRCSP	Our	LRCSP
A6chan2LRs1	0.762 ± .09	0.823 ± .09	0.900 ± .02	0.838 ± .06	0.892 ± .02	0.831 ± .06	0.892 ± .04	0.838 ± .06
A6chan2LRs2	0.758 ± .05	0.741 ± .10	0.814 ± .06	0.741 ± .10	0.814 ± .03	0.792 ± .07	0.844 ± .07	0.821 ± .06
B6chan2LR	0.834 ± .05	0.851 ± .07	0.914 ± .02	0.871 ± .03	0.913 ± .02	0.860 ± .05	0.870 ± .07	0.858 ± .05
D5chan2LR	0.488 ± .09	0.475 ± .08	0.588 ± .03	0.613 ± .03	0.650 ± .09	0.625 ± .03	0.663 ± .08	0.625 ± .03
E5chan2LR	0.935 ± .05	0.675 ± .14	0.930 ± .10	0.795 ± .12	0.935 ± .06	0.795 ± .12	0.935 ± .06	0.795 ± .12
F6chan2LR	0.588 ± .05	0.650 ± .13	0.625 ± .07	0.613 ± .11	0.663 ± .03	0.625 ± .09	0.663 ± .03	0.625 ± .09
G6chan2LR	0.692 ± .08	0.833 ± .09	0.808 ± .07	0.792 ± .08	0.842 ± .08	0.792 ± .08	0.817 ± .07	0.792 ± .08
H6chan2LR	0.527 ± .10	0.580 ± .01	0.720 ± .06	0.613 ± .03	0.773 ± .11	0.660 ± .08	0.753 ± .05	0.653 ± .08

TABLE III
AVERAGE CLASSIFICATION ACCURACY AND STANDARD DEVIATION OF THE PROPOSED MODEL WHEN DIFFERENT NUMBER OF OVERLAPPING WINDOWS ARE USED

Session	K=2	K=3	K=4	K=5
A6chan2LF	0.88 ± .06	0.87 ± .06	0.847 ± .08	0.84 ± .10
A6chan2LRs1	0.90 ± .02	0.86 ± .02	0.79 ± .08	0.79 ± .08
A6chan2LRs2	0.81 ± .06	0.80 ± .04	0.78 ± .03	0.76 ± .02
B6chan2LR	0.91 ± .02	0.88 ± .06	0.87 ± .03	0.87 ± .05
D5chan2LR	0.59 ± .03	0.66 ± .10	0.61 ± .09	0.61 ± .09
E5chan2LR	0.94 ± .06	0.93 ± .10	0.93 ± .10	0.93 ± .10
F6chan2LR	0.63 ± .07	0.63 ± .09	0.61 ± .09	0.61 ± .11
G6chan2LR	0.81 ± .07	0.81 ± .10	0.78 ± .09	0.75 ± .09
H6chan2LR	0.72 ± .06	0.73 ± .06	0.70 ± .09	0.70 ± .12

iterations). In Chichocki's dataset, data for between-session performance evaluation are available only for one subject, that is Subject C, which is conducting two different tasks.

In order to provide a thorough investigation regarding the between-session performance of the models we conducted a set of experiments keeping the parameter K fixed ($K = 2$) and evaluating the performance of the two approaches, our approach and LRCSP, using different number of filters (M parameter). This way, we can, firstly, investigate the effect of M parameter on the models' between-session performance, and, secondly, build different models for conducting statistical tests. For each of the tasks the performance results are presented in Table IV. These results suggest that our approach irrespective of the value of parameter M outperforms the LRCSP model. In order to evaluate the significance of these results we conducted t-tests for accepting or rejecting the null hypothesis that the two models perform the same. For the first task, that is Cchan2LF, the p-values for accepting the null hypothesis are 0.001, 0.008, 0.05, and 0.05 for $M = 1$, $M = 2$, $M = 4$, and $M = 6$ respectively. Therefore, we can safely reject the null hypothesis and conclude that our approach is superior that LRCSP. Regarding the second task, that is Cchan2LR, the corresponding p-values are 0.66, 0.45, 0.19, and 0.24. In this case we cannot safely reject the null hypothesis, despite the fact that our models seems to significantly outperform the LRCSP approach.

