Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-018-0840-1

ORIGINAL RESEARCH

Weighted extreme learning machine for P300 detection with application to brain computer interface

Wanzeng Kong[1,2] · Shijie Guo[1] · Yanfang Long[1] · Yong Peng[1] · Hong Zeng[1] · Xinyu Zhang[1] · Jianhai Zhang[1]

Received: 31 October 2017 / Accepted: 3 May 2018

Abstract
Brain–computer interface (BCI) is a communication system, which brain signals can be analyzed and transformed into commands of external devices. The P300-speller BCI utilizes the P300 evoked potentials, which generated in oddball paradigms, to spell characters. In order to eliminate the imbalance existed between target and non-target stimuli data in electroencephalogram (EEG) signals, we apply weighted extreme learning machine (WELM), which consequently improves the P300 detection accuracy. For starter, the raw EEG signals of all channels are divided into several segments by a given time window. Afterwards, features of each segment are extracted by principal component analysis. Then, WELM executes a classification for above features. Last, outputs of WELM classifier are used to recognize the target character in the P300 speller. Experiments were carried out based on the third BCI Competition Data Set II and achieved an average accuracy of 97%, and it still reached 85% with the first eight high sensitivity channels for P300 responses.

Keywords Brain–computer interface · P300 speller · Principal component analysis · Weighted extreme learning machine

1 Introduction

Brain–computer interface (BCI) is a communication system designed to allow users to interact directly with external environment, which is not dependent on brain's normal output pathways of peripheral nerves and muscles (Vaughan and Wolpaw 2006). A BCI is usually decomposed into four main parts. First, signals are acquired by a signal amplifier. Second, features of different tasks are extracted from raw electroencephalogram (EEG) signals. Third, different classification methods are used to classify above features. Last, the classification results are converted into commands of external devices.

Generally, we focus more on the non-invasive BCI, which is convenient for EEG signals acquisition. EEG is defined as the recording of electrical activity in the brain, which can be collected by electrodes on the scalp's surface. Our interest in the BCI framework is based on P300 event-related potential (ERP), which is the natural response of the brain to certain specific external stimuli. The BCI system based on P300 not only has a simple experimental paradigm, but also can induce P300 potential without any training. P300 waves usually occur around 300 ms after the infrequent stimulus appears. In a oddball paradigm (Qassim et al. 2013), P300 waves are typically used to examine target tasks. The P300 speller is one of the well-known BCIs based on the P300 potential, proposed by Farwell and Donchin (1988).

The task of target characters detection in the P300 speller is usually regarded as a binary classification problem whether the EEG signals contain P300 waves. Detecting P300 waves is the key to character recognition in the P300

✉ Wanzeng Kong
kongwanzeng@hdu.edu.cn

Shijie Guo
shijieguo@163.com

Yanfang Long
yanfang_longsun@163.com

Yong Peng
yongpeng@hdu.edu.cn

Hong Zeng
jivon@hdu.edu.cn

Xinyu Zhang
dianazhang26@163.com

Jianhai Zhang
jhzhang@hdu.edu.cn

[1] College of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China

[2] Fujian Key Laboratory of Rehabilitation Technology, Fuzhou, China

Published online: 16 May 2018

speller paradigm. Therefore, the accuracy of P300 waves detection will ensure a better character recognition rate in the classification step. Pattern recognition techniques are also used for the classification and detection of P300 waves. Support vector machine (SVM) (Liu et al. 2010), fisher linear discriminant analysis (FLDA) (Mirghasemi et al. 2006), stepwise linear discriminant analysis (S-WLDA) (Krusienski et al. 2008), Bayesian linear discriminant analysis (BLD-A) (Hoffmann et al. 2008) and neural networks (Zhang et al. 2007a) have already been used for BCI and P300 potential detection. These classification methods have proved that neural networks can be used for classifying EEG signals and tailoring a BCI, except some linear classifiers.

As the back propagation (BP) neural networks require to adjust learning parameters iteratively, extreme learning machine (ELM) (Huang et al. 2006a, b, 2004; Peng and Lu 2017) has been attracting more and more researchers. Because there is no need to fit any parameters. ELM is a single feedforward neural network (SLFN) by randomly generating input weights and hidden bias. Therefore, ELM has higher scalability and less computational complexity than BP neural networks. Currently, ELM has been used in many fields such as image recognition, text classification and signal processing (Huang et al. 2012; Shi and Lu 2013; Zhang et al. 2007b). However, the performance of the ELM classifier is not very meaningful in the P300 speller. The foremost reason is that P300 waves are generated by the oddball paradigm. As a result, there is an imbalance between the P300 component and the non-P300 component in EEG signals data set. In addition, to aggregate effective data features and reduce data noise, we propose a method that combines principal component analysis (PCA) and weighted extreme machine learning (WELM) (Zong et al. 2013), to detect P300 responses and recognize target characters in the P300 speller.

The remainder of this paper is organized as follows. In Sect. 2, data set and the P300 speller are introduced. In Sect. 3, a method of using WELM classifier for data analysis and a way of channel selection are proposed. In Sect. 4, experiments are conducted. In Sect. 5, some conclusions are drawn.

2 Background

2.1 P300 speller

Farwell and Donchin described a visual paradigm that satisfies the oddball paradigm based on P300 ERPs, called the P300 speller. As shown in Fig. 1, they used a 6 × 6 character matrix. Each cell of the matrix contains a character: [A−Z], [1−9] or [_]. In order to spell a target character, this matrix is shown to the subjects on the computer screen by randomly intensifying the six rows and six columns at a rate of 5.7 Hz. First, the subjects focus on a target character which they want to spell. Then, an evoked response similar to the P300 response occurs naturally (when the intensified row or column contains the desired character). In the P300 speller, the main objective is to detect P300 responses in EEG signals accurately and instantly, which can ensure a high information transfer rate between users and machines.

