Medical & Biological Engineering & Computing manuscript No.
(will be inserted by the editor)

A comparison of feature extraction strategies using wavelet dictionaries and feature selection methods for single trial P300-based BCI

R. Acevedo · Y. Atum · I. Gareis · J. Biurrun Manresa · V. Medina Bañuelos · L. Rufiner

Received: date / Accepted: date

Abstract The P300 component of event-related potentials (ERPs) is widely used in the implementation of brain computer interfaces (BCI). In this context, one of the main issues to solve is the binary classification problem that entails differentiating between electroencephalographic (EEG) signals with and without P300. Given the particularly unfavorable signal-to-noise ratio (SNR) in the single-trial detection scenario, this is a challenging problem in the pattern recognition field. To the best of our knowledge, there are no previous experimental studies comparing feature extraction and selection methods for single trial P300-based BCIs using unified criteria and data. In order to improve the performance and robustness of single-trial classifiers, we analyzed and compared different alternatives for the feature generation and feature selection blocks. We evaluated different orthogonal decompositions based on the Wavelet Transform for feature extraction, as well as different filter, wrapper and embedded alternatives for feature selection. Accuracies over 75% were obtained for most of the analyzed strategies with a relatively low computational cost, making them attractive for a practical BCI implementation using inexpensive hardware.

Keywords Brain-Computer Interface · Feature Generation and Selection · P300

1 Introduction

The natural communication pathways of the brain with the outside world (i.e. peripheral nerves and muscles) can be damaged irreversibly as an aftermath of accidents or illness. This damage

R. Acevedo · Y. Atum · I. Gareis · J. Biurrun Manresa · L. Rufiner
Facultad de Ingenieria, Universidad Nacional de Entre Rios
Oro Verde, Argentina
Tel.: +54-343-4975100 - Fax: +54-343-4975101
E-mail: racevedo@ingenieria.uner.edu.ar

I. Gareis · L. Rufiner
Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional (SINC-UNL-CONICET)
Santa Fe, Argentina

J. Biurrun Manresa
Instituto de Investigación y Desarrollo en Bioingeniería y Bioinformática (IBB), CONICET-UNER
Oro Verde, Argentina

V. Medina Bañuelos
Universidad Autonoma Metropolina, Unidad Iztapalapa
Mexico D.F., Mexico

usually reduces the ability of a person to communicate and interact with the environment. In extreme cases, and despite the preservation of cognitive abilities, patients completely lose muscle control (even ocular control), falling into a condition called *locked-in syndrome* [42]. Nowadays, there are devices designed to partially replace or supplement the functions of the brain natural communication pathways, allowing patients to command computers and thereby interact with the outside world; these systems are called *brain-computer interfaces* (BCI) [46].

A BCI can be based on different physiological phenomena. One of the most widely used is a component of the *event-related potential* (ERP) called P300. When an infrequent or particularly significant auditory, visual or somatosensory stimulus is mixed with frequent or routine stimuli, a P300 is generally evoked over the parieto-occipital cortex, and recorded through the electroencephalogram (EEG) [31]. This phenomenon can be used to implement a BCI called P300 speller, in which the user is prompted to select symbols from a matrix on a computer screen [10]. To determine which symbol the user selected, the system must be able to solve the binary classification problem that entails differentiating between electroencephalographic (EEG) signals with and without P300. To do this, it is common to repeat each stimulus several times to record and subsequently average all the responses. The repetition improves the classification accuracy by enhancing the signal-to-noise ratio (SNR) of the responses. However, it comes at the cost of increasing the time necessary to make a decision, which fatigues the user and may reduce the information transfer rate of the system. It is therefore highly desirable to solve the single trial classification problem with the highest possible accuracy. In this work, we will focus in the binary classification of single trial post-stimulus signals.