Finally, the Epoch plot diagrams are depicted in Fig. 5. In these diagrams the time variability of the signal in the case of between-session evaluation is presented. In order to visually depict the time variability phenomenon all epochs were stacked and reordered using a 1D spectral embedding as described in [30]. The power of the signal in all channels show a similar behavior

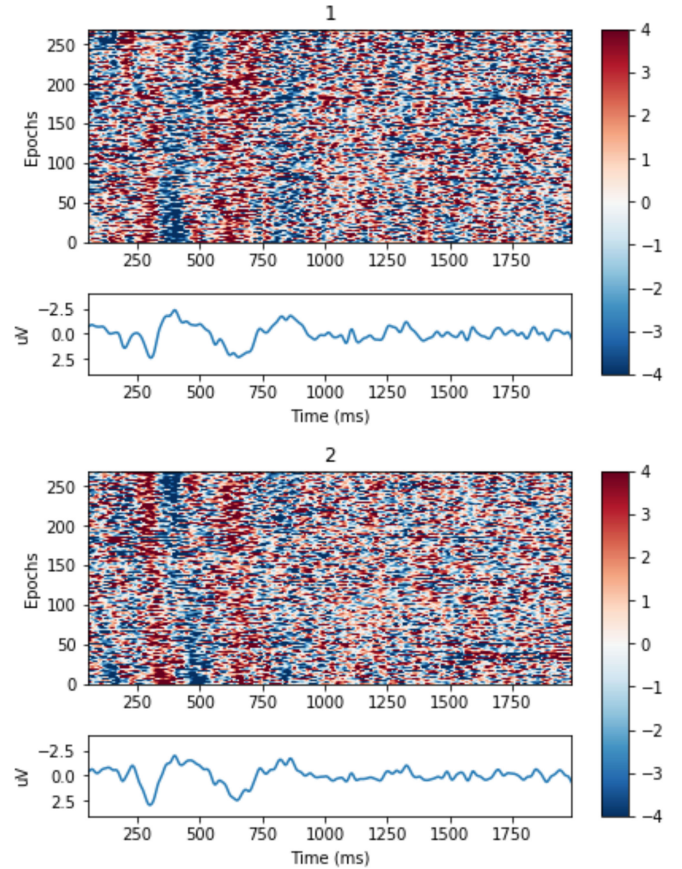


Fig. 5. Epoch plot diagrams for ch1, ch2. Experiment: Subject 3 6chan2LR for 4 sessions.

where the event related potentials are displaced in the duration of the Epochs across sessions. Our scheme though seems to handle it equally well as in the within-session (cross-validated) evaluation where the time variability of the signal is much smaller.

Before we present the prototype hardware implementation of our proposed scheme, we have to state that our approach is a strict generalization of LRCSP. In the case where a single time window is used ($K = 1$), setting the same value for parameter M , i.e., the number of CSP filters, will made the two approaches equivalent, and thus their performances is expected to be the same. Using, however, more than one overlapping time windows, which is a unique feature of our methodology, the two approaches cease to be equivalent, with our approach to achieve higher classification

TABLE IV
AVERAGE CLASSIFICATION ACCURACY AND STANDARD DEVIATION OF THE PROPOSED MODEL ACROSS DIFFERENT SESSIONS WHEN DIFFERENT NUMBER OF CSP FILTERS ARE USED