According to the above introduction, the process of the P300 speller can be divided roughly into two parts. One part is to detect P300 responses, and the other part is to identify characters. For the former, the results of the classification problem correspond to the detection of P300 responses indicating that there are P300 or non-P300 waves (Fig. 2a). For the latter, outputs of the classification algorithm are just P300 or non-P300 responses. A character is determined by two P300 responses, which are generated by the row and

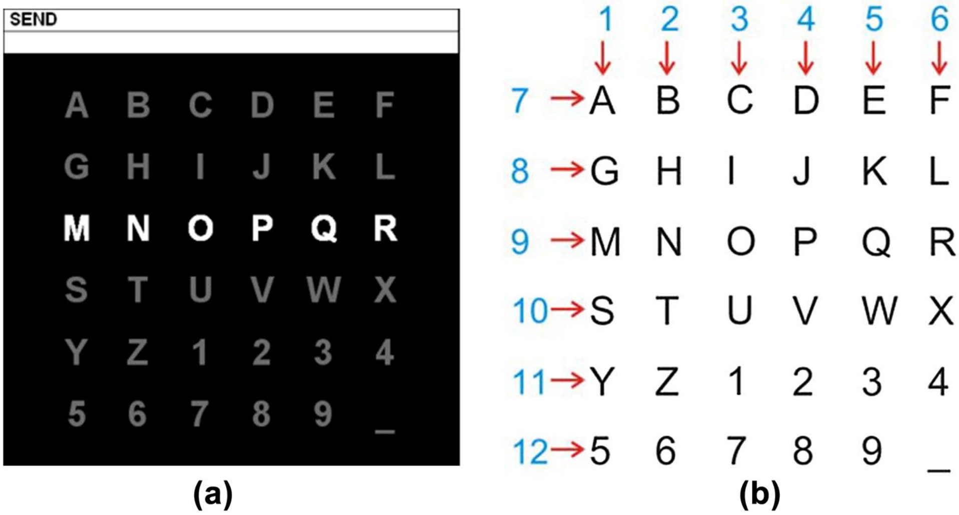

Fig. 1 **a** The P300 speller paradigm with one row highlighted. **b** The sets of rows and columns in the P300 speller paradigm

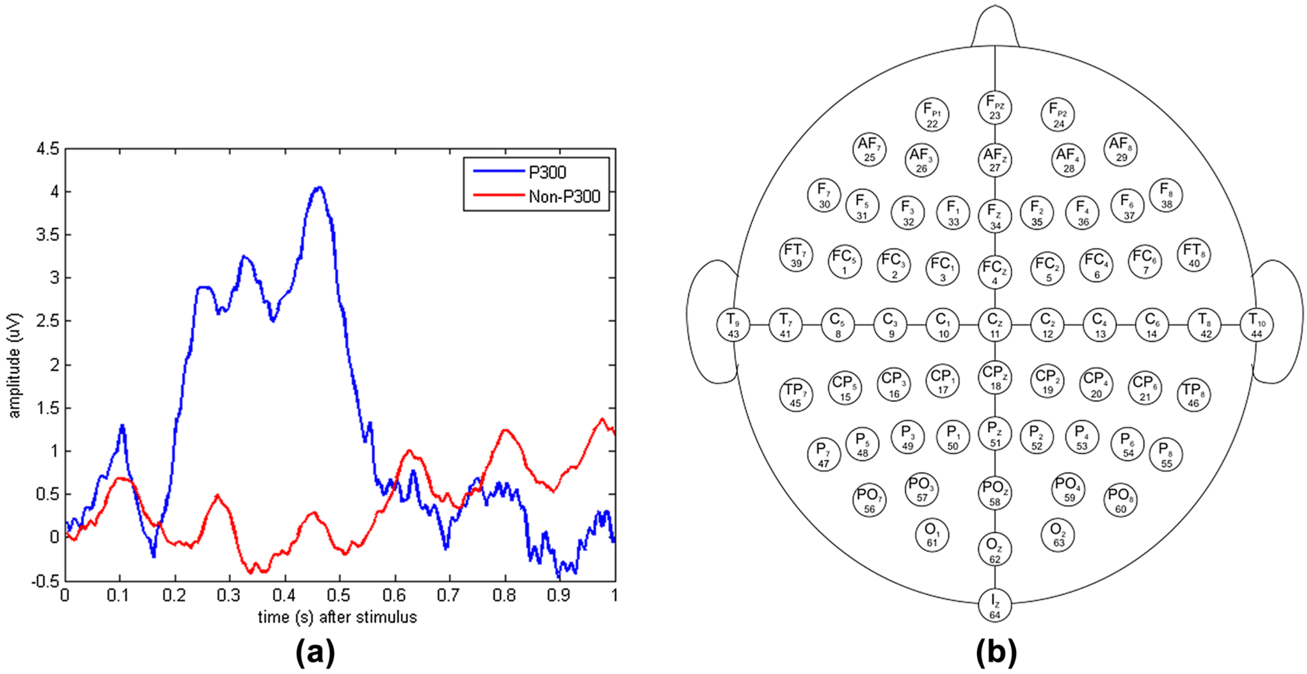

Fig. 2 **a** Comparison of two waves with P300 and non-P300. **b** 64 channels of electrode map

column of the target character. Although a P300 response can appear at a particular moment, no one can be able to produce a clear P300 wave so far. Thus, in order to improve the accuracy of character recognition, all of the rows and columns would be intensified 15 times.

2.2 Data set

We used Data Set II in our experiment. It can be obtained from the BCI Competition III (Blankertz et al. 2006). This data set was originally provided by the Wadsworth Center, New York State Department of Health. This database contains P300 evoked potentials of two subjects. EEG signals (bandpass filtered between 0.1 and 60 Hz and digitized at 240 Hz) were recorded for spelling sessions with the BCI2000 framework (Schalk et al. 2004). As shown in Fig. 2b, there are 64 electrodes placed on different points of human scalp for recording EEG signals. In this experiment, the matrix was first displayed for a 2.5 s period, and during this time each character had the same intensity (i.e., the matrix was blank). Then, each row or column of the matrix was randomly intensified for 100 ms. Last, the matrix was blank for 75 ms. The intensification of rows and columns were randomized in sets of 12. We define a target character recognition as a character epoch. The sets of 12 intensification were repeated 15 times for each character epoch. Therefore, there were $2 \times 15 = 30$ P300 responses in the process of each character recognition.

Each subject has a training set and test set. Training set is composed of 85 character, while the test set contains 100 characters. Each character epoch contains two P300 responses. One is to intensify a row and the other is to a column. Therefore, the number of P300 waves is $2 \times 15 \times 85 = 2550$ in training set, while in test set the number is $2 \times 15 \times 100 = 3000$. The numbers of non-P300 waves are respectively $10 \times 15 \times 85 = 12750$ in training set and $10 \times 15 \times 100 = 15000$ in the test.