The performance of a BCI is highly dependent on the classification and feature extraction methods used to predict the user's intention. Several feature extraction methods have been proposed in the context of P300 based BCIs, among which can be mentioned peak picking and area computation [10], time-frequency representations [4], matched filtering [41], piecewise prony method [14] and temporal features [15]. In particular, wavelet decompositions have been applied in different ways, either using continuous or dyadic transforms [4,25,28,40]. Wavelet representations have a compact support that allows to capture local characteristics of the P300 wave in the time-frequency plane that are discriminative in a computationally efficient way [34]. They also provide a flexible framework, allowing to compactly represent several different types of signal characteristics. Moreover the existence of fast algorithms, combined with feature selection methods allow low-cost BCI hardware implementations. As an example, Saavedra *et al.* [35] proposed a new approach of wavelet denoising, taking into account the correlation between channels, with the aim of improving the SNR ratio in ERP recordings. This was done by combining phase and amplitude information in the wavelet domain to automatically select a time window on each channel that maximizes the separability of classes.

Alternatively, Genetic Algorithms (GAs) have also been previously used for feature generation and selection [7,8,16]. Recursive Feature Elimination (RFE) method has also been proposed with varying degrees of success, for selection of single features or complete channels [32,33]. Worth mentioning are the articles of Lindig-León *et al.* and Kindermans *et al.* that use the same database as the main one in the present work [17,20]. The first one developed a method that aims to reduce the compromise between spelling rate and precision. They used a Bayesian approach to estimate the *a posteriori* probability associated with the classification of each target using a linear discriminant as classifier. The second one proposed a BCI capable of spelling without any training, using inter-subject information as well as language models based on hidden Markov models, and featuring an unsupervised classifier.

There are exhaustive reviews in the literature on classification algorithms [21] as well as on feature extraction and classification algorithms [2,22]. However, these works are limited to reviewing techniques used by different authors on different studies without unified criteria and data. On the other hand, different feature extraction strategies have also been compared within single articles. Amini *et al* used intelligent segmentation technique that approximates P300 signals with non-

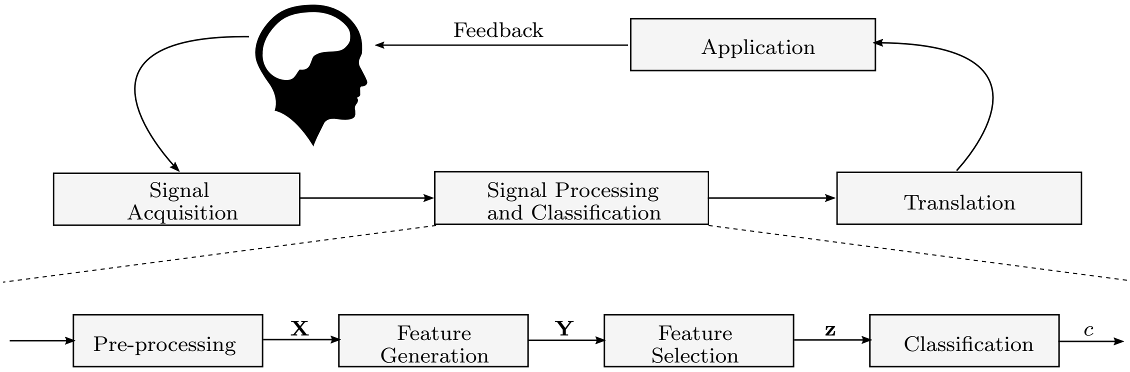


Fig. 1: Typical architecture of a brain computer interface.

uniform straight lines [1], Turnip *et al* compared AAR (Adaptive Autoregressive Model), JADE (Joint Approximation Diagonalization of Eigen-matrices) and SOBI (second- order blind identification) algorithms [43] and Wang *et al* proposed selection of the wavelet coefficients that best represent the P300, according to a Fisher's distance criterion [44]. All these works have been carried out on averaged responses which, as previously stated, constitute an easier problem to solve than single trial classification. However, to the best of our knowledge, there are no previous experimental studies comparing feature extraction and selection methods for *single trial* P300-based BCIs using unified criteria and data.

The aim of this study was to analyze and compare different alternatives for the feature generation and feature selection stages in order to improve the performance and robustness of single-trial classifiers. Different orthogonal decompositions based on the Wavelet Transform for feature generation, as well as different filter, wrapper and embedded alternatives for feature selection were evaluated, in order to select the best method based on commonly used performance indexes for BCIs.