Cchan2LF	M=1		M=2		M=4		M=6	
	Our	LRCSP	Our	LRCSP	Our	LRCSP	Our	LRCSP
train/test session								
s1+s3/s2	0.944	0.816	0.961	0.844	0.944	0.850	0.938	0.850
s2+s3/s1	0.948	0.849	0.942	0.878	0.954	0.903	0.957	0.903
s1+s2/s3	0.951	0.794	0.951	0.803	0.931	0.774	0.941	0.774
Average	0.948	0.820	0.951	0.842	0.943	0.842	0.945	0.842
Std	0.003	0.027	0.009	0.037	0.011	0.064	0.010	0.064
Cchan2LR	M=1		M=2		M=4		M=6	
	Our	LRCSP	Our	LRCSP	Our	LRCSP	Our	LRCSP
train/test session								
s1+s2+s4/s5	0.729	0.747	0.770	0.682	0.655	0.544	0.688	0.544
s1+s2+s5/s4	0.753	0.772	0.633	0.620	0.816	0.775	0.825	0.808
s1+s4+s5/s2	0.650	0.758	0.833	0.800	0.727	0.614	0.677	0.613
s2+s4+s5/s1	0.777	0.510	0.633	0.522	0.817	0.711	0.858	0.705
Average	0.727	0.696	0.717	0.656	0.754	0.661	0.762	0.668
Std	0.055	0.125	0.100	0.116	0.078	0.102	0.092	0.114

results (see Tables II, III, and IV). This superiority, in terms of classification accuracy, lies mainly in the fact that we split the time sequence in overlapping time-segments that are followed by separate CSP modules in order to address the non-stationarity characteristics of the EEG signal.

This superiority, however, in terms of classification results, comes at a cost in the time required to train our model. Specifically, LRSCP requires about 1.5 seconds for training using data for one subject, while approach requires about 23 seconds.² This fact encouraged us to further propose a prototype hardware implementation in order to efficiently speedup computation and make our approach applicable in real application scenarios where continuous model adaptation is needed. Regarding predict times, both methods are able to provide predictions in less than 0.2 seconds. The LRCSP method provides predictions, firstly, by the application of a linear transformation on the data, and, secondly, by the evaluation of the logistic regression function estimated during training. Our approach provides predictions via a single evaluation of a nonlinear function modeled by the proposed network. The prediction times, which are crucial for the applicability of a BCI system, are very small for both methods making them suitable for real-time applications.

D. Hardware Implementation

In this section, a prototype implementation of our novel scheme in custom hardware is demonstrated. Our hardware module can be used as an accelerator hosted in a SoC (System on Chip) of an embedded device such as a processing-capable EEG headset or any other smart device connected to an EEG headset like a mobile phone. For the latter, hardware accelerators are getting very popular nowadays among the SoCs utilized in all the modern phones as they offer high processing power at a very low energy consumption. As the EEG processing, in the form, for example, of a BCI, should be performed continuously in an on-line fashion the corresponding energy consumption is very important.

²these times were obtained on a specific laptop with Intel i7 CPU, and may differ on other machines.

TABLE V
TYPICAL HARDWARE COST ON XILINX ARTIX7 (XQ7A200TRB484-1I) - KINTEX035 (XCKU035-FBVA676-1-C) & KINTEX115 (XQKU115-VLRF1924-2-1) DEVICES (PIPELINE AND UNROLL)

Logic Utilization	Artix7	Kintex35	Kintex115
Number of Flip Flops	39% (105536)	45% (185175)	47% (624077)
Number of Slice LUTs	70% (95319)	78% (158769)	69% (463706)
Number of DSP48E	87% (647)	62% (1067)	65% (3612)
Number of Block RAM 18K	40% (296)	50% (542)	26% (1150)

The prototype implementation of our scheme takes place in a number of low cost as well as high-end FPGA³ devices, so as to demonstrate its efficiency when implemented in hardware. The three devices that are selected are the low-end Xilinx Artix7, the medium-end Xilinx Kintex7-035, and the high-end Xilinx Kintex7-115 UltraScale FPGA devices. Table V presents the hardware cost for the Artix7 and Kintex 35 devices. Due to the inherent parallelism and modularity of the proposed novel scheme, the more hardware resources are being available, the more resources covered by our implementation using more than 60% of DSP in all devices.