3 Method

As shown in Fig. 3, the data processing of the P300 speller contains two parts. The one is that we need to train a classifier by using 85 characters of each subject. The other is that the test data will be used as input to the classifier to predict the values. The process begins with the preprocessing and feature extraction operations of the original EEG signals in the training set and test set. Since the accuracy of P300 classification results is related to the trained classifier, we use the training set to optimize the classifier parameters. Finally, we test the performance of the classifier based on the recognition rate of the test set.

3.1 Preprocessing

Before the feature extraction and establishment of training process, several preprocessing operations were applied to the data set.

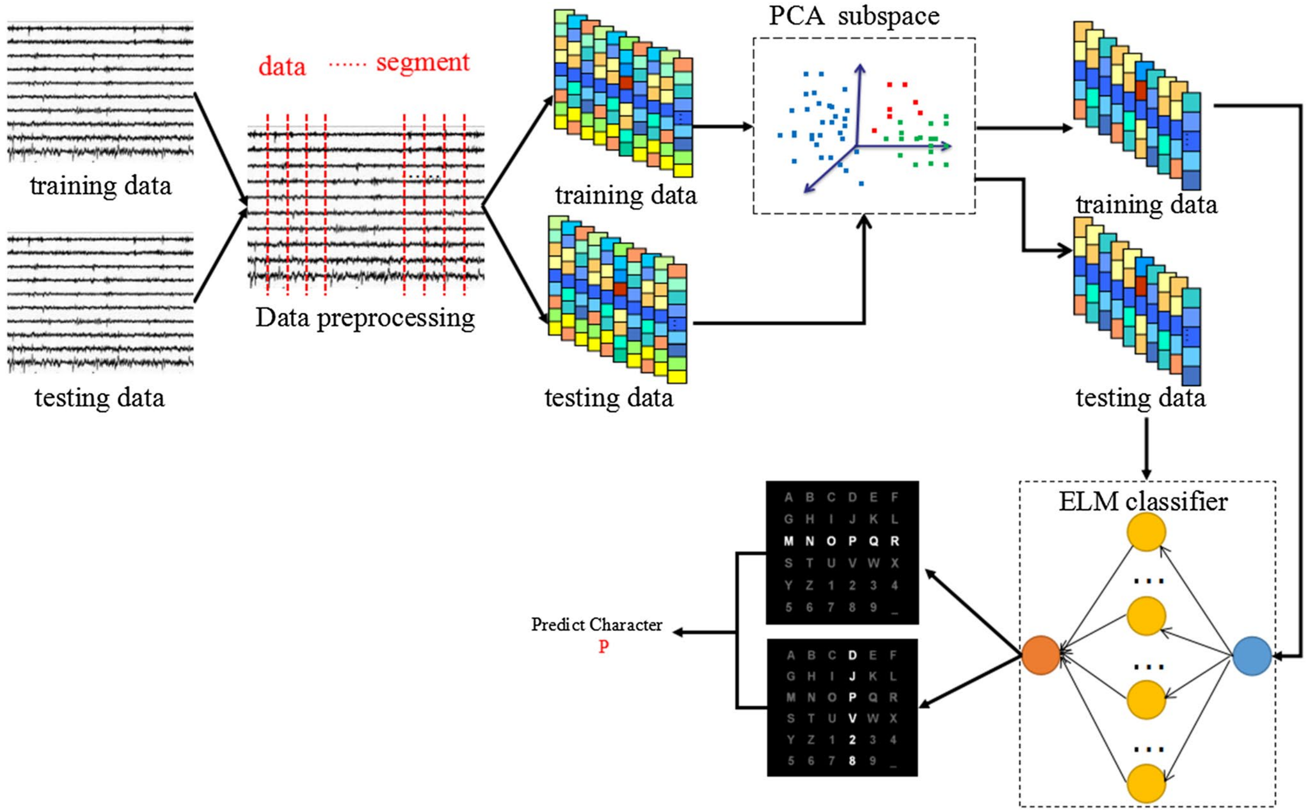

Fig. 3 Data processing of the P300 speller

1. *Filtering* The EEG data were filtered between 1 and 10 Hz with a six-order forward backward Butterworth filter.
2. *Down sample* The sampling frequency is reduced from 240 to 20 Hz by selecting each 12th sample from band-pass-filtered data.
3. *Single-trial extraction* Since the stimuli interval is 175 ms, we extract 700 ms filtered data from the beginning of a row or column intensified.
4. *Artifact rejection* Eye blinks, eye movement, muscle activity, or subject movement can cause large amplitude outliers in EEG database. In order to reduce the effects of such outliers, firstly, we sorted all samples of each channel. Secondly, the 10th percentile and the 90th percentile in EEG data of each channel were computed. Amplitude values are below the 10th percentile or above the 90th percentile, respectively (Hoffmann et al. 2008).
5. *Normalization* The samples of each channel were scaled to the interval [−1 1] by the z-score method. It is performed by (1) for each sample:

$$\mathbf{X}_{normalized} = \frac{\mathbf{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}}, \tag{1}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are mean and standard deviation value of EEG signals in the input sample of each channel.

3.2 Channel selection

After data preprocessing, the number of temporal samples is denoted by N_s. As the sampling rate is 20Hz and the trial duration is 700 ms, N_s is 14. Event-related evoked potentials appear almost 300–400 ms after the stimulus begins. Therefore, we focus on the 6th, 7th and 8th samples of each channel, and whether there are obvious P300 waves. We measure the sensitivity of ith electrode to P300 responses by:

$$\omega_i = \sum_j \left|P_i^j - NP_i^j\right|, i = 1, 2, \dots, N_{elect}, \quad j \in (6, 7, 8), \tag{2}$$

where N_{elect} denotes the number of electrodes, $1 \le i \le N_{elect}$. P_i^j represents the accumulation of the ith electrode contains P300 waves in the j samples. NP_i^j means the accumulation of the ith electrode contains non-P300 waves in the j samples.

3.3 PCA feature extraction

Now, the size of a feature vector is $N_s \times N_{elect}$ (the max value of N_{elect} is 64). Total number of feature vectors in training set is 15300 (12 trials × 15 sequences × 85 characters). We divide the EEG signals into two categories. P300 wave is in class + 1, while non-P300 wave is in class − 1. Out of these 15300 signals, 12750 signals belong to class − 1 and the rest 2550 signals belong to class + 1. Besides, the number of feature vectors in test set is 18000 (12 trials × 15 sequences × 100 characters).