2 Materials and Methods

2.1 BCI architecture

The typical architecture of a BCI, as shown in Figure 1, consists of a signal acquisition block, a signal processing and classification block and a translation block. Particularly, the signal processing and classification block can be further divided into pre-processing, feature generation, feature selection and classification sub-blocks. The pre-processing block receives the temporal signals from the S sensors as input and prepares them for the following processing stages through different techniques, such as filtering or outlier remotion, and segments an N length temporal window of each channel discrete signal $\mathbf{x}_i$ to present a sample matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_S] \in \mathbb{R}^{N \times S}$ as output. The feature generation block transforms the original signals $\mathbf{x}_i$ into their $\mathbf{y}_i$ representations through basis changes and outputs the representations $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_S] \in \mathbb{R}^{N \times S}$. The feature selection process reduces the total number of features from a $N \times S$ matrix to a $\mathbf{z} \in \mathbb{R}^K$ vector, to be presented to the classification block. The sets containing the training patterns at each stage will be denoted by $\mathfrak{T}_{\mathbf{X}}$, $\mathfrak{T}_{\mathbf{Y}}$ and $\mathfrak{T}_{\mathbf{z}}$ and the sets containing the testing patterns by $\mathfrak{S}_{\mathbf{X}}$, $\mathfrak{S}_{\mathbf{Y}}$ and $\mathfrak{S}_{\mathbf{z}}$. The following sections will describe alternatives for feature generation and feature selection sub-blocks, which are the main focus of the present article.

2.2 UAM Database and preprocessing

The dataset[1] used to train and evaluate the different methods was recorded at the Neuroimaging Laboratory of the Department of Electrical Engineering of the UAM (México) using the P300Speller application of the BCI2000 system [39]. Each visual stimulus lasted 62.5 ms, with fixed interstimulus intervals of 125 ms. A g.tech g.USBamp amplifier was used to record 10 EEG channels (Fz, C3, Cz, C4, P3, Pz, P4, PO7, PO8 and Oz) with a sampling frequency of 256 Hz. Recordings were taken from 18 healthy subjects, aged between 21 and 25 years old. For each subject, 3780 post-stimulus signals were recorded, from which 630 contained a P300 response. An 8th-order Chebyshev bandpass filter (0.1-60 Hz) and a notch filter (60 Hz) were applied to the recorded signals, which were afterwards down-sampled to 64 Hz and segmented to obtain 64-sample epochs. All experiments in this work contemplated a training stage and a testing stage. The available epochs were balanced so each class (with P300 and without P300) was equally represented. As a result, 630 epochs from each class were used from each subject: 480 epochs were used for training and 150 epochs were used for testing. The same partition was used in all the experiments to allow a direct comparison between the proposed strategies. The system was trained and tested for each individual, and the results presented are averaged across subjects.

2.3 Feature generation

From the perspective of signal spaces, a discrete signal $\mathbf{x} \in \mathbb{R}^N$ can be written as $\mathbf{x} = \{x_n\}$, with $1 \leq n \leq N$, where it is implicitly represented in the *canonical* basis as a linear combination of displaced Kronecker deltas according to Equation 1:

$$x[n] = \sum_{k=1}^{N} x[k]\delta[n,k]. \tag{1}$$

Given a basis transformation matrix $\mathbf{\Phi} \in \mathbb{R}^{N \times N}$, it is possible to write $\mathbf{x} = \mathbf{\Phi}\mathbf{y}$, where $\mathbf{y}$ can be seen as a representation of $\mathbf{x}$ on another basis. The basis change operation is performed using Equation 2:

$$\mathbf{y} = \mathbf{\Phi}^{-1}\mathbf{x}. \tag{2}$$

This new representation of the original signal could be used to highlight certain features; in this particular case, we are specifically interested in the P300 wave, which is immersed in the background EEG. It should be mentioned that in the case of unitary transformations (such as the ones used in this work), $\mathbf{\Phi}^{-1}$ from Equation 2 is equal to the conjugate transpose of $\mathbf{\Phi}$, which notably simplifies the computations. The canonical basis is implicitly used for the raw temporal patterns, which consist of the concatenation of the 64-sample epochs from the ten channels. Bases obtained through the Discrete Dyadic Wavelet Transform (DDWT) and the Wavelet Packet Transform (WPT) are further evaluated in this work.

