Annals of Operations Research
https://doi.org/10.1007/s10479-020-03921-0

ORIGINAL RESEARCH

Improving P300 Speller performance by means of optimization and machine learning

Luigi Bianchi[1] · **Chiara Liti**[1] · **Giampaolo Liuzzi**[2] · **Veronica Piccialli**[1] · **Cecilia Salvatore**[1]

Accepted: 24 December 2020

Abstract
Brain-Computer Interfaces (BCIs) are systems allowing people to interact with the environment bypassing the natural neuromuscular and hormonal outputs of the peripheral nervous system (PNS). These interfaces record a user's brain activity and translate it into control commands for external devices, thus providing the PNS with additional artificial outputs. In this framework, the BCIs based on the P300 Event-Related Potentials (ERP), which represent the electrical responses recorded from the brain after specific events or stimuli, have proven to be particularly successful and robust. The presence or the absence of a P300 evoked potential within the EEG features is determined through a classification algorithm. Linear classifiers such as stepwise linear discriminant analysis and support vector machine (SVM) are the most used discriminant algorithms for ERPs' classification. Due to the low signal-to-noise ratio of the EEG signals, multiple stimulation sequences (a.k.a. iterations) are carried out and then averaged before the signals being classified. However, while augmenting the number of iterations improves the Signal-to-Noise Ratio, it also slows down the process. In the early studies, the number of iterations was fixed (no stopping environment), but recently several early stopping strategies have been proposed in the literature to dynamically interrupt the stimulation sequence when a certain criterion is met in order to enhance the communication rate. In this work, we explore how to improve the classification performances in P300 based BCIs by combining optimization and machine learning. First, we propose a new decision function that aims at improving classification performances in terms of accuracy and Information Transfer Rate both in a no stopping and early stopping environment. Then, we propose a new SVM training problem that aims to facilitate the target-detection process. Our approach proves to be effective on several publicly available datasets.

Keywords Brain computer interface · MILP mixed integer linear programming · P300 Speller · Support vector machine

1 Introduction

A Brain-Computer Interface (BCI) is a system that records a user's brain activity and allows him to interact with the environment by exploiting both signal processing and machine

Extended author information available on the last page of the article

learning algorithms. In most cases, the recorded signals are noisy, so that filtering or averaging techniques are used to improve the signal-to-noise ratio (SNR). The information embedded in signals that are relevant to characterize the user's mental states are then selected during a feature extraction procedure before being classified and translated into artificial outputs—i.e. into control commands for an output device such as a pointer, a keyboard or a robotic arm (Lotte 2014; Lotte et al. 2018; Quitadamo et al. 2008; Wolpaw and Wolpaw 2012; McCane et al. 2015). BCIs use either electrical, magnetic and metabolic signals (Wolpaw and Wolpaw 2012) recorded with methods such as electroencephalography (EEG), electrocorticography (ECoG), magnetoencephalography (MEG), functional Near Infra-Red Spectroscopy (fNIRS) and functional Magnetic Resonance Imaging (fMRI). In particular, EEG represents one of the most used methods since they are non invasive and inexpensive; for this reason they have been used for a wide variety of tasks (Chaovalitwongse et al. 2006; Khojandi et al. 2019).

In this framework, BCIs based on event-related potentials (ERPs) have proven to be particularly successful and robust (Schreuder et al. 2013). ERPs represent the electrical responses recorded from the brain through EEG techniques after specific events or stimuli. The ERPs are embedded within the general EEG activity (Sur and Sinha 2009), and are time-locked to the processing of a specific stimulus. As their amplitude is lower that the one of the ongoing EEG activity, averaging techniques are employed to increase the SNR: in principle, averaging background noise which is not correlated to an event, such as the ongoing EEG activity, tends to reduce its contribution to a small offset, which can be easily filtered out, while the evoked responses, supposed to be the same after each stimulus, are left unmodified. An ERP-based BCI attempts to detect ERP components to infer the stimulus that the user intended to choose—i.e. the stimulus eliciting the ERP components (Treder and Blankertz 2010; Shahriari et al. 2019).

In 1988, the P300 ERP was first used by Farwell and Donchin within a BCI system (Farwell and Donchin 1988). Their P300 Speller consists of 36 alpha-numeric characters arranged within the rows and columns of a 6×6 matrix. The user's task is to focus the attention on a specific character—i.e. on one of the cells of the matrix. Each of the 6 rows and 6 columns then flashes for few tenths of milliseconds in a random sequence. A sequence of 12 different flashes—the 6 rows and 6 columns—is called an iteration. It constitutes the basis of an *oddball paradigm* in which two classes of stimuli, namely the target (or rare) and the non target (or frequent) which occur with different probabilities (0.166 and 0.833 in this case), elicit two different brain responses. In particular, the target (rare) stimuli should elicit the P300 response, which should not be evoked after a non target (frequent) stimuli. In our case the row and the column containing the attended character represent the target stimuli, while the other ten are the non-target ones. Brain responses to the target and non-target stimuli are distinguished using a classification algorithm. The correct identification of the target row and column allows the desired character's selection, which is located at their intersection (Krusienski et al. 2006, 2008; Sellers et al. 2006).

Later on in the literature, different variations of the original P300 paradigm have been developed in order to improve the speller framework. For instance, in Schaeff et al. (2012), Schreuder et al. (2011), Treder et al. (2011) the authors proposed gaze-independent spellers, i.e. communication systems that can be used by subjects who have impairment at moving their eyes. In all speller paradigms, given a sentence/run to copy-spell, the EEG data are organized in terms of trials, iterations, and sub-trials. A single character selection step is here referred to as a trial. Each trial consists of several iterations/stimulation sequences, during which all the stimuli are intensified once in a pseudo-random order. A single stimulus intensification is here referred to as a sub-trial. The trials' selection process usually involves one or two levels. In the former case, symbols are typically presented successively

thus involving a single selection step. In the latter, the user has to select a group of symbols at the first level and then the target symbol at the second level.

To use a BCI, two phases namely training/calibration and test/online are typically required. During the calibration phase, the user is instructed to focus his/her attention on a specific character (copy task) for which correct labels are then a priori known. The acquired EEG signals are then preprocessed by filtering. A subset of EEG features is extracted to represent the signal in a compact form. The obtained EEG patterns are recognized using a classification algorithm, which is trained on the subset of identified features to determine the presence or the absence of a P300 evoked potential. In the online phase, new EEG patterns are classified using the trained model before being translated into a command for an application. As described above, in ERP-based BCIs, to perform a single selection step, multiple iterations are carried out to improve the SNR. Since each iteration takes about 3 seconds to be completed, this strategy increases the time needed to detect brain signals thus affecting down the communication rate. To overcome this drawback, different early stopping (ES) or Dynamic Stopping methods have been introduced, where after a calibration phase, a suitable termination criterion is established to be tested online when the number of iteration is sufficient to ensure a reliable classification. In this work, we explore how to improve the classification performance by combining optimization and machine learning both in the classical setting with a fixed number of repetitions and in the early stopping setting.

1.1 Literature review

As mentioned above, the presence or the absence of a P300 evoked potential within the EEG features is determined using a classification algorithm (Krusienski et al. 2006).

Formally, the detection of brain responses to the target and non-target stimuli can be translated into a binary classification problem. Let TS be the training set defined as:

$$\begin{aligned} TS = \{&(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}, \\ &i := (k, r, t, f) \quad \forall k = 1 \dots n_k,\ r = 1 \dots n_r,\ t \in T,\ f = 1 \dots n_f\} \end{aligned} \tag{1}$$

where n_k denotes the total number of trials in the training phase and n_r denotes the number of iterations for each trial; the number of flashes n_f and the set of levels T together denote the set of possible stimuli that compose the stimulation sequence (i.e. $n_f = 6$ and $T = \{row,\ column\}$ for P300 Speller's paradigm or $T = \{outer,\ inner\}$ for two-levels paradigms).

During the calibration phase, a classification algorithm is trained over TS to learn the discriminant function f such that

$$f(x) = y, \tag{2}$$

and this function is used in the online phase to spell words or sentences. In the BCI literature, several algorithms have been proposed for addressing this classification problem (Lotte et al. 2018). In particular, linear classifiers such as stepwise linear discriminant analysis (SWLDA) (Draper and Smith 1998), and support vector machine (SVM) (Friedman et al. 2001) are still the most used discriminant algorithms for ERPs' classification (Lotte et al. 2018). These methods classify the brain responses by means of a separating hyperplane (Krusienski et al. 2006). This discriminant function is built on the basis of the training data, and it is defined as:

$$f(x) = w^T x + b, \tag{3}$$

where w is the vector containing the classification weights and b is the bias term. Linear classifiers differ in the way they learn w and b (Krusienski et al. 2006). In (3), the right-hand side is called decision value. Its absolute value is proportional to the distance of the sample points x from the separating hyperplane.

In a standard binary classification problem, for each instance the class label is assigned based on the sign of the relative decision value. However, in a classical P300 Speller (Farwell and Donchin 1988), based on the assumption that a P300 is elicited for one of the six row/column stimuli and finding that the P300 response is invariant to row/column stimulation, the target class is assigned to the stimuli matching the maximum decision values for both the rows and the columns (Krusienski et al. 2006). In general, recalling the definition of T and n_f given in (1), we can identify the target stimulus for trial $k \in \{1 \dots n_k\}$ and iteration $r \in \{1 \dots n_r\}$ as:

$$\text{predicted stimulus}_{(k,r,t)} = \underset{f=1\dots n_f}{\text{argmax}} \left[w^T x_{(k,r,t,f)} + b \right] \quad \forall t \in T \tag{4}$$

The predicted character for trial $k \in \{1 \dots n_k\}$ and iteration $r \in \{1 \dots n_r\}$ is then identified by combining the predicted target stimuli found $\forall t \in T$ (i.e a row target and a column target for the standard P300 paradigm).

As mentioned in Sect. 1, for each character, data recorded from multiple iterations have to be integrated to improve the SNR. To the best of our knowledge there exist two main different iteration-averaging strategies in the literature: (i) **ERP avg**: for each character brains responses to target and non target stimuli are averaged across the iterations before being classified, and (ii) **DV avg**: for each character the decision values of each target and non-target stimulus are averaged across the iterations before assigning the target class. Recently in (Bianchi et al. 2019), a new classification function namely score-based function (SBF) has been introduced for integrating brain responses recorded from multiple iterations. For each character, the SBF exploits a set of heuristically-determined scores to weight each stimulus according to its decision value. For each stimulus, the assigned scores are summed up iteration by iteration. The target class (one for the row and one for the column) is assigned to the stimulus having the highest total score after the last available iteration. The SBF has been introduced for developing an early stopping method (ESM)—i.e. an automatic method that interrupts the stimulation at any point in a trial when a certain criterion, based on the ongoing classification results, is satisfied (see for instance Lenhardt et al. 2008; Zhang et al. Jun 2008; Liu et al. 2010; Höhne et al. 2010; Schreuder et al. 2011; Jin et al. 2011; Throckmorton et al. 2013; Mainsah et al. 2014; Jiang et al. 2018; Vo et al. 2017, 2018; Schreuder et al. 2013; Kha et al. 2017; Gu et al. 2019; Huang et al. 2020). The proposed ESM based on the SBF outperformed the current state-of-the-art early stopping methods proposed in Schreuder et al. (2013). Note that the SBF is *quasi*-opposite to the approach proposed in Kha et al. (2017) where a score is assigned to an SVM based classifier and then an Early Stopping is defined using the cumulative scores of the classifiers. In Huang et al. (2020), an Early Stopping technique is used to improve the accuracy of the P300 speller while it is performing other tasks. Thus the proposed approach allows adapting the classification accuracy to the subject's attention level in real-time. In this paper, we follow the same line of research of Bianchi et al. (2019), by making some further steps to include the information on the protocol into the classification phase. Indeed, the novelty of our approach consists of three points:

1. determine the optimal scores for each participant by solving an optimization problem on her/his training data;

2. solve a modified version of the optimization problem in order to implement an efficient early stopping method;
3. include the information on the decision function (the target is the stimulus having maximum decision value) into the training problem

The great advantage of our method is that the calibration phase (different for each participant) becomes completely automatic and does not need any cross validation phase or manual parameters tuning.

The paper is structured as follows: in Sect. 2, we introduce our new decision function, defining the optimization problems to be solved both in the no stopping and early stopping scenario. In Sect. 3, we introduce a new training problem that keeps into account explicitly the target assignment in BCI, and in Sect. 4 we derive its Wolfe dual. In Sect. 6 we report the behavior of our new approaches on several datasets and finally we draw some conclusions in Sect. 7. In "Appendix 1" we quickly describe the algorithm for solving the dual of our new training problem, while in Appendix 2 we report the detailed numerical results for all the datasets.