In order to extract more effective features and decrease computation time, we use principal component analysis (PCA) method (Selim et al. 2014) to reduce dimensions. PCA is a linear transformation method that the original data can be transformed into an orthogonal space. PCA re-references multi-dimensional data to a new orthogonal basis such that there is no inter-channel covariance. The goal of PCA is using orthogonal basis to re-describe the data space and find effective principal components (PCs). The new basis should try to reveal the relationship in the original data. We define a sample set as $\mathbf{X}$, which comprises N samples, and the dimension of each sample is d. In this paper, the N of training set is 15300, the N of test set is 18000 and the d is $N_s \times N_{elect}$. The steps to implement the PCA algorithm are presented.

1. Sample centralization. The goal is to ensure that the mean value of each dimension is zero:

$$\begin{aligned}\mathbf{x}_i =& \mathbf{x}_i - \bar{\mathbf{x}}, \mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \\ \mathbf{x}_i =& [\mathbf{x}_{i1}, \dots, \mathbf{x}_{id}]^T, \quad i = 1, 2, \dots, N,\end{aligned} \tag{3}$$

   where $\bar{\mathbf{x}}$ is a mean vector of all rows of the matrix $\mathbf{X}$.
2. Computing covariance matrix:

$$\mathbf{C} = \frac{1}{N-1}\mathbf{X}\mathbf{X}^T, \quad \mathbf{C} \in R^{d \times d}. \tag{4}$$

3. Eigenvalue decomposition of covariance matrix $\mathbf{C}$:

$$\mathbf{P}^T\mathbf{C}\mathbf{P} = \mathbf{\Lambda}, \mathbf{P} \in R^{d \times d} \text{ and } \mathbf{\Lambda} \in R^{d \times d}. \tag{5}$$

   where $\mathbf{P}$ is an orthogonal matrix that consists of eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix that consists of its eigenvalues. The projection matrix is constructed by selecting the eigenvectors corresponding to the largest $k(k < d)$ eigenvalues. So the new feature matrix $\mathbf{X}_{new}$ can be calculated by:

$$\mathbf{X}_{new} = \mathbf{P}^T\mathbf{X}, \mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_k]. \tag{6}$$

In this paper, the test set $\mathbf{X}$ is projected onto an orthogonal space generated by the training set. $\mathbf{X}_{new} \in R^{k \times N}$ is a input matrix of the classifier. We will compare the performance of different ELM classifiers. Each classifier uses the input data after the same preprocessing and feature extraction operation to train.

3.4 Classification based on WELM

In this subsection, to make further improvements on classification accuracy, we use a classifier named weighted extreme learning machine. Weighted-ELM is an extension of ELM and constrained-optimization-based ELM algorithm.

Extreme learning machine For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i)$ where $\mathbf{x}_i = [x_{i1}, ..., x_{id}]^T \in R^d$ and $\mathbf{t}_i = [t_{i1}, ..., t_{im}]^T \in R^m$. ELM is a SLFN which has L hidden neurons and randomly generates input weights and hidden bias $(\mathbf{a}_i, b_i), i = 1, 2, ..., L$. Training the ELM neural network aims at learning the map between the input and output patterns. It is mathematically to solve the following equation:

$$\mathbf{f}(\mathbf{x_j}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{t_j}, \quad j = 1, 2, \dots, N, \tag{7}$$

where L denotes the number of hidden-layer nodes, $\boldsymbol{\beta}_i = [\beta_{i1}, ...\beta_{im}]^T$ represents the weight of connecting the ith hidden node and output; $\mathbf{h}(\mathbf{x_j}) = [G(\mathbf{a}_1, b_1, \mathbf{x}_j), ..., G(\mathbf{a}_L, b_L, \mathbf{x}_j)]$, notation $G(\mathbf{a}_i, b_i, \mathbf{x}_j)$ is the activation function of the hidden node; the activation function can be sigmoid, sine, hardlim, tribas or radial function. Once outputs of the ith hidden units are fixed, and it can be solved by minimum square error estimation. The problem becomes a set of linear equations. So the Eq. (7) can be written in matrix form as: $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$

$$\mathbf{H} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \dots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \dots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}. \tag{8}$$

Hence, the hidden layer output matrix $\mathbf{H} = [\mathbf{h}(\mathbf{x_1}), ..., \mathbf{h}(\mathbf{x_N})]^T$ is also called ELM feature space and $\mathbf{T}$ is a target matrix of output layer. In order to solve the $\boldsymbol{\beta}$ matrix, we use the least-square solution to estimation $\boldsymbol{\beta} = \mathbf{H}^{+}\mathbf{T}$ (Chen et al. 2015), where $\mathbf{H}^{+}$ is the moore-penrose inverse of $\mathbf{H}$. So the decision function is $\mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \cdot \boldsymbol{\beta}$.

Weighted extreme learning machine For optimizing the linear system $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$, the constrained-optimization-based ELM (CELM) is used to make minimize training error $\|\xi\|^2 = \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2$ and output weight two-norm of $\boldsymbol{\beta}$. Therefore, the calculation process of CELM can be expressed as formula (8) for binary classification problem (one output node):

$$\begin{aligned} &\min \mathcal{L}_{P(\beta,\xi)} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ &s.\,t.\ \mathbf{h}(\mathbf{x_i})\boldsymbol{\beta} = t_i - \xi_i, \quad i = 1, 2, \dots, N. \end{aligned} \tag{9}$$

We introduced a weight matrix to balance P300 and none-P300 data based on the constrained-optimization-based