E. Performance and Optimization

In this section the performance of the proposed hardware prototype implementations in terms of processing speed per batch (μs) and consumed energy per batch (*mjoule*) is presented. In order to demonstrate the efficiency of our hardware implementation the performance and energy consumption of our prototypes are compared to those triggered by an Intel i7 7700HQ energy efficient mobile CPU running at 3.80 GHz. The code running in the CPU is a single threaded fully optimized (-O3) C++ code, where the hardware id produced by using a C-based description and the Vivado HLS⁴ FPGA design tool, with the optimization pragmas UNROLL and PIPELINE. The hotspot of

³<https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>

⁴<https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html>

TABLE VI
PERFORMANCE AND POWER CONSUMPTION

Batch size:5	Artix7	Kintex35 UltraScale	Kintex115 UltraScale	Intel i7-7700HQ
Training (us)	142.6	83.8	31.95	242.1
Training-energy (mjoules/batch)	0.37	0.28	0.26	4.35
Predict (us)	102.2	61.8	13.34	212.3
Predict-energy (mjoules/batch)	0.27	0.20	0.11	3.82

TABLE VII
OPTIMIZATION IN THE CSP BOTTLENECK OF THE SCHEME

Optimization	No optimization	Unroll	Unroll & Pipeline
CSP Loop latency (us)	843.5	290.8 (2.9x)	19.4 (43.47x)
Total latency Kintex035 (training-us)	2885.2	471.4 (6.12x)	83.8 (34.42x)
Total latency Kintex035 (predict-us)	1742.7	333.2 (5.23x)	61.8 (28.2x)

the algorithm is the dot products in forward and backward propagation and more specifically in the CSP matrix computation. This is because after the CSP dot products the time dimension of the signal is absorbed and consequently the dimensionality of the subsequent layers is drastically reduced. Table VI clearly demonstrates that our hardware prototype (Kintex115) is 7.5x (training) & 16x (predict) faster when compared to the Intel CPU in terms of execution performance. In terms of execution time per watt, measured in *mjoules/batch*, Kintex 115 is more than an order of magnitude more efficient than the Intel CPU.

Table VII demonstrates the effect of the optimizations pragmas in the bottleneck of the algorithm in the overall execution performance. Our hardware implementations consume up to an order of magnitude less energy when compared with the low-power Intel CPU. It should also be stressed that both performance and power consumption numbers are coming from an FPGA prototype implementation; if our system is implemented as a hardware accelerator within a state-of-the-art SoC both the performance and the energy numbers are expected to be improved significantly.

V. CONCLUSIONS

This paper proposes a novel neural network architecture for EEG processing which inherently adopts the well-known method of spatial filtering and extends it by using the notion of attention in deep neural networks. Although, in this work we demonstrate the performance of our approach using a neural network without any fully connected hidden layers, our methodology is a general one that permits the design of deeper architectures. The proposed scheme efficiently handles the inherent time variability of the EEG signals due to its unique architecture that utilizes CSP pipelines across the time domain coupled together with an attention mechanism that automatically weights the hidden states that CSP pipelines are producing. Finally, a very efficient prototype implementation of our scheme in reconfigurable hardware is presented, in order to demonstrate that it

is suitable for ultra low power custom accelerators that can perform seamless EEG processing in portable headsets under very low energy budget.