2 An optimized score based decision function (OSBF)

In Bianchi et al. (2019), a set of heuristically-determined scores has been used to weight and combine the decision values of multiple iterations within an early stopping setting. In this work, we decided to modify the approach by using a set of scores automatically determined by solving a mixed integer linear programming (MILP) problem for each participant. Each stimulus receives a weight according to its decision value: five zones are defined, and each zone gets a different score *a,b,c,d,e*. In particular, the scores are related to the confidence in the classification of the given stimulus as target: the score a is assigned to the stimulus that is most likely to be the target, whereas the stimuli that are highly unlikely to be the target get score e. All the stimuli in the middle get decreasing scores according to the distribution of the decision values.

The zones are identified by considering the decision values of all iterations for all stimuli in the training set and computing the corresponding quartiles Q1, Q2 and Q3. The idea is to produce scores that reflect the distribution of the data.

Figure 1 shows how the scores are assigned depending on the distribution of the quartiles of the decision values in a simple 2-dimensional example. The maximum score a is assigned only if the confidence in the current classification is extremely high: i.e. if the decision value is positive and higher than all the other decision values of the current iteration.

Note that, given the separating hyperplane, the score assignment for each stimulus of each character is known: so, it is possible to build the following binary vectors that represent in a compact form the score vector assignment z for each stimulus of each character:

$$z_s^{k,r,t,f} = \begin{cases} 1 \text{ if stimulus } f \text{ of level } t \text{ gets score } s \text{ at iteration } r \text{ for char } k \\ 0 \text{ otherwise} \end{cases}$$

where $f = 1 \dots n_f$ and $t \in T$ identify the stimulus, $k = 1 \dots n_k$ identifies the character, $r = 1 \dots n_r$ identifies the iteration and, finally, $s = a, \dots, e$ identifies the score. The score assignments depends on the primary aim of the BCI:

(i) if the main focus is the accuracy, the idea is to use all the available iterations for spelling a character (no stopping protocol), also in the online phase.

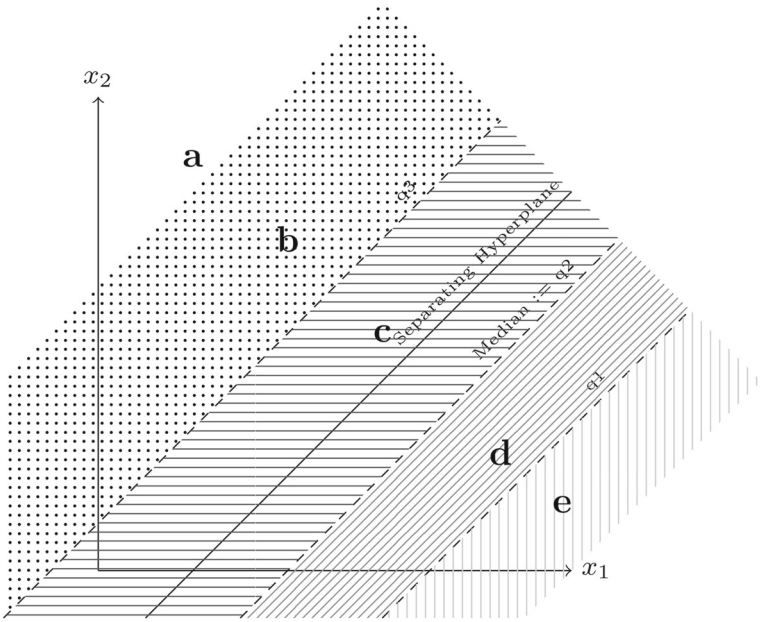

Fig. 1 Graphical representation of the score distribution, reflecting the displacement of the points w.r.t the distribution of the decision values. The different areas represent the confidence of the classification w.r.t the target class. Please note that this is a simplified example where samples are represented as 2-dimensional data points with features x_1 and x_2

(ii) if the idea is to try and speed up the communication, then the performance to be maximized is the transmission rate, trying to reduce the number of iterations needed to spell a character in the online phase (early stopping).

In the next two subsections, we describe the Mixed Integer Linear Programming (MILP) Problems we define in order to find the scores in the two different settings.

2.1 No stopping OSBF

First, we propose a strategy to choose the scores when all the iterations are exploited and the primary focus is to increase the classification accuracy. In this setting, we aim at reliability of the classification and we do so by imposing the following constraints:

1. at the last iteration, we require, if possible, that the score obtained by the target stimulus is larger (with some margin if possible, that implies robustness of the classification) than the score of any non target stimulus. This means that we ask not to fail in the classification after the last available iteration; if this is not possible, a suitable binary variable representing the failure on that stimulus is set to one;
2. to make the classification more robust on the test set, we require that in as many iterations as possible, the score of the target is larger than the one of the non target stimuli;
3. as an objective, we try and maximize the accuracy on the training set, and the number of iteration where the classification is robust.

Our main variable in the optimization problem is the vector of scores $s = (a\ b\ c\ d\ e)^T$.

Table 1 Description of the set of indexes, parameters and variables used in the no stopping and early stopping OSBF

Set		Description
Set of indexes		
K		Set of trials in the offline phase
R		Set of iterations that compose each trial
T		Set of levels of the paradigm
F		Set of stimuli in the stimulation sequence
Name	**Set of Indexes**	**Description**
Parameters		
z	K, R, T, F	Binary vector of 5 components which represents the zones partition. The i-th component of $z^{k,r,t,f}$ is set to 1 if and only if stimuli (k,r,t,f) is assigned to the i-th confidence zone
Name	**Set of Indexes**	**Description**
Variables		
x	K, R, T	Binary variable which equals 1 if the target stimuli is correctly detected at current iteration in no stopping OSBF. For early stopping OSBF, this variable equals 1 if the early stopping condition is verified for the first time on the target at the current iteration, and it is not satisfied by any non target stimulus earlier
err	K, T	Binary variable which equals 1 if the target stimuli was not correctly detected at the last iteration possible
s		Score vector $s = \{a, b, c, d, e\}$
Δ		Reliability Threshold

We add an auxiliary variable to try and impose some distance between the score of the target stimulus and the scores of the non target stimuli that we call Δ, and that represents a measure of reliability of the classification. Further, we add some binary variables:

- $x_t^{k,r}$: binary variable that is equal to 1 if the target of character k for level t has a score at iteration r that is larger than the score of any non target stimulus plus Δ
- err_t^k: binary variable that is equal to 1 if the target is not correctly classified for character k at level t, i.e. if at the last iteration the target score is lower or equal to the score of some non target stimulus

The MILP problem to be solved is then the following:

$$\max \sum_{t \in T} \left(\left(1 - \frac{\sum_{k=1}^{n_k} err_t^k}{n_k} \right) + \frac{1}{n_k n_r} \sum_{k=1}^{n_k} \sum_{r=1}^{n_r} x_t^{k,r} \right) \tag{5}$$

$$s_1 \leq u \tag{6}$$

$$s_{j+1} \leq s_j - 1 \, \forall j = 1, \ldots, 4 \tag{7}$$

$$s_3 \geq 0 \tag{8}$$

$$s_5 \geq l \tag{9}$$

$$\Delta \geq s_1 - s_5 + 1 \tag{10}$$

$$1 - err_t^k \leq x_t^{k,nr} \; \forall k \in \{1 \dots n_k\}, \; \forall t \in T \tag{11}$$

$$\sum_{r=1}^{\bar{r}} s^T z^{k,r,t,f} + \Delta \leq \sum_{r=1}^{\bar{r}} s^T z^{k,r,t,trg(k,t)} + M(1 - x_t^{k,\bar{r}}) \quad \forall k, \bar{r}, t, f : f \neq trg(k,t) \tag{12}$$

$$err_t^k, x_t^{k,r} \in \{0,1\} \quad \forall k \; \forall t \; \forall r \tag{13}$$

where l and u are chosen bounds on the possible values of the scores, and M is large enough to make the constraints trivially satisfied when the corresponding binary variable $x_t^{k,r}$ is zero. The objective function, that has to be maximized, is composed by two terms: the percentage of success on the training set, and the average number of iterations where the classification is robust and reliable. We then have the following constraints:

(i) Constraints (6), (7) and (9) impose that the scores are bounded and that are ordered in decreasing order and differ of at least one; whereas constraint (8) imposes that the first three scores are nonnegative

(ii) constraint (10) imposes a lower bound on the threshold to ensure reliability of the classification. Indeed this lower bound ensures that the threshold has a minimum value depending on the scores: in particular $s_1 - s_5 + 1$ represents the maximum difference in score that can be assigned to different flashes in a single iteration. Therefore, even in the worst possible scenario, where two flashes get the same score, there must be at least one iteration where one gets the maximum score and the other the minimum score to break the parity.

(iii) constraints (11) impose that variable err_t^k is 1 if and only if $x_t^{k,n_r} = 0$, that is it represents an unreliable classification at the last iteration.

(iv) constraints (12) impose that if at iteration $\bar{r}$ the classification is reliable for the target $trg(k,t)$ of character k at level t, then the corresponding binary variable $x_t^{k,r}$ is set to 1.

2.2 Early stopping OSBF

Problem (5) can be modified in order to improve the system performance in terms of speed, implementing an automatic Early Stopping Method, similarly to Bianchi et al. (2019).

The idea is again to use the scores s and the threshold Δ at each iteration of the test phase to verify an early stopping condition: during the test phase, the stimuli are ordered according to the sum of their scores and, if the difference in score between the first and second stimulus is greater than the threshold Δ, the method classifies the target character and the remaining iterations are not performed.

In order to adapt problem (5) to the early stopping setting, we introduce some further constraints, and modify the meaning of some binary variables:

$$\max \sum_{t \in T} \left(1 - \frac{1}{n_k} \sum_{k=1}^{n_k} err_t^k - \frac{100 \times n_{fl}}{60} \frac{SOA}{n_k} \left(\sum_{k=1}^{n_k} \sum_{r=1}^{n_r} r x_t^{k,r} + n_r \sum_{k=1}^{n_k} err_t^k \right) \right) \tag{14}$$

$$(6) - (10)$$

$$1 - err_t^k \leq \sum_{r=1}^{n_r} x_t^{k,r} \; \forall k \in \{1 \dots n_k\} \tag{15}$$

$$\sum_{r=1}^{n_r} x_t^{k,r} \leq 1 \; \forall k \in \{1 \ldots n_k\} \tag{16}$$

(12)

$$\sum_{r=1}^{\bar{r}} s^T z^{k,r,t,f} \geq \sum_{r=1}^{\bar{r}} s^T z^{k,r,t,f} - \Delta + 1 \tag{17}$$

$$- M(1 - x_t^{k,\bar{r}}) \quad \forall k, \forall t, \forall \tilde{r} < \bar{r}, \forall f : \; f \neq trg(k,t)$$

$$err_t^k, x_t^{k,r} \in \{0,1\} \quad \forall k \; \forall t \; \forall r \tag{18}$$

In this case, the objective function keeps into account both the percentage of success (to be maximized) and the time needed for classification (to be minimized). Note that the second term (which represents the trial duration in minutes) was multiplied by a factor 100 to make the two terms of the objective function comparable. We then have some further constraints, since in this case we are interested in the first iteration where the following early stopping condition is met:

$$\sum_{r=1}^{\bar{r}} s^T z^{k,r,t,f} + \Delta \leq \sum_{r=1}^{\bar{r}} s^T z^{k,r,t,trg(k,t)}. \tag{19}$$

In this model, we set the binary variables $x_t^{k,r}$ in such a way that it is 1 if and only if the early stopping condition (19) is verified for the first time on the target at iteration r, and it is not satisfied by any non target stimulus earlier. This is imposed by the combination of constraints (12), (16) and (17).

We stress that in both the no stopping and the early stopping scenarios, the MILP problem is solved using the training set data (the same used to build the hyperplane), whereas the score efficiency is evaluated on the test set.

3 A new training problem

As already pointed out in the introduction, in order to achieve a good classification accuracy it is fundamental to exploit the information that at each iteration there is exactly one target stimulus for each level, assigning then the target class to the stimulus having the maximum decision value. Our idea is to try and add this protocol knowledge already in the training problem.