ELM, which is called weighted-ELM (WELM). There is a set of training data $(\mathbf{x}_i, t_i), i = 1, 2, \ldots, N$, where t_i is either $+\,1$ or $-\,1$. We define an $N \times N$ diagonal matrix $\mathbf{W}$. The value of the matrix $\mathbf{W}$ is associated with the positive and negative proportions in the data samples. And we define the number of positive and negative samples with N_p and N_n respectively,

$$\mathbf{W} = \begin{bmatrix} \frac{1}{N_p} & & & & 0 \\ & \frac{1}{N_p} & 0 & & \\ & \ddots & & & \\ & 0 & \frac{1}{N_n} & & \\ 0 & & & \frac{1}{N_n} & \end{bmatrix}_{N \times N}, \tag{10}$$

where $N = N_n + N_p$. Then, we have an optimization problem mathematically written as

$$\begin{aligned} &\min \mathcal{L}_{P(\boldsymbol{\beta},\xi)} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\mathbf{W}\sum_{i=1}^{N}\xi_i^2 \\ &s.t.\ h(x_i)\boldsymbol{\beta} = t_i - \xi_i, i = 1, 2, \ldots, N \end{aligned} \tag{11}$$

According to the Karush–Kuhn–Tucker (KKT) theorem (Fletcher 1981), the dual optimization problem for (11) is

$$\mathcal{L}_{D(\boldsymbol{\beta},\xi)} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\mathbf{W}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i(\mathbf{h}(\mathbf{x_i})\boldsymbol{\beta} - t_i + \xi_i) \tag{12}$$

where C is a regularization factor, which is fitting function of smoothness. Each Lagrange multiplier α_i corresponds to the ith training sample, and ξ_i is the training error vector of one output node with respect to the training sample $\mathbf{x}_i$. We can formulate partial derivative for different parameters of $\mathcal{L}_{D(\boldsymbol{\beta},\xi)}$.

Therefore, a binary classification problem has one output node. The decision function of weighted ELM-(WELM) is:

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{W}\mathbf{H}\mathbf{H}^T\right)\mathbf{W}\mathbf{T}, \tag{13}$$

where N is small or

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}\mathbf{W}\mathbf{T}, \tag{14}$$

where N is large. According to the definition of ELM kernel, we have

$$\Omega_{ELM} = \mathbf{H}\mathbf{H}^T : \Omega_{ELM(i,j)} = \mathbf{h}(\mathbf{x_i}) \cdot \mathbf{h}(\mathbf{x_j}) = K(\mathbf{x}_i, \mathbf{x}_j). \tag{15}$$

So the output function of kernel WELM is

$$\begin{aligned} f(\mathbf{x}) &= h(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \\ &= \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_i) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}\left(\frac{\mathbf{I}}{C} + \Omega_{ELM}\right). \end{aligned} \tag{16}$$

The matrix $\mathbf{H}$ is a feature map of WELM from the input layer to the hidden layer. However, it is not necessary that the feature mapping function $\mathbf{h}(\mathbf{x})$ is always known. According to traditional kernel with Mercers conditions defined, the kernel functions can include but are not limited to Linear kernel: $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y} + C$, C is a constant; Gaussian kernel: $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma||\mathbf{x} - \mathbf{y}||)$, γ is the kernel parameter; Polynomial Kernel: $K(\mathbf{x}, \mathbf{y}) = (a\mathbf{x}^T\mathbf{y} + C)^d$, C and a are constants, d is the power exponent.

3.5 Character recognition

The outputs of classifier in test set can be directly applied to character recognition. There are 15 sequences numbered from 1 to 15. These numbers indicate that the total times of a row or column has been repeatedly intensified. And each sequence consists of 12 values output by the classifier. In general, the P300 wave has a positive value and the non-P300 wave has a negative value. However, it is possible that the values of the classifier are misconceived. So we do not use the positive and negative values to predict the characters directly. In order to accurately detect the target character that the subject is focusing on, we cumulate the values of P300 detection for each of the 12 rows and columns. The first six values represent the six columns, and the last six values denote the six rows. The row and column of the predicted character are defined by:

$$\begin{aligned} r &= argmax(\Sigma_{i=1}^{n} f(V(i,j))), 1 \le j \le 6, \\ c &= argmax(\Sigma_{i=1}^{n} f(V(i,j))), 7 \le j \le 12, \end{aligned} \tag{17}$$

where $V(i, j)$ is the pattern of the sequence corresponding to the subject responding to the row or column intensified j. The row and column of the predicted character in the matrix are r and c, respectively. Besides, n is the number of sequences.

4 Experiment results

4.1 Channel selection results

According to ω_i (i.e, the sensitivity of the ith electrode to P300 responses of each subject) we have drawn the brain topographic map, as shown in Fig. 4.

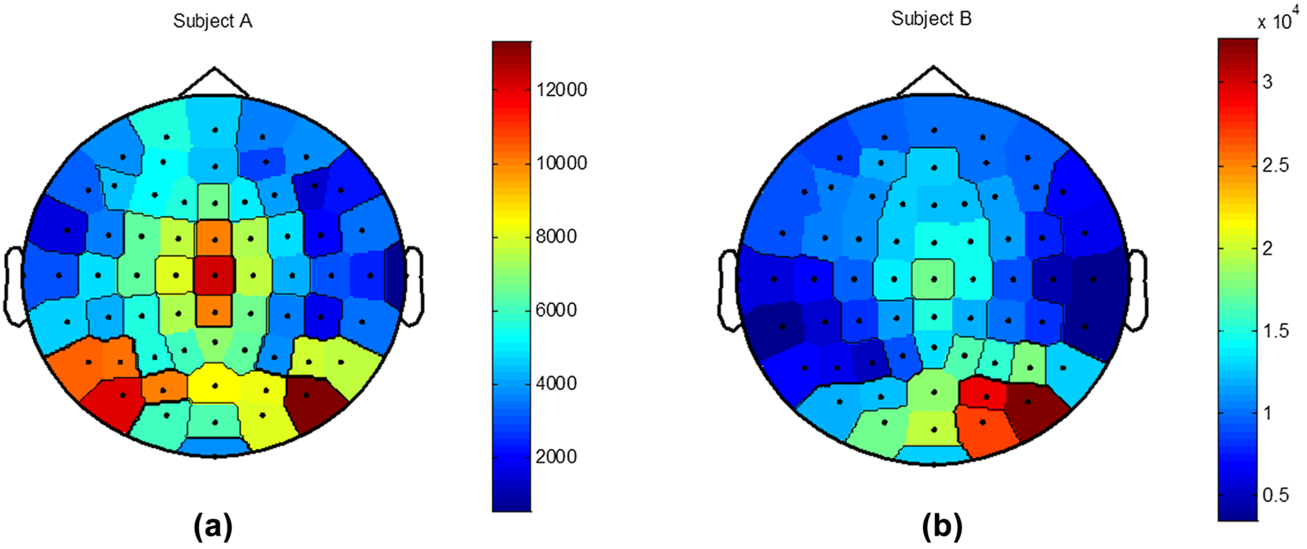

Fig. 4 **a** The sensitivity of all electrodes of subject A. **b** The sensitivity of all electrodes of subject B

Table 1 Eight preferred electrodes of subject A and B

Subject	Number							
	1	2	4	5	8	10	13	15
A	FC_z	C_z	CP_z	P_7	P_5	PO_7	PO_3	PO_8
B	C_z	P_6	PO_z	PO_4	PO_8	O_1	O_z	O_2

Next, we obtain the first eight values after sorting ω_i in descending order, and acquire these corresponding channels. (i.e., the first eight electrodes with better sensitivity to P300 responses were selected as the preferred channels.) See Table 1.