REFERENCES

- [1] J. N. Mak and J. R. Wolpaw, "Clinical applications of brain-computer interfaces: Current state and future prospects," *IEEE Rev. Biomed. Eng.*, vol. 2, pp. 187–199, 2009. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2862632/>
- [2] S. Muthong, P. Vateekul, and M. Sriyudthasak, "Stacked probabilistic regularized LDA on partitioning non-stationary EEG data for left/right hand imagery classification," in *Proc. IEEE EMBS Conf. Biomed. Eng. Sci.*, 2016, pp. 301–306.
- [3] S. R. Liyanage, C. Guan, H. Zhang, K. K. Ang, J. Xu, and T. H. Lee, "Dynamically weighted ensemble classification for non-stationary EEG processing," *J. Neural Eng.*, vol. 10, no. 3, 2013, Art. no. 036007.
- [4] X. Yu, P. Chum, and K.-B. Sim, "Analysis the effect of PCA for feature reduction in non-stationary EEG based motor imagery of BCI system," *Optik-Int. J. Light Electron Opt.*, vol. 125, no. 3, pp. 1498–1502, 2014.
- [5] S. D. Muthukumaraswamy and K. D. Singh, "Visual gamma oscillations: The effects of stimulus type, visual field coverage and stimulus motion on MEG and EEG recordings," *Neuroimage*, vol. 69, pp. 223–230, 2013.
- [6] A. R. Marathe, A. J. Ries, and K. McDowell, "Sliding HDCA: Single-Trial EEG classification to overcome and quantify temporal variability," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 2, pp. 201–211, Mar. 2014.
- [7] N. Robinson, A. P. Vinod, K. K. Ang, K. P. Tee, and C. T. Guan, "EEG-based classification of fast and slow hand movements using wavelet-CSP algorithm," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2123–2132, Aug. 2013.
- [8] B. Rivet, A. Souilmiac, V. Attina, and G. Gibert, "xDawn algorithm to enhance evoked potentials: Application to brain-computer interface," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 8, pp. 2035–2043, Aug. 2009.
- [9] J. Lin, S. Liu, G. Huang, Z. Zhang, and K. Huang, "The recognition of driving action based on EEG signals using wavelet-CSP algorithm," in *Proc. IEEE 23rd Int. Conf. Digit. Signal Process.*, 2018, pp. 1–5.
- [10] B. Lin, W. Zhang, and X. Zhang, "Feature extraction of motion imagination EEG based on S transform and CSP," in *Proc. 37th Chin. Control Conf.*, 2018, pp. 4426–4429.
- [11] K. Makantasis, A. Doulamis, N. Doulamis, and A. Voulodimos, "Common Mode Patterns for Supervised Tensor Subspace Learning," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom 2019, pp. 2927–2931. doi: 10.1109/ICASSP.2019.8682616.
- [12] R. Zhang, "A new motor imagery EEG classification method FB-TRCSP+RF based on CSP and random forest," *IEEE Access*, vol. 6, pp. 44 944–44 950, 2018.
- [13] Z. J. Koles, "The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG," *Electroencephalography Clin. Neurophysiology*, vol. 79, no. 6, pp. 440–447, Dec. 1991.
- [14] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 441–446, Dec. 2000.
- [15] K. Fukunaga and W. L. Koontz, "Application of the Karhunen-Loeve expansion to feature selection and ordering," *IEEE Trans. Comput.*, vol. 10, no. 4, pp. 311–318, Apr. 1970.
- [16] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller, "Increase information transfer rates in BCI by CSP extension to multi-class," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 733–740.
- [17] T. Yan, T. Jingtian, and G. Andong, "Multi-class EEG classification for brain computer interface based on CSP," in *Proc. Int. Conf. Biomed. Eng. Inform.*, 2008, vol. 2, pp. 469–472.
- [18] H. Lu, H.-L. Eng, C. Guan, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularized common spatial pattern with aggregation for eeg classification in small-sample setting," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 12, pp. 2936–2946, Dec. 2010.
- [19] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 2, pp. 355–362, Feb. 2011.

[20] Y. Zhang, Q. Zhao, G. Zhou, X. Wang, and A. Cichocki, "Regularized CSP with fisher's criterion to improve classification of single-trial ERPS for BCI," in *Proc. 9th Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2012, pp. 891–895.

[21] P. von Bnau, F. C. Meinecke, S. Scholler, and K. R. Müller, "Finding stationary brain sources in EEG data," in *Proc. Annu. Int. Conf. IEEE Eng. Med.*, Aug. 2010, pp. 2810–2813.

[22] W. Samek, C. Vidaurre, K.-R. Müller, and M. Kawanabe, "Stationary common spatial patterns for brain-computer interfacing," *J. Neural Eng.*, vol. 9, no. 2, 2012, Art. no. 026013.