Given the definition (1) of training set, the standard training problem to solve in order to find a separating hyperplane according to the SVM approach is the following (Piccialli and Sciandrone 2018):

$$\begin{aligned} \min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \; & \frac{1}{2}\|w\|^2 + C_1 \sum_{i \in TS} \xi_i \\ & y_i(w^T x_i + b) \geq 1 - \xi_i \qquad \forall i \in TS \\ & \xi_i \geq 0 \qquad \forall i \in TS \end{aligned}$$

In this work, we modify the training problem including the information that the target stimuli should receive the maximum decision value among all the other flashes. Let's denote by trg_i the target stimulus for the stimulation sequence where the stimulus i belongs: so, in

particular, if $i = (k, r, t, f)$ we will have:

$$trg_i = (k, r, t, f') \in TS \quad \& \quad y_{trg_i} = 1$$

Then, we want to impose:

$$w^T x_{(k,r,t,f_1)} + b \geq w^T x_{(k,r,t,f_2)} + b \qquad \forall k, r, t, f_1, f_2 : \\ y_{(k,r,t,f_1)} = 1 \,\&\, \quad y_{(k,r,t,f_2)} = -1 \tag{20}$$

From now on, in order to simplify the notation, we will write constraints (20) in the following more compact form:

$$w^T x_{trg_i} + b \geq w^T x_i + b \qquad \forall i \in TS : y_i \neq 1 \tag{21}$$

and we add slack variables to avoid infeasibility, getting the following set of constraints:

$$w^T x_{trg_i} - w^T x_i \geq 1 - \eta_i \quad \forall i \in TS : y_i \neq 1 \tag{22}$$

$$\eta_i \geq 0 \quad \forall i \in TS : y_i \neq 1 \tag{23}$$

Now we simply plug these constraints into the primal problem getting the new training problem based on the maximum decision function:

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2}\|w\|^2 + C_1 \sum_{i \in TS} \xi_i + C_2 \sum_{i \in TS}^{y_i \neq 1} \eta_i \tag{24}$$

$$y_i (w^T x_i + b) \geq 1 - \xi_i \quad \forall i \in TS \tag{25}$$

$$w^T z_i \geq 1 - \eta_i \quad \forall i \in TS : y_i \neq 1 \tag{26}$$

$$\xi_i \geq 0 \quad \forall i \in TS \tag{27}$$

$$\eta_i \geq 0 \quad \forall i \in TS : y_i \neq 1 \tag{28}$$

where the vector z is defined as:

$$z_i = x_{trg_i} - x_i \quad \forall i \in TS : y_i \neq 1$$

4 Wolfe Dual of the new training problem

In order to build the Wolfe Dual of the quadratic optimization problem (24)–(28), it is necessary to introduce the dual multipliers of the constraints:

- $\lambda_i \quad \forall i \in TS$: the multiplier associated to constraints (25)
- $\rho_i \quad \forall i \in TS : y_i \neq 1$: the multiplier associated to constraints (26)
- $\mu_i \quad \forall i \in TS$: the multiplier associated to constraints (27)
- $\theta_i \quad \forall i \in TS : y_i \neq 1$: the multiplier associated to constraints (28)

Let us define the vector λ and ρ as the vectors of size l_1 and l_2 respectively containing λ_i ($\forall i \in TS$) and ρ_i ($\forall i \in TS : y_i \neq 1$). Then we define the following matrix $\Sigma \in \Re^{(l1+l2)\times n}$:

$$\Sigma = \begin{pmatrix} y^1(x^1)^T \\ \vdots \\ y^{l_1}(x^{l_1})^T \\ (z^1)^T \\ \vdots \\ (z^{l_2})^T \end{pmatrix}$$

The following proposition holds:

Proposition 1 *The dual problem of problem (24) is*

$$\min \frac{1}{2} \begin{pmatrix} \lambda^T & \rho^T \end{pmatrix} \Sigma \Sigma^T \begin{pmatrix} \lambda \\ \rho \end{pmatrix} - e^T \lambda - e^T \rho \tag{29}$$

$$y^T \lambda = 0 \tag{30}$$

$$0 \leq \lambda \leq C_1 e \tag{31}$$

$$0 \leq \rho \leq C_2 e \tag{32}$$

Proof The Wolfe dual of problem (24)–(28) is given by:

$$\max_{w,b,\lambda,\rho,\mu,\theta} L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) \tag{33}$$

$$\nabla_w L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = 0 \tag{34}$$

$$\nabla_b L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = 0 \tag{35}$$

$$\nabla_\xi L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = 0 \tag{36}$$

$$\nabla_\eta L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = 0 \tag{37}$$

$$\lambda, \rho, \mu, \theta \geq 0 \tag{38}$$

where $L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta)$ is the Lagrangian of optimization problem (24)–(28) that can be expressed as follows:

$$\begin{aligned} L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = \frac{1}{2}\|w\|^2 + C_1 \sum_{i \in TS} \xi_i + C_2 \sum_{i \in TS}^{y_i \neq 1} \eta_i \\ - \sum_{i \in TS} \lambda_i \left(y_i (w^T x_i + b) - 1 + \xi_i \right) + \\ - \sum_{i \in TS}^{y_i \neq 1} \rho_i \left(w^T z_i - 1 + \eta_i \right) - \sum_{i \in TS} \mu_i \xi_i - \sum_{i \in TS}^{y_i \neq 1} \theta_i \eta_i \end{aligned} \tag{39}$$

By rearranging terms equation 39 can be rewritten as:

$$\begin{aligned} L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = \frac{1}{2}\|w\|^2 + \sum_{i \in TS} \xi_i (C_1 - \lambda_i - \mu_i) + \sum_{i \in TS} \lambda_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i + \\ + \sum_{i \in TS}^{y_i \neq 1} \eta_i (C_2 - \rho_i - \theta_i) - w^T \left(\sum_{i \in TS} \lambda_i y_i x_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i z_i \right) - b \sum_{i \in TS} \lambda_i y_i \end{aligned} \tag{40}$$

The constraints of the Wolfe Dual (equations 34-37) can now be computed based on the Lagrangian function in equation 40. The equation $\nabla_w L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = 0$ leads to an expression for w:

$$w = \left(\sum_{i \in TS} \lambda_i y_i x_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i z_i \right), \tag{41}$$

whereas the equation $\nabla_b L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) = 0$ leads to the constraint

$$\sum_{i \in TS} \lambda_i y_i = 0 \tag{42}$$

Equation $\frac{\partial L(w,b,\xi,\eta,\lambda,\rho,\mu,\theta)}{\partial \xi_i} = 0$ allows to derive μ_i as a function of λ:

$$C_1 - \lambda_i - \mu_i = 0 \qquad \forall i \in TS \tag{43}$$

whereas $\frac{\partial L(w,b,\xi,\eta,\lambda,\rho,\mu,\theta)}{\partial \eta_i} = 0$ results in an expression of θ_i as a function of ρ_i

$$C_2 - \rho_i - \theta_i = 0 \qquad \forall i \in TS : y_i \neq 1 \tag{44}$$

Non-negativity of the multipliers $\lambda,\ \rho,\ \mu,\ \theta$ combined with equations (43) and (44) result in the following set of constraints:

$$0 \leq \lambda_i \leq C_1 \qquad \forall i \in TS \tag{45}$$

$$0 \leq \rho_i \leq C_2 \qquad \forall i \in TS : y_i \neq 1 \tag{46}$$

We can plug equations (43) and (44) in the objective function, getting:

$$\begin{aligned} L(w, b, \xi, \eta, \lambda, \rho, \mu, \theta) &= \frac{1}{2}\|w\|^2 + \sum_{i \in TS} \xi_i (C_1 - \lambda_i - \mu_i) + \sum_{i \in TS} \lambda_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i + \\ &+ \sum_{i \in TS}^{y_i \neq 1} \eta_i (C_2 - \rho_i - \theta_i) - w^T \left(\sum_{i \in TS} \lambda_i y_i x_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i z_i \right) - b \sum_{i \in TS} \lambda_i y_i = \\ &= \frac{1}{2}\|w\|^2 + \sum_{i \in TS} 0 \times \xi_i + \sum_{i \in TS} \lambda_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i + \sum_{i \in TS}^{y_i \neq 1} 0 \times \eta_i - w^T w - 0 \times b = \\ &= -\frac{1}{2}\|w\|^2 + \sum_{i \in TS} \lambda_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i \end{aligned} \tag{47}$$

The Wolfe Dual of problem (24)–(28) can then be expressed by using equation (47) as objective and equations (41), (42), (45), (46) as constraints.

$$\min \frac{1}{2}\|w\|^2 - \sum_{i \in TS} \lambda_i - \sum_{i \in TS}^{y_i \neq 1} \rho_i \tag{48}$$

$$w = \left(\sum_{i \in TS} \lambda_i y_i x_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i z_i \right) \tag{49}$$

$$\sum_{i \in TS} \lambda_i y_i = 0 \tag{50}$$

$$0 \leq \lambda_i \leq C_1 \qquad \forall i \in TS \tag{51}$$

$$0 \leq \rho_i \leq C_2 \qquad \forall i \in TS : y_i \neq 1 \tag{52}$$

Note that:

$$\|w\|^2 = w^T w = \left(\sum_{i \in TS} \lambda_i y_i x_i + \sum_{i \in TS}^{y_i \neq 1} \rho_i z_i \right)^T \left(\sum_{i' \in TS} \lambda_{i'} y_{i'} x_{i'} + \sum_{i' \in TS}^{y_{i'} \neq 1} \rho_{i'} z_{i'} \right) =$$

$$\sum_{i \in TS} \sum_{i' \in TS} (\lambda_i \lambda_{i'} y_i y_{i'} (x_i)^T x_{i'}) + \sum_{i \in TS}^{y_i \neq 1} \sum_{i' \in TS}^{y_{i'} \neq 1} (\rho_i \rho_{i'} (z_i)^T x_{i'}) + 2 \sum_{i \in TS} \sum_{i' \in TS}^{y_{i'} \neq 1} (\lambda_i \rho_{i'} y_i (x_i)^T z_{i'})$$

Let us define the vector λ and ρ as the vectors of size l_1 and l_2 respectively containing λ_i $(\forall i \in TS)$ and ρ_i $(\forall i \in TS : y_i \neq 1)$. Then we define the following matrix $\Sigma \in \Re^{(l1+l2)\times n}$:

$$\Sigma = \begin{pmatrix} y^1 (x^1)^T \\ \vdots \\ y^{l_1} (x^{l_1})^T \\ (z^1)^T \\ \vdots \\ (z^{l_2})^T \end{pmatrix}$$

The dual problem can then be rewritten as

$$\min \frac{1}{2} \begin{pmatrix} \lambda^T & \rho^T \end{pmatrix} \Sigma \Sigma^T \begin{pmatrix} \lambda \\ \rho \end{pmatrix} - e^T \lambda - e^T \rho \tag{53}$$

$$y^T \lambda = 0 \tag{54}$$

$$0 \leq \lambda \leq C_1 e \tag{55}$$

$$0 \leq \rho \leq C_2 e \tag{56}$$

that is still a quadratic convex programming problem.

5 Algorithmic framework

In Fig. 2, we summarize the proposed approach, which takes in input the EEG signals of a P300 Speller task for a subject and outputs the performance reached in terms of accuracy and ITR. The same pipeline is used within both the no stopping and the early stopping setting.

This generic framework allows for performing several design choices for both the preprocessing phase and the hyperplane construction. Details on how we preprocessed our datasets are reported in Sect. 6.1. Regarding the hyperplane construction method, we used both the standard SVM problem and the training problem proposed in Sect. 3.

Our framework requires to execute an offline phase in which the hyperplane construction problem is solved and the score vector is computed by using one of the MILP problems proposed in Sects. 2.1 (no stopping setting) and 2.2 (early stopping setting). In the early stopping setting the MILP problem will also output the reliability threshold Δ. Solving the MILP problems requires to partition the decision values of offline data in 5 confidence zones (as specified in Fig. 1) which are retrieved on the basis of the distribution of the decision values. The output of the offline phase is composed both by the computed hyperplane, the scores vector and eventually the threshold Δ. These values are then used in an online scenario in order to evaluate subject's performance both in terms of accuracy (%) and Information Transfer Rate (bit/min), which is of particularly interest in the early stopping setting.