Discrete Dyadic Wavelet Transform

The Discrete Dyadic Wavelet Transform (DDWT) is one of the most commonly used methods to generate orthogonal bases from the Wavelet Transform, due to its simple and inexpensive computational implementation. A dyadic wavelet is a function $\psi(t) \in L^2(\mathbb{R})$ such that the family of functions $\psi_{k,n}(t) = 2^{k/2}\psi(2^k t - n)$ for $k, n \in \mathbb{Z}$ form an orthonormal basis in $L^2(\mathbb{R})$. To define useful orthogonal wavelets, the *multiresolution analysis* (MRA) is commonly used. The MRA establishes a relation between the pair of wavelet and scaling functions and a pair of quadrature mirror filters:

[1] Available in `http://akimpech.izt.uam.mx/p300db/doku.php`

the low-pass filter is associated with the scaling function and the high-pass filter is associated with the wavelet function. Using the results derived from the MRA and appropriately discretizing the signal and filters, it is possible to obtain the following recursive expressions:

$$\begin{aligned} c_{m-1}[n] &= \sqrt{2}\sum_{i=-\infty}^{+\infty} h[i-2n]c_m[i], \\ d_{m-1}[n] &= \sqrt{2}\sum_{i=-\infty}^{+\infty} g[i-2n]c_m[i], \end{aligned} \tag{3}$$

where c_m are the approximation coefficients, d_m are the detail coefficients in scale m, $h[i]$ and $g[i]$ are i^{th} samples of the impulse responses belonging to the discretized low-pass and high-pass quadrature mirror filters associated with the scaling and wavelet functions, respectively. Equations 3 state that the approximation and detail coefficients on a certain scale can be obtained through the approximation coefficients from the previous scale. This property can be exploited to efficiently compute the DDWT through a filter bank. Equivalently to Equation 3, a recursive expression can be used to reconstruct the transformed signal:

$$c_m[n] = \sum_{i=-\infty}^{+\infty} h[n-2i]c_{m-1}[i] + \sum_{i=-\infty}^{+\infty} g[n-2i]d_{m-1}[i], \tag{4}$$

where $k < l$ and $m = l, ..., k-1$.

It is important to mention that most of the wavelet and scaling functions cannot be written in closed-form and are defined trough their corresponding filters. To obtain the discrete ψ_{m-1} and ϕ_{m-1} associated with the i^{th} coefficient of the $m-1$ scale, vectors with value one in the sample i and zeros otherwise have to be inverted using Equation 4. This process is called the *cascade* algorithm and it allows to obtain temporal functions ψ and ϕ. Using the the discretized versions of ψ and ϕ, the DDWT can be written as a discrete basis change matrix:

$$\mathbf{\Phi}^{-1}_{\mathrm{DDWT}} = \Big[\{\phi_l[i-2^{l-k+1}n]\}_n \in \mathbb{Z}; \{\psi_m[i-2^{m-k+1}n]\}_{k-1<m\leq l, n\in\mathbb{Z}} \in \mathbb{Z}\Big], \tag{5}$$

where $0 < i \leq N-1$ indicates the rows and the columns vary with m, n and l to complete N columns in total.

In the case of the DDWT patterns, the multiresolution decomposition was performed up to level 6. The decomposition was applied independently to each channel and consisted of the coefficients corresponding to the first (32 coefficients), second (16 coefficients), third (8 coefficients), fourth (4 coefficients), fifth (2 coefficients) and sixth (1 coefficient) details and the sixth approximation (1 coefficient), totaling 64 coefficients per channel, and consequently 640 features were obtained per pattern. The Daubechies wavelet with 4 vanishing moments (db4) with periodic padding was used [2].