[23] S. Lemm, B. Blankertz, G. Curio, and K. R. Müller, "Spatio-spectral filters for improving the classification of single trial EEG," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 9, pp. 1541–1548, Sep. 2005.

[24] Z. J. Koles, M. S. Lazar, and S. Z. Zhou, "Spatial patterns underlying population differences in the background EEG," *Brain Topography*, vol. 2, no. 4, pp. 275–284, Jun. 1990. [Online]. Available: <https://doi.org/10.1007/BF01129656>

[25] A. Yuksel and T. Olmez, "A neural network-based optimal spatial filter design method for motor imagery classification," *PLoS One*, vol. 10, no. 5, May 2015, Art. no. e0125039.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv: 1409.0473 2014.

[27] B. Reuderink and M. Poel, "Robustness of the common spatial patterns algorithm in the BCI-pipeline," Univ. Twente, Tech. Rep. DTR08-9/TR-CTIT-08-52, 2008.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[29] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[30] A. Gramfort, R. Keriven, and M. Clerc, "Graph-based variability estimation in single-trial event-related neural responses," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 5, pp. 1051–1061, May 2010.



Antonis Nikitakis received the Engineering Diploma in electrical and computing engineering from the Democritus University of Thrace, Komotini, Greece, with specialization in hardware computer and cryptography, the master's degree in electronic and computer engineering from the Technical University of Crete, Chania, Greece, with specialization in computer architecture and hardware design, the Ph.D. degree in electronic and computer engineering from Technical University of Crete with area of specialization in SoC design in computer vision applications,

and the bachelor's degree in psychology from the Psychology Department of the University of Crete. He is currently an Engineer and a Psychologist. A part of his research which presented in his Thesis titled: High Performance Low Power Embedded Vision Systems, rewarded in ESTIMedia 2012 with the Best paper award. He has years of experience as a Hardware Engineer working in the research and the industry. He has also collaborated with plenty of companies and university to carry out European Programs. Moreover, he has collaborated with experimental psychology labs and he combines his psychology knowledge with his expertise in computer vision and machine learning.



Konstantinos Makantasis received the Computer Engineering Diploma from the Technical University of Crete (TUC), Crete, Greece, and the master's degree from the Department of Production Engineering and Management, TUC, and the Ph.D. degree from the same school working on detection and semantic analysis of objects and events through visual cues. His diploma thesis entitled "Human face detection and tracking using AIBO robots," while his master thesis entitled "Persons' fall detection through visual cues." He is mostly involved and interested in computer vision, both for visual spectrum (RGB) and hyperspectral data, and in machine learning/pattern recognition and probabilistic programming. He has more than 20 publications in international journals and conferences on computer vision, signal and image processing, and machine learning. He has been involved for more than seven years as a Researcher in numerous European and national competing research programs (Interreg, FP7, Marie Curie actions) towards the design, development and validation of state-of-the-art methodologies, and cutting-edge technologies in data analytics and computer vision.



Nikolaos Tampouratzis received the B.Sc. degree from the University of Crete, Crete, Greece, and the M.Sc. and Ph.D. degrees from the Technical University of Crete, Chania, Greece, with specialization in computer architecture and hardware design. He is currently a Researcher with Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece, working on simulation tools for computing systems. He has joined Telecommunication Systems Institute, Technical University of Crete since October 2012 as a Research Associate, providing research and development services to several EU-funded research projects. Apart from AUTH, he currently also provides research and development services to Synelxis Solutions Ltd., as part of the EU-funded EUROEXA research project.



Ioannis Papaefstathiou received the Ph.D. degree in computer science from the University of Cambridge, Cambridge, U.K. He is currently a Manager of the HPC and Embedded Systems Group of Synelxis Solutions S.A. and an Associate Professor with the Electrical and Computer Engineering School, Aristotle University of Thessaloniki, Thessaloniki, Greece. His current research interests focus on architectures and efficient implementations of HPC, CPS, and reconfigurable systems.