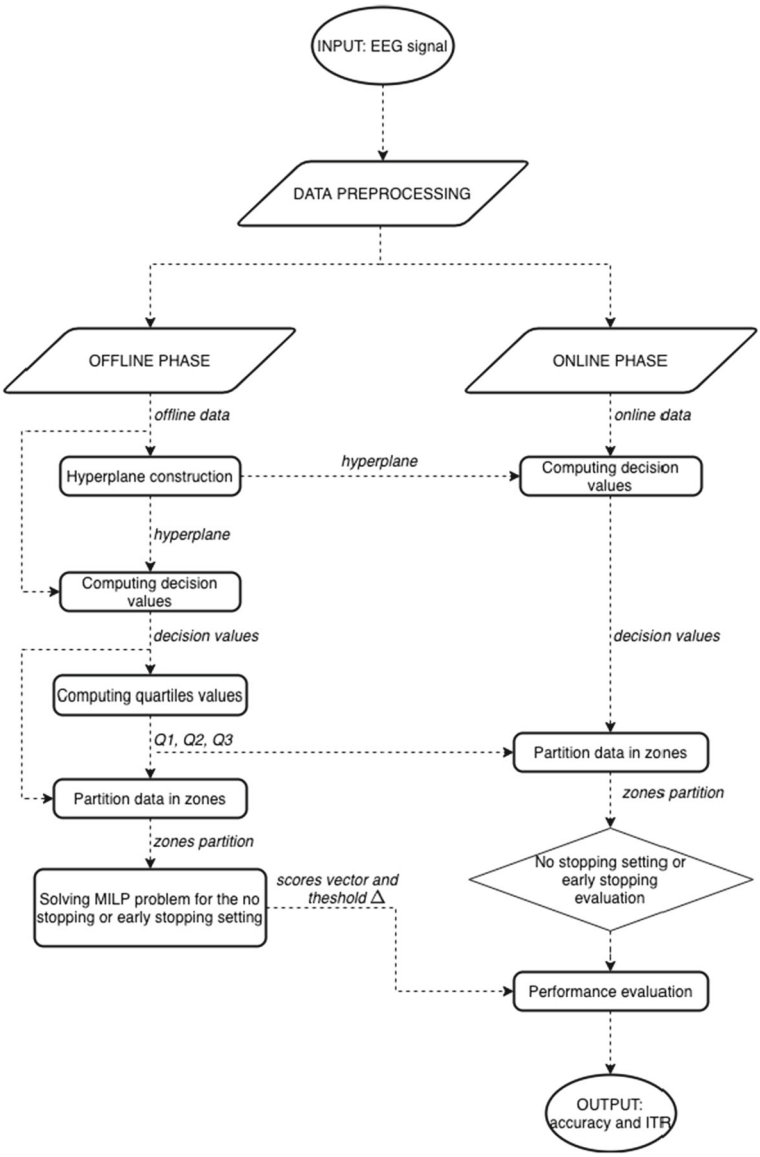

Fig. 2 Flowchart of the proposed algorithmic framework , which takes in input the EEG signal of a subject in a P300 Speller task. The proposed framework requires to perform an offline phase in order to instantiate the process. Results from the offline phase are then used to automatically decode online EEG signals and evaluate the performance results reached by the subject. This pipeline is valid both in a no stopping and early stopping environment; in the latter case, in the online phase the early stopping condition will be evaluated at every incoming iteration before computing the performances obtained for the subject

6 Numerical results

6.1 Dataset

We tested our approaches on five different datasets:

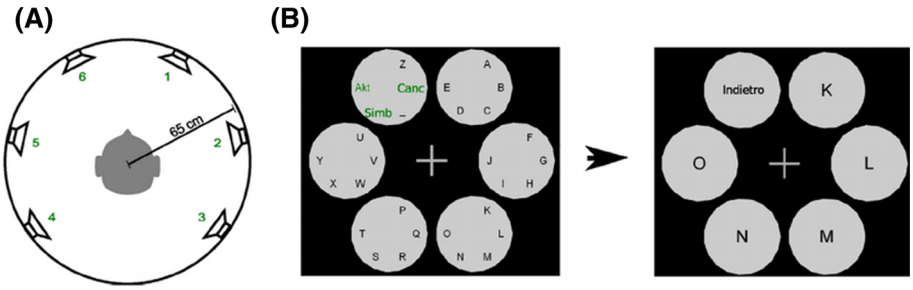

Fig. 3 Graphical representation of the AMUSE paradigm in which six speakers are places all over the subject. In the first level, each speaker is used to represent a set of characters, while on the second level each speaker is used to represent a single character among the previously selected set

AMUSE	The protocol is based on auditory stimulus elicited by means of spatially located speakers, we have two levels, 15 rounds, six classes for each level, see Fig. 3 (Schreuder et al. 2011). It is performed on healthy subjects and downloadable by the BNCI horizon website http://bnci-horizon-2020.eu/database/data-sets.
P300 Speller	The protocol is the classical P300 Speller (Farwell and Donchin 1988), performed on 10 healthy subjects.
ALS P300 Speller	The protocol is the classical P300 Speller (Farwell and Donchin 1988), performed on 8 patients suffering of Amyotrophic Lateral Sclerosis (ALS).
MVEP	It is a visual protocol in which a moving pattern generates a movement-onset visual evoked potential that is used to recognize the user's choice. This protocol is based on modifications of Cake Speller protocol (Treder et al. 2011). Sixteen healthy subjects have been involved in the study.
Center Speller	It is a visual protocol where we have a visual stimulus elicited by means of three different stimuli, two levels, 10 rounds, six classes for each level (Treder et al. 2011). It is performed on 13 healthy subjects.
Akimpech	It is a P300 Speller performed on 27 healthy subjects, the number of characters is 16 with 15 iterations for each character in the calibration phase, whereas in the online phase changes depending on the subject.

Details of the datasets are reported in Table 2. Please note that we have evaluated our strategy on EEG data recorded from 95 subjects thus assessing its generalization capabilities.

All EEG signals were pre-processed and features were extracted with the NPXLab Suite (Bianchi 2018). Two principal pre-processing operations were applied:

- Electrodes selection: for the datasets Center Speller, MVEP, and AMUSE (see Sect. 6.1) we kept just the electrodes belonging to the 10-20 EEG placement. This strategy allows us to reduce both the dimension of the dataset and the overfitting;
- k-decimation: this technique was applied to all datasets in order to reduce overfitting. In this case, we down-sampled the EEG signal from every electrode by replacing each k consecutive samples with their average value.

Let's recall that the OSBF strategy requires to compute the quartiles of the training set decision values in order to assign scores to stimuli. In this scenario, we stress that, for the standard P300 Speller's paradigm, stimuli corresponding to the intensification of rows and columns are considered separately; in fact, we observed that the distribution of the decision

Table 2 Dataset parameters. The following characteristics are reported: number of subjects (NS), total number of trails in the training set, total number of trials in the test set, type of paradigm, participants (part. H = healthy, ALS = amyotrophic lateral sclerosis patient), number of sensors (Sens.), modality (mod. A = auditory, V = visual), number of possible symbols (Symb.), total number of stimuli in the selection process (for all possible levels), the maximum number of iterations in the original setting, the SOA (stimulus onset asynchrony), the overhead (OH pre and post-stimulus pauses), the value of k used in the k-decimation preprocessing operation

Dataset	NS	#Train	#Test	Part.	Sens.	Mod.	Symb.	Stim.	Max It.	SOA	OH	k
AMUSE Schreuder et al. (2011)	16	384	809	H	61	A	30	12	15	0.175	18.25	10
CenterSpeller Treder et al. (2011)	13	220	538	H	63	V	30	12	10	0.217	8.25	10
MVEP Schaeff et al. (2012)	15	270	606	H	57	V	30	12	10	0.266	11.7	4
P300Speller Aricò et al. (2014)	10	120	60	H	8	V	36	12	8	0.250	7.25	12
ALSP300Speller Riccio et al. (2013)	8	120	160	ALS	16	V	36	12	10	0.250	8	12
Akimpech Ledesma-Ramirez et al. (2010)	27	432	790	H	10	V	36	12	15	0.188	4	10

values was different for row and column stimuli. The other paradigms we considered are based on two-levels of selection: in this case, we considered stimuli corresponding to the outer and inner level together for computing the quartiles, since we observed similar distributions of the decision values.

6.2 No stopping scenario

As a first step, we evaluate the impact of choosing the scores by solving problem (5). We compare our strategy with both the classical **DV avg** approach and the SBF decision function (Bianchi et al. 2019) where we sum up the heuristically determined scores for all the available iterations (i.e., we use it in a no stopping fashion). We build the separating hyperplane by training a linear SVM with the package Liblinear (Fan et al. 2008). We try both the L1 and L2 loss, and since there is no clear winner, we report the results obtained with both the losses. Table 3 shows the accuracy—i.e. the percentage of correctly classified characters—obtained by the different approaches. Findings in Table 3 show that the OSBF outperforms the other two approaches since it reaches the highest accuracy on all the datasets. Please note that the OSBF is computationally cheap since the solution of problem (5) is extremely fast (order of few seconds for each MILP problem), and does not require any cross-validation phase (we fixed parameters C_1 in all the experiments). In order to further improve the accuracy, we try and build the hyperplane by solving the dual problem (33). We call this approach M-SVM. In order to solve problem (33), we apply a modification of the dual coordinate algorithm as described in the "Appendix 1". Also in this case, we do not perform any cross validation but we fix $C_2 = 0.1 \times C_1$ and we use the same value of C_1 of the previous experiment.

A statistical test (Wilcoxon Matched Pair Test, $p < 0.05$) performed on the detailed data of Table 13 (see "Appendix 2") indicated that OSBF M-SVM performed better than any other methods after Bonferroni correction, with the only exception of OSBF-L2 SVM, whose difference was statistically significant only before the aforementioned correction for multiple comparisons.

Looking at the average results in Table 4 it emerges that the results obtained by OSBF applied to the M-SVM improve on average only on some datasets, with a significant improvement on the two most difficult datasets: the one containing ALS patients and AMUSE. The intuition was that it could help only when standard SVM is not "good enough". In order to better understand the contribution of the new training problem, we look at the single-subject results, dividing the participants (across all the datasets) into two classes:

Class 1 subjects where the standard SVM problem is better than the new M-SVM;
Class 2 subjects where the standard SVM problem is worse than the new M-SVM.

We observed that Class 1 and Class 2 contain respectively about 16% and 23% of the subjects among all datasets, whereas in the remaining 61% of the subjects the standard SVM and the M-SVM perform exactly the same.

In Table 5, we report the average accuracy on both classes, and it is quite evident that the new training problem helps whenever the starting accuracy is not too high. When the starting accuracy is high, the performance does not change or gets worse probably for the overfitting. Interestingly, adding the constraints on the maximum decision value can be interpreted as a form of data augmentation. Indeed, if we include the bias b into the vector w, augmenting each data point in the training set with a last component equal to 1, we can reinterpret the constraints (22) as standard sign constraints imposed on the point z_i, with label $\hat{y}_i = 1$. Therefore, we are augmenting our training set by adding the points z_i, as shown in Fig. 4, and this results in balancing the dataset.

Table 3 Accuracy comparison between OSBF, SBF and standard approach DV-avg

Dataset	DV-avg L1-SVM	DV-avg L2-SVM	SBF L1-SVM	SBF L2-SVM	OSBF L1-SVM	OSBF L2-SVM
ALSP300Speller	0.919	0.913	0.913	0.944	0.95	**0.963**
P300Speller	0.95	**0.967**	0.95	0.95	**0.967**	**0.967**
CenterSpeller	**0.957**	0.953	0.953	0.94	0.953	0.949
AMUSE	0.741	0.756	0.752	0.753	0.763	**0.796**
MVEP	0.754	0.743	0.739	0.747	**0.777**	0.76
Akimpech	0.954	0.967	0.955	**0.978**	0.962	**0.978**

Table 4 Accuracy obtained by OSBF with the different hyperplanes: standard SVM with L1 loss, standard SVM with L2 loss, and the new M-SVM obtained by solving problem (33)

Dataset	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
ALSP300Speller	0.95	0.962	**0.975**
P300Speller	**0.967**	**0.967**	**0.967**
CenterSpeller	0.953	0.949	**0.96**
AMUSE	0.763	0.796	**0.806**
MVEP	0.777	0.76	**0.783**
Akimpech	0.962	**0.978**	0.974

6.3 Early stopping scenario

As a second step, we consider the early stopping version of both the SBF (that is the current state of the art for early stopping methods) and the OSBF. In order to evaluate the performance of the proposed method with respect to the number of iterations needed for an accurate classification also the theoretical Information transfer rate (ITR, bit/min) has been computed. The ITR is a communication measure based on Shannon channel theory with some simplifying assumptions. It can be computed by dividing the number of bits transmitted per trial (or bit rate, bits/trial) by the trial duration in minute. We compute the bit-rate, using the definition proposed in Wolpaw et al. (1998), as:

$$\mathrm{B} = \log_2 N + P \log_2 P + (1 - P) \log_2 \frac{(1 - P)}{(N - 1)}, \tag{57}$$

where N is the number of possible symbols in the speller grid and P is the probability that the target symbol is accurately classified at the end of a trial. From (57) the ITR is computed as:

$$\mathrm{ITR} = \frac{B}{\text{trial duration}} \tag{58}$$

where

$$\text{trial duration} = \frac{\mathrm{SOA} \cdot f_s \cdot \bar{i}}{60} \text{ min}. \tag{59}$$

In (59), SOA refers to the stimulus-onset asynchrony; f_s represents the number of stimuli in each stimulation sequence and $\bar{i}$ is the mean number of used iterations to select a symbol. In Tables 6, 7, 8 and 9 the results obtained with the early stopping setting are shown. Findings in Table 6 further corroborates the potentials of the OSBF since its outperforms the SBF, no matter what hyperplane is used. In Table 7 we compare the early stopping results in terms of accuracy obtained with the OSBF with the different hyperplanes (L1-SVM, L2-SVM and M-SVM): we can notice that, in this case, the M-SVM reaches a higher level of accuracy than the other methods among almost all datasets. Tables 8 and 9 show the results in terms of theoretical ITR. In this case, we can see that all the strategies reach comparable results and there is not a clear winner. This is confirmed by the statistical analysis (Wilcoxon test) which did not reveal any significant difference among the various methods when applied to the data contained on either Tables 14 and 15 included in "Appendix 2". We can then conclude that the OSBF strategy is a more conservative approach than the SBF, since it manages to keep a high level of accuracy preserving the communication speed.