Wavelet Packet Transform

The *Wavelet Packet Transform* (WPT) is a generalized version of the DDWT. It performs the decomposition of the high frequency components (details) as well as the low frequency components (approximations). As in the DDWT, the discrete temporal waveforms associated with each coefficient in each scale can be obtained using the cascade algorithm to get a dictionary of waveforms $\mathfrak{D}$. However, the decomposition $\mathfrak{D}$ in the WPT can be redundant, i.e. the number of waveforms in $\mathfrak{D}$ is larger than the waveforms' dimension. There are nevertheless several different possible subsets

[2] The DDWT decompositions were obtained using the MATLAB Wavelet Toolbox (Mathworks, Inc., Natica, MA, USA)

γ of functions capable of constituting orthogonal bases. In fact, with a depth k, a total of $2^{2(k-1)}$ different orthogonal bases $\mathbf{\Phi}_\gamma, \gamma \in \mathit{\Gamma}$ can be extracted from the complete WPT decomposition dictionary $\mathfrak{D}$ [36].

Different methods have been proposed to select the best orthogonal basis among all the aforementioned possibilities. Some of these methods exploit the particular tree structure of the WPT to implement a fast algorithm to perform this task. Specifically, Wikerhauser and Coifman proposed an algorithm to adaptively estimate the best basis for a given signal $\mathbf{x}$, according to a convenient cost function [6]. The cost function $J : \mathbf{y} = y_i \rightarrow \mathbb{R}$ has to be additive, i.e. $J(0) = 0$ and $J(\mathbf{y}) = J(\mathbf{y}_i)$. Using this notation, the problem can be written as:

$$\min_{\gamma \in \mathit{\Gamma}} J(\mathbf{\Phi}_\gamma^{-1}\mathbf{x}). \tag{6}$$

The most common form of cost function used in this cases is the Shannon Entropy, since the choice of the appropriate basis is usually made with the aim of retaining the greatest amount of information in the smallest number of coefficients [38]. The fast method used to solve problem is called Best Orthogonal Basis (BOB) [6]. There is a variant of this algorithm that can be applied to several signals called Joint BOB, used to obtain the best representation in terms of the efficiency of all the signals representations simultaneously. However, these methods do not necessarily provide good representations in terms of classification performance.

For the classification problem, the cost function should measure the discrimination power of each possible decomposition rather than the flatness of the signal representation energy distribution. To tackle this problem, Saito and Coifman proposed a method called Local Discriminant Basis (LDB) that, in a similar way to the BOB algorithm, allows to efficiently select an orthogonal basis from a WPT dictionary with the objective of maximizing the separation between the different classes [36]. Several discrimination measures have been proposed, among them are the *relative entropy* (also called Kullback-Leibler distance or I-Divergence), the *J-Divergence* and the *sum of squares* [37]. As in the case of the BOB algorithm, the cost function used in LDB should be additive. It is important to remark that along the optimal basis the LDB algorithms also provides a score for each of its elements, allowing to rank them according to their discrimination power.

As for the DDWT, the WPT decomposition was performed up to level 6. The Daubechies wavelet with 4 vanishing moments (db4) with periodic padding was used and the decompositions were obtained [3]. As expected, the BOB algorithm did not perform better than DDWT on preliminary ERP classification experiments [30]. For this reason, only the LDB algorithm has been considered in this work to extract orthogonal bases from the WPT dictionary [37]. As mentioned before, the LDB algorithm returns a ranking for the basis elements selected. The number of coefficients was hence reduced using this information, keeping the best 18 coefficients in each channel [29].

2.4 Feature selection

Feature selection methods can be classified in filter, wrapper and embedded models [12]. In the *filter model*, the feature evaluation criteria is independent of the classification stage. Therefore, the classifier learning process is not included in the feature selection method [12]. Two simple filter feature selection methods were tested in this work. The first one was the remotion of the lower ranking basis elements provided by the LDB algorithm and the second one was the remotion of coefficients after *wavelet smoothing* (WS). For the *wrapper model*, the classifier is considered as a black box and its performance is used to select the optimal feature subset [45]. In this case, a feedback loop is established between the classifier and the feature selection block. Different random

[3] The WPT decompositions were obtained using the MATLAB Wavelet Toolbox (Mathworks, Inc., Natica, MA, USA)

search techniques can be used in this model, from which *Genetic Algorithms* (GAs) were chosen to be tested in this study. Lastly, in the *embedded model*, the feature selection method and the classifier can not be separated [45]. This is due to the fact that both stages are integrated, performing the feature selection during the classifier's learning process. Generally, the selection method is specific for each type of classifier. An example of embedded model is the recursive feature elimination (RFE) algorithm tested in this work.