It is obvious that the eight electrodes selected in Table 1 are completely consistent with Fig. 4. However, only the PO_8 and C_z electrodes are the same. The primary reason is that the amplitude of P300 and non-P300 waves of each channel of subject A and B are not exactly the same.

4.2 Performance of different classifiers

The main task of the P300 speller is to improve the recognition rate. Therefore, we use three different ELM classifiers to detect P300 waves. According to the size of data sets and characteristics of the algorithm, we decide to use 700 hidden nodes and 10 times average calculation for ELM and CELM classifiers. And the WELM classifier uses a linear kernel as the decision function. The performance of each classifier is evaluated by the recognition accuracy of test characters with all channels (N_{elect} is 64).

The results of testing accuracy with different number of sequences of subject A and B are shown in Fig. 5. In this figure, the difference in accuracy between ELM and CELM classifiers is not particularly obvious for subject A and B. The WELM classifier can obtain a better result than both ELM and CELM. Testing accuracy of WELM classifier for subject A is better than B from the 1st to 12th sequence. In the 15th sequence, subject A can recognize 98 characters, while subject B recognizes fewer characters compared with A. The reason of this phenomenon may be that the subject B's attention is more focused than A at the beginning of the flash. With the increasing number of flashes, subject B's attention has begun to decrease.

4.3 Results on the set of BCI III competition

When the researchers used competition data set, the final result was based on the average value of the character recognition rate of subject A and B. The best result is obtained with WELM classifier with a recognition rate of 97% in the 15th sequence as shown in Fig. 6. The recognition rates of the ELM and CELM classifiers are 86.3 and 86.6% in the 15th sequence, respectively.

The final result of the experiment shows that our proposed method can indeed improve character recognition. In practical applications, fewer electrodes will be more convenient. According to our channel selection and classification methods, we also achieved an accuracy of 85.0% in the test characters. See Table 2.

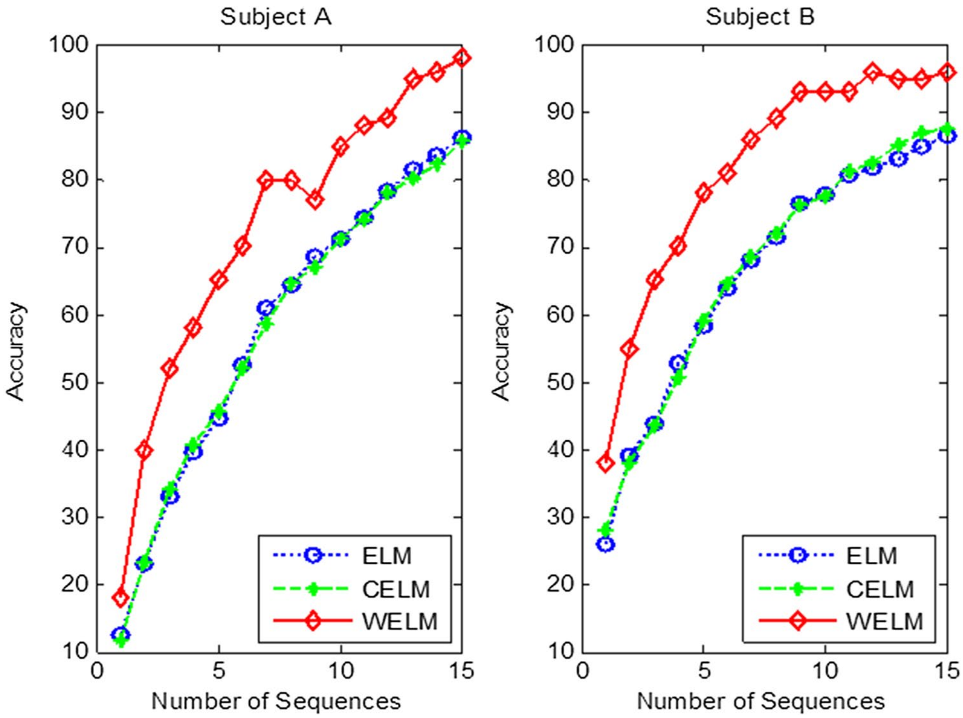

Fig. 5 The accuracy (in percent) of subject A and B using three different ELM classifiers

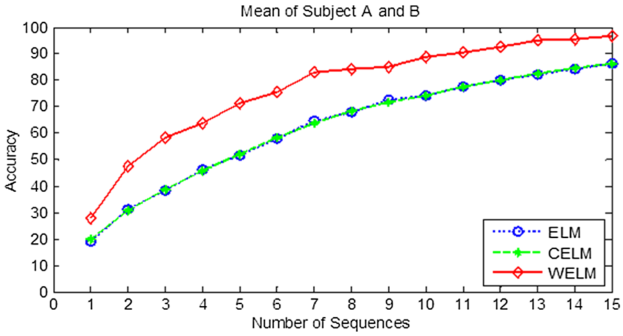

Fig. 6 The average accuracy (in percent) of subject A and B using three different ELM classifiers in different sequences

Moreover, the best competitor of the competition also chose eight channels. The average recognition rates of SVM algorithm (Rakotomamonjy and Guigue 2008) and the proposed method with eight electrodes are shown in Table 3.