Table 5 Here we divide the subjects into two sets: the ones where M-SVM performs worse than the standard L2 SVM and the ones where M-SVM performs better than the standard SVM

Dataset	% Class 1	% Class 2	Class 1 L2-SVM	Class 1 M-SVM	Class 2 L2-SVM	Class 2 M-SVM	Tot L2-SVM	Tot M-SVM
ALSP300Speller	12.5	25	0.95	0.9	0.9	0.975	0.963	0.975
CenterSpeller	7.7	46.2	1	0.977	0.902	0.928	0.95	0.96
AMUSE	25	37.5	0.865	0.842	0.695	0.738	0.796	0.806
MVEP	20	40	0.859	0.827	0.714	0.789	0.76	0.783
Akimpech	17.9	3.6	0.952	0.919	0.923	0.974	0.978	0.974
P300Speller	0	0	–	–	–	–	–	–

Columns *% Class 1* and *% Class 2* report the percentage of subjects belonging to class 1 and class 2 respectively. Columns 3-6 contain the accuracy obtained by subjects belonging to class 1 or class 2 using the L2-SVM or M-SVM hyperplane. Finally, columns *Tot L2-SVM* and *Tot M-SVM* report the average accuracy computed on the entire datasets. Results on P300 Speller dataset are not shown since in this case L2-SVM and M-SVM perform exactly the same and, so, class 1 and 2 are empty

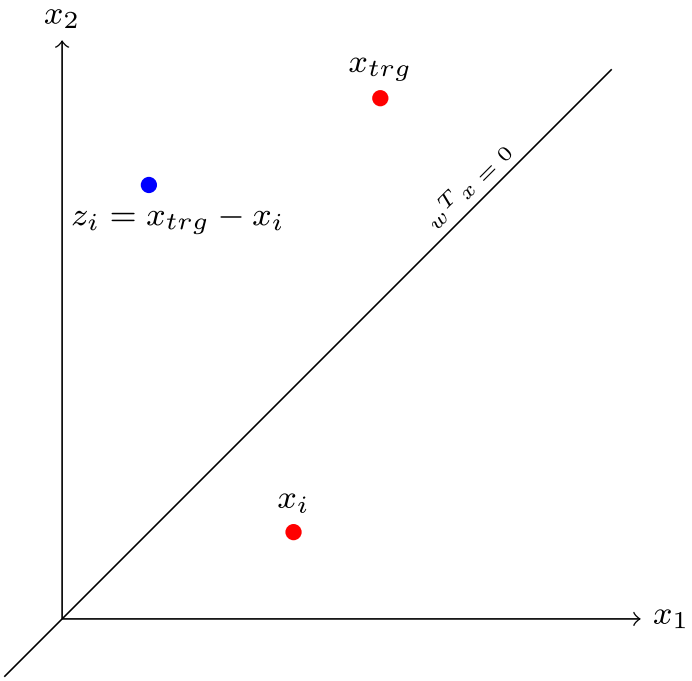

Fig. 4 Given the point x_i with label $y_i = -1$, it is possible to reinterpret constraints (22) as adding to the training set the points z_i with label $\hat{y}_i = 1$. Please note that this is a simplified example where samples are represented with 2-dimensional data points with features x_1 and x_2

Table 6 Accuracy obtained in the early stopping setting by the SBF and OSBF using the separation hyperplane given by a linear SVM with both L1 and L2 losses

Dataset	SBF L1-SVM	SBF L2-SVM	OSBF L1-SVM	OSBF L2-SVM
ALSP300Speller	0.85	0.863	**0.925**	**0.925**
P300Speller	0.95	0.95	**0.967**	0.95
CenterSpeller	0.903	0.893	**0.944**	0.931
AMUSE	0.636	0.647	**0.756**	0.744
MVEP	0.712	0.712	**0.748**	0.744
Akimpech	0.911	0.934	0.943	**0.948**

As a final analysis, in Tables 10, 11 and 12 we compare the no stopping and early stopping configurations respectively with L1-SVM, L2-SVM and M-SVM hyperplanes. It is reasonable to expect that the early stopping setting leads to a increase of ITR; on the other hand, we can expect a loss in terms of accuracy. In order to evaluate these phenomena, we report the ITR levels obtained both in the no stopping and early stopping setting and we compute both the percentage of increase in terms of ITR and the percentage of loss in terms of accuracy. We can notice that, no matter what hyperplane is used, the early stopping configuration always leads to a consistent increase in terms of ITR with a small percentage of accuracy loss.

Table 7 Accuracy obtained in the early stopping setting by the OSBF with the different hyperplanes: standard SVM with L1 loss, standard SVM with L2 loss, and the new M-SVM obtained by solving problem (33)

Dataset	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
ALSP300Speller	0.925	0.925	**0.944**
P300Speller	**0.967**	0.95	**0.967**
CenterSpeller	0.944	0.931	**0.95**
AMUSE	0.756	0.744	**0.76**
MVEP	0.748	0.744	**0.760**
Akimpech	0.943	**0.948**	0.942

Table 8 ITR (bit/min) obtained in the early stopping setting by the SBF and OSBF using the separation hyperplan given by a linear SVM with both L1 and L2 losses

Dataset	SBF L1-SVM	SBF L2-SVM	OSBF L1-SVM	OSBF L2-SVM
ALSP300Speller	20.187	19.234	20.187	**20.716**
P300Speller	**34.38**	34.086	32.343	30.356
CenterSpeller	**28.372**	28.042	27.76	27.649
AMUSE	12.67	12.929	14.490	**15.232**
MVEP	9.434	9.252	10.245	**10.626**
Akimpech	34.593	**38.464**	35.061	36.532

Table 9 ITR (bit/min) obtained in the early stopping setting by the OSBF with the different hyperplanes: standard SVM with L1 loss, standard SVM with L2 loss, and the new M-SVM obtained by solving problem (33)

Dataset	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
ALSP300Speller	20.187	20.716	**21.79**
P300Speller	**32.343**	30.356	32.11
CenterSpeller	27.76	27.649	**27.909**
AMUSE	14.490	15.232	**15.35**
MVEP	10.245	**10.626**	10.431
Akimpech	35.061	**36.532**	34.811

Table 10 Comparison of the ITR (bit/min) between the no stopping and early stopping setting for the OSBF decision function by using the standard SVM training problem with L1 loss

Dataset	OSBF L1-SVM NS	OSBF L1-SVM ES	ITR increase (%)	Accuracy loss (%)
ALSP300Speller	9.342	20.187	116.1	2.6
P300Speller	12.172	32.343	165.7	0.0
CenterSpeller	10.281	27.760	170.0	0.9
AMUSE	5.895	14.490	145.8	0.9
MVEP	5.940	10.245	72.5	3.8
Akimpech	10.077	35.061	247.9	2.0

Table 11 Comparison of the ITR (bit/min) between the no stopping and early stopping setting for the OSBF decision function by using the standard SVM training problem with L2 loss

Dataset	OSBF L2-SVM NS	OSBF L2-SVM ES	ITR increase (%)	Accuracy loss (%)
ALSP300Speller	9.452	20.716	119.2	3.9
P300Speller	12.172	30.356	149.4	1.7
CenterSpeller	10.240	27.649	170.0	2.0
AMUSE	6.276	15.232	142.7	6.5
MVEP	5.723	10.626	85.7	2.0
Akimpech	10.370	36.532	252.3	3.1

Table 12 Comparison of the ITR (bit/min) between the no stopping and early stopping setting for the OSBF decision function by using the M-SVM training problem

Dataset	OSBF M-SVM NS	OSBF M-SVM ES	ITR increase (%)	Accuracy loss (%)
ALSP300Speller	9.823	21.790	121.8	3.2
P300Speller	12.172	32.110	163.8	0.0
CenterSpeller	10.439	27.909	167.4	1.0
AMUSE	6.367	15.350	141.1	5.8
MVEP	6.002	10.431	73.8	2.9
Akimpech	10.278	34.811	238.7	3.3

7 Conclusions

BCIs are proposed for a wide range of applications, such as those for communicating (Sellers et al. 2006), for entertainment (Bianchi 2020), environmental or neuroprostheses control (Muller-Putz and Pfurtscheller 2008), rehabilitation (Bockbrader et al. 2018), and supporting diagnoses (Lugo et al. 2016), to name a few. Moreover, the same paradigm, such as the P300 discussed in this manuscript, can be used in all of them. Despite this large scenario, all of these applications try to get one among these two goals:

(i) to improve the accuracy of the systems, which usually is equivalent to minimize the occurrence of classification errors;
(ii) to maximize the communication speed or, in a more general sense, the information transfer rate.

These two strategies are usually pursued because the consequences of errors are different such as in the cases of a BCI used to drive a wheelchair, as compared to a BCI used to communicate: in the first case an error can harm a user, while in the second case it only leads to a typo. In any case, depending on the application, the strategies to achieve such goals can vary a lot.

This paper focuses on the classification problem that arises in many BCI protocols. The idea was to exploit the knowledge on the protocol in order to improve the classification accuracy and the communication speed of the BCI, that is to achieve both goals or at least to shorten the distance among them, thus allowing a painless tuning of a BCI according to the target application.

The novelty of our approach is three-fold. First, for each participant we determine the optimal scores by solving an optimization problem on her/his training data; secondly, we

implement an efficient early stopping method by solving a modified version of the optimization problem connected to training; finally, we successfully include the information on the decision function (i.e. the target always is the stimulus having maximum decision value) into the training problem. The main advantage of our proposed methodology is that the preliminary calibration phase becomes completely automatic so that a cross validation phase or manual parameters tuning is no longer fundamental.

Our method achieves such results by means of two main ingredients:

(i) the use of a MILP problem to assign a 'reliability score' to the classification of each stimulus in every iteration
(ii) the definition of a new training problem that keeps into account that the target class is assigned to the stimulus having the maximum decision value.

Both these elements have been applied in two different scenarios: a first one where accuracy was the main focus and all the iterations available for each subject were used both in the calibration and the online phase; a second one where the focus was to improve the communication speed, and hence an early stopping strategy was implemented in the online phase. In order to evaluate the approaches we conducted an extensive experimentation on datasets coming from different protocols and including both healthy subjects and ALS patients. The results show how we were able to improve accuracy and ITR on all the datasets, proving once more that combining machine learning tools to problem knowledge can significantly improve performances.

To our knowledge, according to the literature, it is the first time that a methods was successfully used and performed better than any other in either accuracy and information transfer rate. Moreover, this was verified on six different publicly available datasets, which include either healthy subjects or ALS patients. This remarkable result, which has been obtained through offline analysis, once verified also during online recording sessions, may represents the new gold standard in P300 based BCI paradigms and provide a significant improvement for all the applications that make use of it.

The use of an optimization problem to define how reliable a classification is, can be used also in the context of collaborative BCIs (Poli et al. 2014), where a group decision is made on the basis of the EEG signals of a certain number of subjects. The idea is to use an optimization problem to assign a score to each subject for each trial of a certain task as for example recognizing a face in an image containing a crowded environment (Valeriani and Poli 2019). In this case, the idea is to evaluate from the EEG the reliability of the subject on that task using an automated process based again on a the solution of a MILP problem. This is material of future work.

Funding Open Access funding provided by Universitá degli Studi di Roma Tor Vergata

Appendix

Dual Coordinate Descent Algorithm

In this section we describe how we modified the Dual Coordinate Descent Algorithm proposed in Hsieh et al. (2008) in order to find the separating hyperplane for problem 24-28. The Dual Coordinate Descent Algorithm basically solves the dual problem applying a Gauss Seidel decomposition method where each variable constitutes a block, and the subproblem with respect to a single variable is globally solved analytically. We adapt the algorithm by modifying the following points:

- how the gradient of the objective function is computed;
- how the hyperplane is updated.

In particular, we can write the objective function f and the separating hyperplane w as:

$$f = \frac{1}{2}\sum_{i=1}^{l_1}\sum_{j=1}^{l_1}\lambda_i\lambda_j y_i y_j x_i^T x_j + \sum_{i=1}^{l_1}\sum_{j=1}^{l_2}\lambda_i\rho_j y_i x_i^T z_j + \frac{1}{2}\sum_{i=1}^{l_2}\sum_{j=1}^{l_2}\rho_i\rho_j z_i z_j - \sum_{i=1}^{l_1}\lambda_i - \sum_{i=1}^{l_2}\rho_j \tag{60}$$

$$w = \sum_{i=1}^{l_1}\lambda_i y_i x_i + \sum_{i=1}^{l_2}\rho_i z_i \tag{61}$$

Let's define the vector $\alpha = \left[\lambda^T \ \rho^T\right] \in \mathbb{R}^{l_1 \times l_2}$. Equations 60 and 61 can equivalently be defined with respect to vector α. We can then express the i-th component of the gradient of $f(\alpha)$ as:

$$\nabla_i f = \begin{cases} \sum_{j=1}^{l_1}\alpha_j y_i y_j x_i^T x_j + \sum_{j=l_1}^{l_1+l_2}\alpha_j y_i x_i^T z_j - 1 & \text{if } i < l_1 \\ \sum_{j=1}^{l_1}\alpha_j y_j x_j^T z_j + \sum_{j=l_1 1}^{l_1+l_2}\alpha_j z_i^T z_j - 1 & \text{otherwise} \end{cases} \tag{62}$$

which can be rewritten as:

$$\nabla_i f = \begin{cases} y_i w^T x_i - 1 & \text{if } i < l_1 \\ w^T z_i - 1 & \text{otherwise} \end{cases} \tag{63}$$

We note that in principle we may use liblinear if we include the bias b into the vector w, augmenting each data point in the training set with a last component equal to 1, and we set $C_1 = C_2$. However, it turns out from the experiments that is more effective to treat the z_i separately defining a dedicated parameter C_2 to be chosen in cross validation, which is why we defined our own training algorithm.

Algorithm 1 A Dual Coordinate Descent Algorithm for problem 24-28

$k \leftarrow 0$
$\alpha \leftarrow [\lambda^T \ \rho^T]$
$\alpha^0 \leftarrow 0, w^0 \leftarrow 0$
while α^k not optimal **do**
  $\alpha^{k,1} \leftarrow \alpha^k, w^{k,1} \leftarrow w^k$
  for $i = 0$ to $l_1 + l_2$ **do**

$$\nabla_i f(\alpha^{k,i}) = \begin{cases} y_i (w^{k,i})^T x_i - 1 & \text{if } i < l_1; \\ (w^{k,i})^T z_i - 1 & \text{otherwise} \end{cases}$$

$$\nabla_i^P f(\alpha^{k,i}) = \begin{cases} min(\nabla_i f(\alpha^{k,i}), 0) & \text{if } \alpha^{k,i} = 0 \\ max(\nabla_i f(\alpha^{k,i}), 0) & \text{if } \alpha^{k,i} = C \\ \nabla_i f(\alpha^{k,i}) & \text{otherwise} \end{cases}$$

    if $\nabla_i^P f(\alpha^{k,i}) == 0$ **then**
      $\alpha_i^{k+1} = \alpha_i^k, w^{k+1,i} = w^{k,i}$
    else

$$\alpha_i^{k+1} = min\left(C, max\left(0, \alpha_i^k - \frac{\nabla_i f(\alpha^{k,i})}{Q_{i,i}}\right)\right)$$

$$w^{k,i+1} = \begin{cases} w^{k,i} + y_i \left(\alpha_i^{k+1} - \alpha_i^k\right) x_i & \text{if } i < l_1; \\ w_{k,i} + \left(\alpha_i^{k+1} - \alpha_i^k\right) z_i & \text{otherwise;} \end{cases}$$

$$\alpha^{k,i+1} = \left(\alpha_i^k, \ \ldots \ \alpha_i^{k+1}, \ \alpha_{i+1}^k, \ \ldots \ \alpha_{l_1+l_2}^k\right)$$

    end if
  end for
end while

Detailed numerical results

As a supplement, we provide the detailed results obtained for all subjects for all considered datasets. In Table 13 we provide the results obtained in the no stopping setting, while in Tables 14 and 15 the results for the early stopping setting are reported.

Table 13 Detail of the Accuracy results obtains with all no stopping framework mentioned

Dataset	Subj.	DV-avg L1-SVM	DV-avg L2-SVM	DV-avg M-SVM	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
ALSP300Speller	1	0.850	0.900	0.850	0.900	0.950	0.900	0.950	0.900	0.950
ALSP300Speller	2	0.800	0.850	0.850	0.850	0.850	0.850	0.850	0.900	1.000
ALSP300Speller	3	0.800	0.800	0.850	0.950	1.000	0.950	0.900	0.950	0.900
ALSP300Speller	4	0.950	0.800	0.900	0.850	0.950	0.850	0.950	0.950	0.950
ALSP300Speller	5	1.000	1.000	1.000	0.950	0.900	0.950	1.000	1.000	1.000
ALSP300Speller	6	1.000	1.000	1.000	0.900	0.950	0.950	1.000	1.000	1.000
ALSP300Speller	7	0.950	0.950	0.950	0.900	0.950	0.950	0.950	1.000	1.000
ALSP300Speller	8	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	4	0.833	0.833	0.833	0.833	0.833	0.833	0.833	0.833	0.833
P300Speller	5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	6	0.833	1.000	0.833	0.833	1.000	0.833	1.000	1.000	1.000
P300Speller	7	1.000	1.000	1.000	1.000	0.833	1.000	1.000	1.000	1.000
P300Speller	8	0.833	0.833	0.833	0.833	0.833	0.833	0.833	0.833	0.833
P300Speller	9	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	10	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
CenterSpeller	VPiac	0.929	0.952	0.929	0.905	0.905	0.905	0.952	0.929	0.929
CenterSpeller	VPiba	0.974	0.947	0.974	0.947	0.974	0.947	0.947	0.947	0.974
CenterSpeller	VPibb	1.000	1.000	1.000	0.974	1.000	0.947	1.000	1.000	1.000
CenterSpeller	VPibc	1.000	1.000	1.000	0.977	1.000	0.977	1.000	1.000	0.977
CenterSpeller	VPibd	0.976	0.927	0.951	0.976	0.951	0.976	0.927	0.927	0.951
CenterSpeller	VPibe	1.000	1.000	1.000	1.000	0.969	1.000	1.000	1.000	1.000

Table 13 continued

Dataset	Subj.	DV-avg L1-SVM	DV-avg L2-SVM	DV-avg M-SVM	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
CenterSpeller	VPibf	1.000	1.000	1.000	0.977	0.977	0.977	0.977	1.000	1.000
CenterSpeller	VPibg	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
CenterSpeller	VPibh	0.750	0.769	0.769	0.827	0.635	0.788	0.808	0.788	0.827
CenterSpeller	VPibi	0.940	0.940	0.960	0.960	0.980	0.960	0.960	0.960	0.980
CenterSpeller	VPibj	0.930	0.930	0.907	0.860	0.884	0.860	0.837	0.814	0.837
CenterSpeller	VPica	0.973	0.946	0.973	1.000	0.973	1.000	1.000	1.000	1.000
CenterSpeller	VPsaf	0.974	0.974	1.000	0.974	0.974	0.949	0.974	0.974	1.000
AMUSE	VPfar	0.711	0.644	0.711	0.600	0.556	0.578	0.711	0.711	0.711
AMUSE	VPfau	0.845	0.845	0.810	0.828	0.862	0.845	0.862	0.862	0.879
AMUSE	VPfav	0.849	0.849	0.849	0.830	0.849	0.830	0.849	0.849	0.849
AMUSE	VPfaw	0.750	0.806	0.861	0.806	0.861	0.833	0.889	0.889	0.917
AMUSE	VPfax	0.756	0.756	0.732	0.768	0.744	0.720	0.756	0.756	0.793
AMUSE	VPfaz	0.969	0.969	0.969	0.969	0.969	0.969	0.969	0.969	0.969
AMUSE	VPfca	0.974	0.947	0.974	0.974	1.000	0.974	0.947	1.000	0.974
AMUSE	VPfcb	0.695	0.712	0.780	0.847	0.746	0.746	0.712	0.847	0.831
AMUSE	VPfcc	0.969	0.969	0.938	0.938	0.969	0.938	0.969	0.969	0.969
AMUSE	VPfcd	0.743	0.857	0.743	0.800	0.857	0.857	0.800	0.914	0.886
AMUSE	VPfcg	0.682	0.591	0.621	0.712	0.727	0.697	0.697	0.697	0.697
AMUSE	VPfch	0.259	0.379	0.397	0.276	0.241	0.276	0.241	0.431	0.500
AMUSE	VPfcj	0.391	0.551	0.536	0.449	0.377	0.493	0.536	0.580	0.638
AMUSE	VPfck	0.736	0.717	0.736	0.698	0.698	0.660	0.660	0.698	0.679
AMUSE	VPfcm	0.617	0.600	0.633	0.683	0.717	0.600	0.700	0.650	0.700
AMUSE	VPkw	0.909	0.909	0.909	0.848	0.879	0.909	0.909	0.909	0.909
MVEP	VPfat	0.821	0.923	0.897	0.872	0.923	0.821	0.949	0.897	0.897
MVEP	VPgdf	0.647	0.618	0.618	0.647	0.647	0.588	0.706	0.618	0.647

Table 13 continued

Dataset	Subj.	DV-avg L1-SVM	DV-avg L2-SVM	DV-avg M-SVM	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
MVEP	VPgdg	0.821	0.744	0.821	0.692	0.769	0.718	0.769	0.744	0.795
MVEP	VPiac	0.667	0.667	0.727	0.606	0.697	0.667	0.727	0.697	0.697
MVEP	VPiba	0.519	0.500	0.519	0.537	0.537	0.593	0.556	0.500	0.648
MVEP	VPibe	1.000	0.973	1.000	0.973	0.973	0.946	1.000	1.000	1.000
MVEP	VPibs	0.760	0.780	0.760	0.820	0.660	0.740	0.840	0.840	0.820
MVEP	VPibt	0.810	0.833	0.810	0.786	0.833	0.786	0.762	0.786	0.786
MVEP	VPibu	0.468	0.553	0.489	0.426	0.468	0.447	0.532	0.489	0.489
MVEP	VPibv	0.833	0.806	0.833	0.833	0.750	0.889	0.889	0.806	0.944
MVEP	VPibw	0.975	0.925	0.975	0.975	0.975	1.000	0.925	0.975	0.950
MVEP	VPibx	0.917	0.861	0.917	0.917	0.917	0.944	0.972	0.917	0.944
MVEP	VPiby	0.684	0.632	0.684	0.737	0.711	0.684	0.737	0.763	0.711
MVEP	VPice	0.659	0.705	0.682	0.614	0.614	0.591	0.591	0.659	0.659
MVEP	VPicv	0.730	0.622	0.676	0.649	0.730	0.649	0.703	0.703	0.757
Akimpech	ACS	0.923	0.962	0.962	0.923	0.962	0.962	0.962	0.962	0.962
Akimpech	APM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	ASG	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	ASR	0.973	0.919	0.865	0.919	0.973	0.811	0.946	0.919	0.865
Akimpech	CLL	0.974	1.000	1.000	0.949	0.949	0.949	0.974	0.923	0.974
Akimpech	CLR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	DCM	0.980	0.980	0.980	1.000	0.959	0.959	1.000	1.000	0.980
Akimpech	DLP	0.957	0.957	0.913	0.913	1.000	1.000	0.957	1.000	1.000
Akimpech	DMA	0.833	0.800	0.800	0.833	0.833	0.833	0.900	0.867	0.833
Akimpech	ELC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	FSZ	0.867	0.933	0.900	0.967	1.000	0.933	0.933	0.967	0.967

Table 13 continued

Dataset	Subj.	DV-avg L1-SVM	DV-avg L2-SVM	DV-avg M-SVM	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
Akimpech	GCE	0.964	0.964	0.929	0.929	0.929	0.929	0.893	0.964	0.964
Akimpech	ICE	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	IZH	0.950	0.950	0.950	0.975	0.975	0.975	1.000	0.975	0.975
Akimpech	JCR	0.667	0.944	0.944	0.889	1.000	1.000	0.778	1.000	1.000
Akimpech	JLD	1.000	1.000	1.000	1.000	1.000	0.957	1.000	1.000	1.000
Akimpech	JMR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	JSC	0.962	0.962	0.962	0.808	0.962	0.923	0.923	1.000	0.962
Akimpech	JST	1.000	1.000	1.000	0.971	1.000	0.971	0.971	1.000	1.000
Akimpech	LAC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	LAG	0.977	0.953	0.930	0.977	0.977	0.977	0.953	0.977	0.953
Akimpech	LGP	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	LPS	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	MoMR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	PGA	0.818	0.841	0.818	0.841	0.886	0.864	0.864	0.886	0.886
Akimpech	WFG	0.907	0.953	0.953	0.930	1.000	0.977	0.977	0.977	0.977
Akimpech	XCL	1.000	1.000	1.000	0.952	1.000	1.000	0.952	1.000	1.000

Table 14 Detail of the Accuracy results obtains with all early stopping framework mentioned