Table 4 lists the accuracy of the proposed method and other related methods. In this table, we present recognition rates of test characters in the 1th, 2th, 3th, 4th, 5th, 10th, 15th sequence of two subjects. The RUSBagging SVMs (Shi et al. 2015) achieved 96.7% accuracy in the 15th sequence. Other methods such as Ensemble of SVMs (Rakotomamonjy and Guigue 2008), LDA with PCA (Elsawy et al. 2013), and multi-convolutional neural networks (MCNN) (Cecotti and Gräser 2011) also achieved not bad results. Obviously, the average accuracy of 97% of our proposed method reached the highest level in the 15th sequence.

Table 5 presents the results of the best eight competitors and ours. It can be seen that the first best competitor of the competition obtained an accuracy of 96.5% by using ensemble support vector machines (ESVM) (Rakotomamonjy and Guigue 2008) method. A more detailed description can be found on the web-site http://www.bbci.de/competition/iii/results/#albany.

In order to further compare, we present all the spelling mistakes of test characters of both subjects. See Fig. 7. Small numbers indicate the sequence number of each target character in test set. Subject A's target character

Table 2 The accuracy (%) of two subjects with only eight electrodes and increasing number of sequences

Subject	Number of sequences							
	1	2	4	5	8	10	13	15
A	13	26	37	41	58	65	70	80
B	34	41	57	65	74	80	87	90
Mean	23.5	33.5	47.0	**53.0**	66.0	72.0	78.5	**85.0**

Bold: this method achieved average accuracy rates of 53% and 85% in the 5th and 15th sequences of BCI III competition requirements, respectively

Table 3 The average accuracy (%) of SVM algorithm, and the proposed method with only eight electrodes

Sequence	Method		
	SSVM (Rakotomamonjy and Guigue 2008)	ESVM (Rakotomamonjy and Guigue 2008)	PCA + WELM
5	31	40	53
15	70	80	85

is marked with a green rectangle box, and the predicted character is marked with a red rectangle box. The target character of subject B is marked with a green circle, and the predicted character is marked with a red circle. We can note that every mistaken predicted character is really near the target character in the speller layout. Most of the time, it is either in the same row or in the same column. From the view of the results, the paradigm of P300 speller is still defectiveness. It is highly possible to improve the performance of character recognition by changing the rows or columns flashing model.

5 Conclusion

A state-of-the-art machine learning algorithm for characters recognition has been presented, which is based on WELM and PCA. This method is appropriate for the high-dimensional and imbalance EEG data. Compared with other methods, we finally decided to ignore the rank of electrodes in the speller system for each subject. The feature selection of EEG data involving all channels can actually reduce training time and increase P300 features. Until now, the detection of P300 waves remains a very challenging problem by using machine learning and neural networks. Competition of organizer just provided EEG signals of two subjects. We have not verified the reliability and robustness of our algorithm with a large number of subjects yet. In addition, to choose fewer channels is also an arduous task for the BCI applications. If the algorithm is validated to be effective, we will apply it to a real-time BCI system.

Table 5 The average accuracy (%) of the proposed method and eight best competitors in dataset II of BCI competition III

Team	Method	Accuracy (sequences)		Rank
		15	5	
Our team	PCA + WELM	97	71.5	
Rakotomamonjy and Guigue (2008)	Average ensemble SVMs	96.5	73.5	First
Breiman (1996)	ICA + bagging vote SVM	90.5	55	Second
Liu et al. (2005)	PCA + t-statistic	90	59.5	Third
Hoffmann et al. (2005)	Gradient boosting	89.5	53	Fourth
Breiman (1996)	Bagging with LDA	87.5	57.5	Fifth
Solis-Escalante et al. (2006)	Spatial PCA + RBF SVM	78.5	46	Seventh
Bennett et al.(2008)	Kernel LDA	75	34	Eighth

Table 4 The accuracy (%) of the proposed method and other related methods

Method	Subject	Sequence						
		1	2	3	4	5	10	15
MCNN (Cecotti and Gräser 2011)	A	18	31	50	54	61	82	97
	B	39	55	62	64	77	92	94
	Mean	28.5	43	56	59	69	87	95.5
ESVM (Rakotomamonjy and Guigue 2008)	A	16	32	52	60	72	83	97
	B	35	53	62	64	75	91	96
	Mean	25.5	42.5	57	64	73.5	87	96.5
RSVMs (Shi et al. 2015)	A	16.2	34.5	56.7	–	68	83.7	97.3
	B	36.8	57.8	68.2	–	78.7	95.7	96.1
	Mean	26.5	46.1	62.4	–	73.4	89.7	96.7
WELM	A	18	37	49	56	62	85	98
	B	32	51	61	66	76	91	95
	Mean	25	44	55	61	69	88	96.5
PCA + WELM	A	18	40	52	58	65	85	98
	B	38	55	65	70	78	93	96
	Mean	28	47.5	58.5	64	71.5	89	97

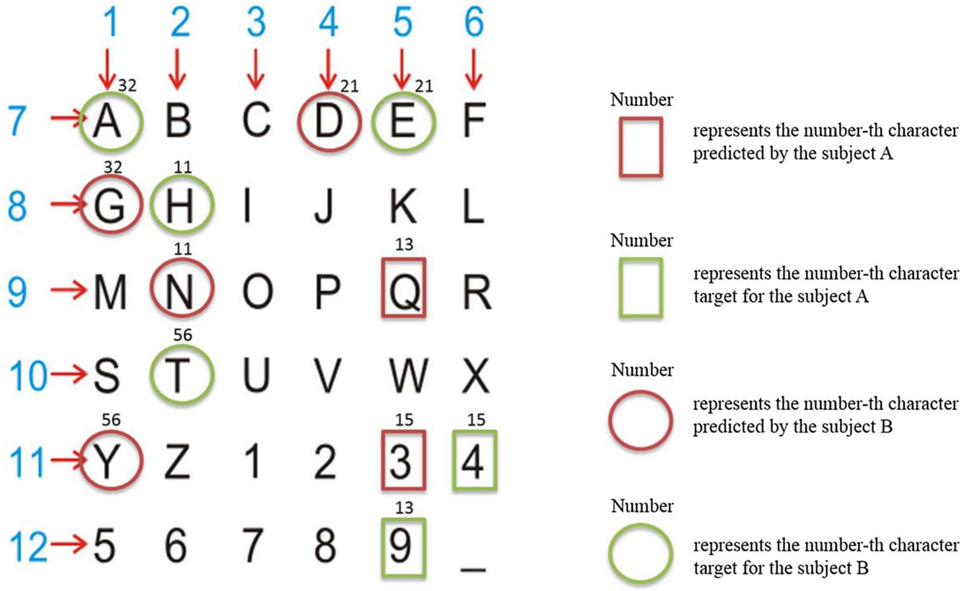
1 2 3 4 5 6
7 A B C D E F
8 G H I J K L
9 M N O P Q R
10 S T U V W X
11 Y Z 1 2 3 4
12 5 6 7 8 9 _
32 21 21 32 11 11 13 56 56 15 15 13
Number
represents the number-th character predicted by the subject A
Number
represents the number-th character target for the subject A
Number
represents the number-th character predicted by the subject B
Number
represents the number-th character target for the subject B