Dataset	Subj.	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
ALSP300Speller	1	0.950	0.800	0.900	0.950	0.850	1.000
ALSP300Speller	2	0.750	0.750	0.700	0.800	0.900	0.850
ALSP300Speller	3	0.850	0.850	0.750	0.800	0.950	0.800
ALSP300Speller	4	0.750	0.900	0.750	0.900	0.900	0.900
ALSP300Speller	5	0.850	0.850	0.850	1.000	0.950	1.000
ALSP300Speller	6	0.800	0.850	0.850	1.000	0.900	1.000
ALSP300Speller	7	0.900	0.900	0.900	0.950	0.950	1.000
ALSP300Speller	8	0.950	1.000	1.000	1.000	1.000	1.000
P300Speller	1	0.833	0.833	0.833	1.000	1.000	1.000
P300Speller	2	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	3	1.000	1.000	1.000	1.000	0.833	1.000
P300Speller	4	0.833	0.833	0.833	0.833	0.833	0.833
P300Speller	5	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	6	0.833	1.000	0.833	1.000	1.000	1.000
P300Speller	7	1.000	0.833	1.000	1.000	1.000	1.000
P300Speller	8	1.000	1.000	1.000	0.833	0.833	0.833
P300Speller	9	1.000	1.000	1.000	1.000	1.000	1.000
P300Speller	10	1.000	1.000	1.000	1.000	1.000	1.000
CenterSpeller	VPiac	0.833	0.857	0.810	0.881	0.881	0.857
CenterSpeller	VPiba	0.921	0.921	0.921	0.947	0.947	0.974
CenterSpeller	VPibb	0.947	0.921	0.947	1.000	0.974	1.000
CenterSpeller	VPibc	0.932	0.909	0.909	1.000	1.000	0.977
CenterSpeller	VPibd	0.878	0.927	0.878	0.927	0.951	0.927
CenterSpeller	VPibe	0.969	0.938	0.969	0.969	1.000	0.969

Table 14 continued

Dataset	Subj.	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
CenterSpeller	VPibf	0.907	0.907	0.907	1.000	0.930	1.000
CenterSpeller	VPibg	1.000	1.000	1.000	1.000	1.000	1.000
CenterSpeller	VPibh	0.769	0.615	0.788	0.827	0.788	0.846
CenterSpeller	VPibi	0.900	0.940	0.960	0.960	0.940	0.960
CenterSpeller	VPibj	0.814	0.837	0.837	0.837	0.791	0.837
CenterSpeller	VPica	0.919	0.892	0.946	0.973	0.946	1.000
CenterSpeller	VPsaf	0.949	0.949	0.949	0.949	0.949	1.000
AMUSE	VPfar	0.489	0.489	0.489	0.689	0.644	0.644
AMUSE	VPfau	0.759	0.828	0.741	0.828	0.862	0.845
AMUSE	VPfav	0.717	0.830	0.811	0.830	0.830	0.811
AMUSE	VPfaw	0.528	0.667	0.694	0.806	0.833	0.861
AMUSE	VPfax	0.646	0.500	0.634	0.732	0.683	0.683
AMUSE	VPfaz	0.938	0.875	0.938	0.969	0.938	0.969
AMUSE	VPfca	0.737	0.763	0.658	0.974	0.974	0.974
AMUSE	VPfcb	0.559	0.576	0.644	0.729	0.712	0.814
AMUSE	VPfcc	0.844	0.844	0.906	0.969	0.938	0.875
AMUSE	VPfcd	0.714	0.714	0.657	0.829	0.857	0.829
AMUSE	VPfcg	0.515	0.606	0.470	0.667	0.636	0.636
AMUSE	VPfch	0.293	0.224	0.293	0.362	0.362	0.414
AMUSE	VPfcj	0.435	0.362	0.493	0.449	0.478	0.565
AMUSE	VPfck	0.547	0.528	0.528	0.642	0.679	0.679
AMUSE	VPfcm	0.583	0.667	0.417	0.683	0.600	0.617
AMUSE	VPkw	0.879	0.879	0.909	0.939	0.879	0.939
MVEP	VPfat	0.846	0.974	0.769	0.923	0.897	0.897
MVEP	VPgdf	0.529	0.588	0.441	0.588	0.618	0.618
MVEP	VPgdg	0.564	0.769	0.667	0.795	0.744	0.795

Table 14 continued

Dataset	Subj.	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
MVEP	VPiac	0.727	0.667	0.545	0.606	0.667	0.667
MVEP	VPiba	0.556	0.519	0.611	0.593	0.463	0.593
MVEP	VPibe	0.892	0.919	0.892	1.000	1.000	1.000
MVEP	VPibs	0.800	0.580	0.600	0.680	0.800	0.780
MVEP	VPibt	0.786	0.833	0.786	0.786	0.786	0.786
MVEP	VPibu	0.404	0.468	0.447	0.511	0.468	0.468
MVEP	VPibv	0.889	0.750	0.889	0.944	0.861	0.917
MVEP	VPibw	0.975	0.950	0.875	1.000	1.000	0.950
MVEP	VPibx	0.833	0.833	0.806	0.917	0.917	0.889
MVEP	VPiby	0.711	0.684	0.658	0.605	0.684	0.711
MVEP	VPice	0.545	0.523	0.523	0.591	0.614	0.659
MVEP	VPicv	0.622	0.622	0.568	0.676	0.649	0.676
Akimpech	ACS	0.808	0.923	0.846	0.846	0.846	0.885
Akimpech	APM	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	ASG	1.000	0.962	0.962	0.923	0.962	0.962
Akimpech	ASR	0.811	0.946	0.649	0.946	0.892	0.838
Akimpech	CLL	0.846	0.872	0.923	0.872	0.923	0.974
Akimpech	CLR	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	DCM	0.939	0.898	0.918	0.959	0.959	0.939
Akimpech	DLP	0.826	0.870	1.000	0.957	0.957	0.957
Akimpech	DMA	0.900	0.800	0.800	0.900	0.867	0.867
Akimpech	ELC	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	FSZ	0.967	1.000	0.900	0.933	0.967	0.967
Akimpech	GCE	0.857	0.893	0.821	0.929	0.893	0.893

Table 14 continued

Dataset	Subj.	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
Akimpech	ICE	0.917	1.000	0.875	1.000	1.000	1.000
Akimpech	IZH	0.875	0.850	0.800	0.925	0.925	0.900
Akimpech	JCR	0.889	0.944	0.833	1.000	0.944	0.889
Akimpech	JLD	1.000	1.000	0.913	1.000	1.000	1.000
Akimpech	JMR	0.923	0.923	0.923	1.000	1.000	0.962
Akimpech	JSC	0.654	0.846	0.885	0.808	0.923	0.885
Akimpech	JST	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	LAC	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	LAG	0.930	0.907	0.930	0.907	0.953	0.907
Akimpech	LGP	1.000	1.000	1.000	1.000	1.000	1.000
Akimpech	LPS	0.800	0.800	0.800	0.800	0.800	0.800
Akimpech	MoMR	1.000	0.941	0.941	1.000	0.941	1.000
Akimpech	PGA	0.818	0.864	0.841	0.841	0.886	0.886
Akimpech	WFG	0.930	0.977	0.907	0.977	0.953	0.977
Akimpech	XCL	0.905	1.000	0.952	0.952	1.000	0.952

Table 15 Detail of the ITR (bit/min) results obtains with all early stopping framework mentioned

Dataset	Subj.	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
ALSP300Speller	1	16.599	15.734	19.593	16.599	15.239	20.175
ALSP300Speller	2	12.222	12.556	11.111	12.222	13.845	13.722
ALSP300Speller	3	16.492	19.073	16.298	16.492	21.032	15.210
ALSP300Speller	4	16.424	12.501	13.826	16.424	15.879	16.836
ALSP300Speller	5	20.577	20.771	21.207	20.577	20.565	22.236
ALSP300Speller	6	21.654	19.193	18.605	21.654	20.305	22.356
ALSP300Speller	7	18.508	21.477	23.429	18.508	20.565	23.236
ALSP300Speller	8	39.018	32.566	36.928	39.018	38.296	40.548
P300Speller	1	25.132	23.148	25.132	31.020	30.263	31.815
P300Speller	2	38.774	40.025	40.025	41.359	40.025	41.359
P300Speller	3	32.652	22.977	30.263	31.815	19.547	29.542
P300Speller	4	33.831	36.650	36.650	24.433	25.132	24.433
P300Speller	5	41.359	44.314	41.359	37.599	37.599	37.599
P300Speller	6	25.132	31.020	26.655	27.573	29.542	29.542
P300Speller	7	27.573	21.454	27.573	28.855	26.974	27.573
P300Speller	8	30.263	28.855	31.020	21.990	19.991	20.456
P300Speller	9	41.359	42.786	41.359	34.466	34.466	34.466
P300Speller	10	47.722	49.631	47.722	44.314	40.025	44.314
CenterSpeller	VPiac	20.249	26.589	23.718	23.546	24.404	21.732
CenterSpeller	VPiba	26.307	26.794	26.794	27.868	26.982	30.014
CenterSpeller	VPibb	31.684	31.049	31.816	33.617	30.699	32.973
CenterSpeller	VPibc	29.426	27.713	28.289	30.473	31.336	29.083
CenterSpeller	VPibd	23.146	27.908	24.662	23.647	26.704	23.576
CenterSpeller	VPibe	36.558	33.161	37.370	32.653	33.707	32.973
CenterSpeller	VPibf	32.471	26.430	31.821	32.140	30.004	31.824
CenterSpeller	VPibg	43.509	43.085	44.608	39.607	43.725	38.569
CenterSpeller	VPibh	18.407	11.128	16.535	16.797	15.199	19.050
CenterSpeller	VPibi	28.770	31.514	31.356	27.147	27.185	26.935
CenterSpeller	VPibj	22.652	24.898	24.808	18.391	16.365	18.844
CenterSpeller	VPica	25.320	24.957	27.354	25.990	24.959	25.704
CenterSpeller	VPsaf	30.342	29.324	31.062	28.999	28.167	31.544
AMUSE	VPfar	5.549	5.370	6.552	8.610	9.986	8.314
AMUSE	VPfau	14.682	16.384	17.185	14.834	16.699	16.139
AMUSE	VPfav	15.535	14.670	15.456	19.101	21.385	19.395
AMUSE	VPfaw	8.541	12.963	12.811	14.112	16.804	16.256
AMUSE	VPfax	10.425	8.435	10.154	11.186	10.287	9.579
AMUSE	VPfaz	29.548	31.219	31.078	33.581	30.235	37.179
AMUSE	VPfca	19.655	20.983	17.787	27.462	27.767	27.462
AMUSE	VPfcb	7.303	8.242	10.630	8.356	10.413	11.195

Table 15 continued

Dataset	Subj.	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
AMUSE	VPfcc	20.129	22.135	24.316	22.509	25.576	22.971
AMUSE	VPfcd	15.004	14.126	13.374	13.763	15.183	14.970
AMUSE	VPfcg	8.621	10.283	7.204	9.691	9.228	9.166
AMUSE	VPfch	1.838	1.146	1.469	2.819	2.862	3.136
AMUSE	VPfcj	3.197	2.200	3.654	4.565	5.378	7.027
AMUSE	VPfck	9.682	9.056	9.368	9.594	10.557	10.408
AMUSE	VPfcm	5.763	7.058	4.565	9.227	7.871	8.002
AMUSE	VPkw	27.244	22.588	28.013	22.433	23.480	24.398
MVEP	VPfat	14.958	14.291	13.818	17.155	15.372	14.972
MVEP	VPgdf	5.504	5.419	3.910	5.314	5.845	6.016
MVEP	VPgdg	6.586	11.509	9.982	8.776	8.822	9.048
MVEP	VPiac	11.131	8.422	7.097	6.886	8.350	8.398
MVEP	VPiba	4.186	5.435	7.175	5.528	4.035	6.015
MVEP	VPibe	16.537	17.687	16.239	18.598	21.599	18.751
MVEP	VPibs	8.185	6.961	6.306	6.972	9.618	9.155
MVEP	VPibt	9.232	10.368	9.426	12.149	11.743	12.149
MVEP	VPibu	2.253	3.136	2.854	4.108	3.430	3.672
MVEP	VPibv	10.438	7.861	10.650	14.113	12.383	13.729
MVEP	VPibw	13.403	13.192	17.706	17.485	19.418	15.786
MVEP	VPibx	15.977	14.671	15.071	16.467	16.972	15.894
MVEP	VPiby	10.564	7.123	9.098	6.992	8.197	8.767
MVEP	VPice	5.280	4.897	5.019	5.710	6.264	6.892
MVEP	VPicv	7.271	7.813	6.522	7.422	7.343	7.215
Akimpech	ACS	23.354	29.594	27.111	22.840	23.990	24.666
Akimpech	APM	51.115	53.921	54.563	50.551	51.691	48.586
Akimpech	ASG	40.801	43.749	42.566	36.551	40.254	41.175
Akimpech	ASR	22.004	20.972	13.311	22.561	20.642	16.259
Akimpech	CLL	24.106	29.092	30.928	23.146	29.745	32.356
Akimpech	CLR	52.884	47.413	45.833	45.833	41.666	39.285
Akimpech	DCM	45.059	42.503	42.313	42.218	43.689	40.103
Akimpech	DLP	21.249	31.112	28.765	28.278	28.668	25.819
Akimpech	DMA	18.975	25.073	24.142	25.724	28.174	26.192
Akimpech	ELC	53.088	50.737	53.710	50.182	52.083	48.076
Akimpech	FSZ	34.140	45.379	29.082	32.874	35.871	34.417
Akimpech	GCE	25.441	28.303	26.500	25.405	25.961	24.513
Akimpech	ICE	35.864	38.841	28.723	35.256	34.203	33.536
Akimpech	IZH	31.358	32.417	28.353	31.212	32.067	29.545
Akimpech	JCR	17.051	21.163	22.774	23.031	23.047	19.216

Table 15 continued

Dataset	Subj.	SBF L1-SVM	SBF L2-SVM	SBF M-SVM	OSBF L1-SVM	OSBF L2-SVM	OSBF M-SVM
Akimpech	JLD	49.818	45.833	39.010	44.788	47.577	42.177
Akimpech	JMR	29.005	41.494	39.660	37.878	41.044	39.746
Akimpech	JSC	17.758	27.944	25.540	19.469	28.301	25.784
Akimpech	JST	53.710	57.053	55.220	51.691	54.347	51.115
Akimpech	LAC	58.760	57.772	60.043	55.443	50.182	52.083
Akimpech	LAG	39.163	38.809	41.354	36.548	41.729	36.667
Akimpech	LGP	41.540	50.737	51.497	46.928	51.497	50.551
Akimpech	LPS	21.166	23.951	22.754	22.754	22.754	24.598
Akimpech	MoMR	36.569	37.083	33.674	35.714	32.850	35.166
Akimpech	PGA	17.631	24.322	21.851	22.807	24.922	22.353
Akimpech	WFG	31.372	41.167	37.770	34.324	33.587	35.640
Akimpech	XCL	41.036	52.083	46.840	42.640	45.833	40.279

References

Bnci horizon website. http://bnci-horizon-2020.eu/database/data-sets

Aricò, P., Aloise, F., Schettini, F., Salinari, S., Mattia, D., & Cincotti, F. (2014). Influence of P300 latency jitter on event related potential-based brain-computer interface performance. *Journal of Neural Engineering*, *11*(3),

Bianchi, L. (2018). The npxlab suite 2018: A free features rich set of tools for the analysis of neuro-electric signals. *WSEAS Transactions on Systems and Control*, *13*(3), 145–152.

Bianchi, L. (2020). A videogame driven by the mind: Are motor acts necessary to play? In: *Advances in intelligent systems and computing, 2020, FICC*, pp. 40–50

Bianchi, L., Liti, C., & Piccialli, V. (2019). A new early stopping method for p300 spellers. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *27*(8), 1635–1643. https://doi.org/10.1109/TNSRE.2019.2924080.

Bockbrader, M. A., Francisco, G., Lee, R., Olson, J., Solinsky, R., & Boninger, M. L.: Brain computer interfaces in rehabilitation medicine. PM&R **10**(9S2), S233–S243 (2018). https://doi.org/10.1016/j.pmrj.2018.05.028.

Chaovalitwongse, W. A., Prokopyev, O. A., & Pardalos, P. M. (2006). Electroencephalogram (eeg) time series classification: Applications in epilepsy. *Annals of Operations Research*, *148*(1), 227–250. https://doi.org/10.1007/s10479-006-0076-x.

Draper, N. R., & Smith, H. (1998). *Applied regression analysis* (Vol. 326). New York: Wiley.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Farwell, L. A., & Donchin, E. (1988). Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, *70*(6), 510–523.

Friedman, J., Hastie, T., & Tibshirani, R. (2001) The elements of statistical learning, vol. 1. Springer series in statistics New York.

Gu, Z., Chen, Z., Zhang, J., Zhang, X., & Yu, Z. L. (2019). An online interactive paradigm for P300 braincomputer interface speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *27*(2), 152–161.

Höhne, J., Schreuder, M., Blankertz, B., & Tangermann, M.: Two-dimensional auditory P300 speller with predictive text system. In: *2010 Annual international conference of the IEEE engineering in medicine and biology*, pp. 4185–4188 (2010). https://doi.org/10.1109/IEMBS.2010.5627379

Hsieh, C. J., Chang, K. W., Lin, C. J., Keerthi, S. S., & Sundararajan, S. (2008) A dual coordinate descent method for large-scale linear svm. In: *Proceedings of the 25th international conference on machine learning*, pp. 408–415

Huang, Y., He, F., Xu, M., & Qi, H. (2020). Operate P300 speller when performing other task. *Journal of Neural Engineering*. http://doi.org/10.1088/1741-2552/abb4a6.

Jiang, J., Yin, E., Wang, C., Xu, M., & Ming, D. (2018). Incorporation of dynamic stopping strategy into the high-speed SSVEP-based BCIs. *Journal of Neural Engineering*, *15*(4), 046025.

Jin, J., Allison, B. Z., Sellers, E. W., Brunner, C., Horki, P., Wang, X., et al. (2011). An adaptive P300-based control system. *Journal of Neural Engineering*, *8*(3), 036006.

Kha, V. A., Nguyen, D. N., Kha, H. H., & Dutkiewicz, E. (2017). Dynamic stopping using eSVM scores analysis for event-related potential brain-computer interfaces. In: *2017 11th international symposium on medical information and communication technology (ISMICT)*, pp. 82–85.

Khojandi, A., Shylo, O., & Zokaeinikoo, M. (2019). Automatic eeg classification: A path to smart and connected sleep interventions. *Annals of Operations Research*, *276*(1), 169–190. https://doi.org/10.1007/s10479-018-2823-1.

Krusienski, D. J., Sellers, E. W., Cabestaing, F., Bayoudh, S., McFarland, D. J., Vaughan, T. M., et al. (2006). A comparison of classification techniques for the P300 speller. *Journal of Neural Engineering*, *3*(4), 299.

Krusienski, D. J., Sellers, E. W., McFarland, D. J., Vaughan, T. M., & Wolpaw, J. R. (2008). Toward enhanced p300 speller performance. *Journal of Neuroscience Methods*, *167*(1), 15–21.

Ledesma-Ramirez, C., Bojorges-Valdez, E., Yáñez-Suarez, O., Saavedra, C., Bougrain, L., & Gentiletti, G. G. (2010). An open-access p300 speller database

Lenhardt, A., Kaper, M., & Ritter, H. (2008). An adaptive P300-based online Brain'computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *16*(2), 121–130. https://doi.org/10.1109/TNSRE.2007.912816.

Liu, T., Goldberg, L., Gao, S., & Hong, B. (2010). An online brain-computer interface using non-flashing visual evoked potentials. *Journal of Neural Engineering*, *7*(3), 036003.

Lotte, F.: A tutorial on eeg signal-processing techniques for mental-state recognition in brain–computer interfaces. In: *Guide to brain-computer music interfacing*, pp. 133–161. Springer (2014)

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., et al. (2018). A review of classification algorithms for EEG-based brain-computer interfaces: A 10 year update. *Journal of Neural Engineering*, *15*(3),

Lugo, Z. R., Quitadamo, L. R., Bianchi, L., Pellas, F., Veser, S., Lesenfants, D., Real, R. G. L., Herbert, C., Guger, C., Kotchoubey, B., Mattia, D., Kbler, A., Laureys, S., & Noirhomme, Q. (2016). Cognitive processing in non-communicative patients: what can event-related potentials tell us? Frontiers in Human Neuroscience **10**, 569. https://doi.org/10.3389/fnhum.2016.00569.

Mainsah, B. O., Colwell, K. A., Collins, L. M., & Throckmorton, C. S. (2014). Utilizing a language model to improve online dynamic data collection in P300 spellers. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *22*(4), 837–846.

McCane, L.M., Heckman, S.M., McFarland, D.J., Townsend, G., Mak, J.N., Sellers, E.W., Zeitlin, D., Tenteromano, L.M., Wolpaw, J.R., Vaughan, T.M. (2015). P300-based brain-computer interface (bci) event-related potentials (erps): People with amyotrophic lateral sclerosis (als) vs. age-matched controls. Clinical Neurophysiology **126**(11), 2124–2131

Muller-Putz, G. R., & Pfurtscheller, G. (2008). Control of an Electrical Prosthesis With an SSVEP-Based BCI. *IEEE Transactions on Biomedical Engineering*, *55*(1), 361–364. https://doi.org/10.1109/TBME.2007.897815.

Piccialli, V., Sciandrone, M.: Nonlinear optimization and support vector machines. 4OR **16**(2), 111–149 (2018)

Poli, R., Valeriani, D., & Cinel, C. (2014). Collaborative brain-computer interface for aiding decision-making. *PloS One*, *9*(7), e102693.

Quitadamo, L. R., Marciani, M. G., Cardarilli, G. C., & Bianchi, L. (2008). Describing different brain computer interface systems through a unique model: A UML implementation. *Neuroinformatics*, *6*(2), 81–96.

Riccio, A., Simione, L., Schettini, F., Pizzimenti, A., Inghilleri, M., Olivetti Belardinelli, M., et al. (2013). Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis. *Frontiers in Human Neuroscience*, *7*, 732.

Schaeff, S., Treder, M. S., Venthur, B., & Blankertz, B. (2012). Exploring motion veps for gaze-independent communication. *Journal of Neural Engineering*, *9*(4), 045006.

Schreuder, M., Höhne, J., Blankertz, B., Haufe, S., Dickhaus, T., & Tangermann, M. (2013). Optimizing event-related potential based brain-computer interfaces: A systematic evaluation of dynamic stopping methods. *Journal of Neural Engineering*, *10*(3), 036025.

Schreuder, M., Rost, T., & Tangermann, M. (2011). Listen, you are writing! speeding up online spelling with a dynamic auditory BCI. *Frontiers in Neuroscience*, *5*, 112.

Sellers, E. W., Krusienski, D. J., McFarland, D. J., Vaughan, T. M., & Wolpaw, J. R. (2006). A P300 event-related potential brain-computer interface (BCI): The effects of matrix size and inter stimulus interval on performance. *Biological Psychology*, *73*(3), 242–252.

Sellers, E. W., Kubler, A., & Donchin, E. (2006). Brain-computer interface research at the university of south florida cognitive psychophysiology laboratory: The P300 Speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(2), 221–224.

Shahriari, Y., Vaughan, T. M., McCane, L., Allison, B. Z., Wolpaw, J. R., & Krusienski, D. J. (2019). An exploration of BCI performance variations in people with amyotrophic lateral sclerosis using longitudinal EEG data. *Journal of Neural Engineering*, *16*(5), 056031.

Sur, S., & Sinha, V. (2009). Event-related potential: An overview. *Industrial Psychiatry Journal*, *18*(1), 70.

Throckmorton, C. S., Colwell, K. A., Ryan, D. B., Sellers, E. W., & Collins, L. M. (2013). Bayesian approach to dynamically controlling data collection in P300 spellers. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *21*(3), 508–517.

Treder, M. S., & Blankertz, B. (2010). (c) overt attention and visual speller design in an erp-based brain-computer interface. *Behavioral and Brain Functions*, *6*(1), 28.

Treder, M. S., Schmidt, N. M., & Blankertz, B. (2011). Gaze-independent brain-computer interfaces based on covert attention and feature attention. *Journal of Neural Engineering*, *8*(6), 066003.

Valeriani, D., & Poli, R. (2019). Cyborg groups enhance face recognition in crowded environments. *PloS One*, *14*(3), e0212935.

Vo, K., Nguyen, D. N., Kha, H. H., & Dutkiewicz, E. (2017). Subject-independent P300 BCI using ensemble classifier, dynamic stopping and adaptive learning. In: *GLOBECOM 2017-2017 IEEE global communications conference*, pp. 1–7

Vo, K., Pham, T., Nguyen, D. N., Kha, H. H., & Dutkiewicz, E. (2018). Subject-independent ERP-based brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(4), 719–728. https://doi.org/10.1109/tnsre.2018.2810332.

Wolpaw, J., & Wolpaw, E. W. (2012). *Brain-computer interfaces: Principles and practice*. New York: Oxford University Press.

Wolpaw, J. R., Ramoser, H., McFarland, D. J., & Pfurtscheller, G. (1998). EEG-based communication: Improved accuracy by response verification. *IEEE Transactions on Rehabilitation Engineering*, *6*(3), 326–333.

Zhang, H., Guan, C., & Wang, C. (2008). Asynchronous P300-based brain-computer interfaces: A computational approach with statistical models. *IEEE Transactions on Biomedical Engineering*, *55*(6), 1754–1763.

Affiliations

Luigi Bianchi[1] · Chiara Liti[1] · Giampaolo Liuzzi[2] · Veronica Piccialli[1] · Cecilia Salvatore[1]

✉ Veronica Piccialli
veronica.piccialli@uniroma2.it

Luigi Bianchi
luigi.bianchi@uniroma2.it

Chiara Liti
chiara.liti@uniroma2.it

Giampaolo Liuzzi
giampaolo.liuzzi@diag.uniroma1.it

Cecilia Salvatore
cecilia.salvatore@uniroma2.it

[1] Department of Civil Engineering and Computer Science, University of Rome Tor Vergata, Rome, Italy

[2] Department of Computer, Control and Management Engineering, Sapienza University of Rome, Rome, Italy