Fig. 7 The locations of the target and predicted characters of subject A and B

Acknowledgements This research was supported by National Natural Science Foundation of China (Grant nos. 61671193 and 61602140), Science and Technology Program of Zhejiang Province (2018C04012, 2017C33049), Science and technology Platform Construction Project of Fujian Science and Technology Department (2015Y2001).

References

Bennett KP, Kunapuli G, Hu J, Pang J-S (2008) Bilevel optimization and machine learning. In: IEEE world congress on computational intelligence. Springer, pp 25–47

Blankertz B, Muller KR, Krusienski DJ, Schalk G, Wolpaw JR, Schlogl A, Pfurtscheller G, Millan JR, Schroder M, Birbaumer N (2006) The BCI competition III: validating alternative approaches to actual BCI problems. IEEE Trans Neural Syst Rehabil Eng 14(2):153–159

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Cecotti H, Gräser A (2011) Convolutional neural networks for P300 detection with application to brain–computer interfaces. IEEE Trans Pattern Anal Mach Intell 33(3):433–445

Chen D, Chen X, Lin F, Kitzman H (2015) Systemize the probabilistic discrete event systems with moore-penrose generalized-inverse matrix theory for cross-sectional behavioral data. J Biom Biostat 6(219):2

Elsawy AS, Eldawlatly S, Taher M, Aly GM (2013) A principal component analysis ensemble classifier for P300 speller applications. In: Image and signal processing and analysis (ISPA), 2013 8th international symposium on, IEEE, pp 444–449

Farwell LA, Donchin E (1988) Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalogr Clin Neurophysiol 70(6):510–523

Fletcher R (1981) Practical methods of optimization: vol. 2: constrained optimization. Wiley, Somerset, p 224

Hoffmann U, Garcia G, Vesin JM, Diserens K, Ebrahimi T (2005) A boosting approach to P300 detection with application to brain–computer interfaces. In: Neural engineering, 2005. Conference proceedings. 2nd international IEEE EMBS conference on, IEEE, pp 97–100

Hoffmann U, Vesin JM, Ebrahimi T, Diserens K (2008) An efficient P300-based brain–computer interface for disabled subjects. J Neurosci Methods 167(1):115–125

Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Neural networks, 2004. Proceedings. 2004 IEEE international joint conference on, IEEE, vol 2, pp 985–990

Huang GB, Chen L, Siew CK et al (2006a) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892

Huang GB, Zhu QY, Siew CK (2006b) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501

Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B (Cybernetics) 42(2):513–529

Krusienski DJ, Sellers EW, McFarland DJ, Vaughan TM, Wolpaw JR (2008) Toward enhanced P300 speller performance. J Neurosci Methods 167(1):15–21

Liu Y, Zhou Z, Hu D, Dong G (2005) T-weighted approach for neural information processing in P300 basedbrain-computer interface. In: International conference on neural networks and brain, 2005. ICNN&B'05, vol 3. IEEE, pp 1535–1539

Liu YH, Weng JT, Kang ZH, Teng JT, Huang HP (2010) An improved SVM-based real-time P300 speller for brain–computer interface. In: Systems man and cybernetics (SMC), 2010 IEEE international conference on, IEEE, pp 1748–1754

Mirghasemi H, Fazel-Rezai R, Shamsollahi M (2006) Analysis of P300 classifiers in brain computer interface speller. In: Engineering in medicine and biology society, 2006. EMBS'06. 28th annual international conference of the IEEE, IEEE, pp 6205–6208

Peng Y, Lu BL (2017) Discriminative extreme learning machine with supervised sparsity preserving for image classification. Neurocomputing 261:242–252

Qassim YT, Cutmore TR, James DA, Rowlands DD (2013) Wavelet coherence of EEG signals for a visual oddball task. Comput Biol Med 43(1):23–31

Rakotomamonjy A, Guigue V (2008) BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller. IEEE Trans Biomed Eng 55(3):1147–1154

Schalk G, McFarland DJ, Hinterberger T, Birbaumer N, Wolpaw JR (2004) BCI 2000: a general-purpose brain–computer interface (BCI) system. IEEE Trans Biomed Eng 51(6):1034–1043

Selim AE, Wahed MA, Kadah YM (2014) Electrode reduction using ICA and PCA in P300 visual speller brain–computer interface system. In: Biomedical engineering (MECBME), 2014 middle east conference on, IEEE, pp 357–360

Shi LC, Lu BL (2013) EEG-based vigilance estimation using extreme learning machines. Neurocomputing 102:135–143

Solis-Escalante T, Gentiletti GG, Yanez-Suarez O (2006) Single trial P300 detection based on the empirical mode decomposition. In: 28th Annual international conference of the IEEE engineering in medicine and biology society, 2006. EMBS'06. IEEE, pp 1157–1160

Shi X, Xu G, Shen F, Zhao J (2015) Solving the data imbalance problem of P300 detection via random under-sampling bagging SVMs. In: Neural networks (IJCNN), 2015 international joint conference on, IEEE, pp 1–5

Vaughan TM, Wolpaw JR (2006) The third international meeting on brain-computer interface technology: making a difference. IEEE Trans Neural Syst Rehabil Eng 14(2):126–127

Zhang JC, Xu YQ, Yao L (2007a) P300 detection using boosting neural networks with application to BCI. In: Complex medical engineering, 2007. CME 2007. IEEE/ICME international conference on, IEEE, pp 1526–1530

Zhang R, Huang GB, Sundararajan N, Saratchandran P (2007b) Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Trans Comput Biol Bioinform 4(3):485–495

Zong W, Huang GB, Chen Y (2013) Weighted extreme learning machine for imbalance learning. Neurocomputing 101:229–242

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.