Although the aforementioned is generally true, it is incorrect to state that a particular feature selection methodology belongs exclusively to an embedded or wrapper model. Instead, this classification depends on the relationship between the feature selection method and the classifier. In this way, most algorithms (including RFE and GAs) can be implemented as part of a wrapper or an embedded model [5].

Wavelet Smoothing

When performing a decomposition trough the DDWT, the frequency range of each of the *details* and that of the resulting *approximation* is known, so it is possible to eliminate the ranges of frequencies that are not of interest by zeroing the corresponding coefficients. The smoothing process consists on removing the coefficients corresponding to the the components associated with the signal's fastest changes. It can be considered as a projection of the signal representation $\mathbf{y}$ into an orthogonal subcomplete basis [30].

Considering that a large part of the energy in the frequency spectrum of the P300 is below 8 Hz [13], two alternatives of wavelet smoothing were tested. The first one consisted on removing the coefficients corresponding to the first detail (≈16-32 Hz), thus generating patterns consisting of 320 features (32 per channel) which will subsequently be called *DDWT-D1*. The second alternative consisted on removing the coefficients corresponding to the first and second details (≈8-32 Hz), obtaining patterns of 160 features that will be called *DDWT-D1D2*.

Genetic Algorithm

GAs are based on the manipulation of a population of possible solutions to a given problem. These solutions are encoded as binary chains that represent the genetic material of a population of individuals [24]. Artificial operators of selection, crossover and mutation are applied in a stochastic search process to find the best individual (best solution) thus simulating a natural evolutionary process. Each potential solution is associated with a fitness value that measures its goodness. The fitness function is related to the performance of the classifier for patterns generated with this particular subset of features (dimensionality was also taken into account). The method's implementation can be seen in Algorithm 1.

Algorithm 1 Genetic Algorithm Wrapper's pseudocode

```
procedure GAWRAPPER($\mathfrak{T}_{\mathbf{Y}}$)                ▷ Input: feature vectors
    initialize(P): a population of $p$ individuals is randomly initialized.
    Fitness=evaluate(P): for each individual, a new set of patterns is generated (training and validation),
  the classifier is trained and the fitness value is computed based on its performance.
    while $best$(Fitness) $\leq$ $Required\_Fitness$ do
        S=select(P): selection of new parents is performed.
        O=crossover(S): crossover is used for each pair of parents to generate offsprings.
        P=mutate(O): each offspring is mutated.
        Fitness=evaluate(P).
return $\mathbf{P}_{best\_of\_all}$                           ▷ Output: selected features
```

For the wrapper task, the number of bits for each individual is its total number of features, so that each bit is related with a particular feature of the input pattern. In order to reduce the size of search space, a unique template for all the channels is used. In this way, the i^{th} bit value indicates the presence (1) or absence (0) of the i^{th} feature for all channels. The population size was set to 100 individuals. Each individual can be represented with a vector $\mathbf{p} \in \mathbb{Z}_2^K$ where K is the number of bits and is chosen equal to the number of features per channel (64, 32 or 16). This result, along with the number of active bits of the individual, is used to compute the fitness function

$$A(\mathbf{p}; \mathfrak{T}_{\mathbf{z}}) = w_a \text{Accuracy}(\mathbf{p}; \mathfrak{T}_{\mathbf{z}}) + w_c \frac{1}{\sum_{i=1}^{K} \mathbf{p}[i]}. \quad (7)$$

where $w_a = 0.80$, $w_c = 0.20$ and Accuracy$(\mathbf{p}; \mathfrak{T}_{\mathbf{z}})$ consists on the accuracy obtained training an LDA using a portion of $\mathfrak{T}_{\mathbf{z}}$ and testing it with its other portion, in both cases using only the coefficients selected by $\mathbf{p}$. This equation is defined to maximize the classification accuracy and minimize the number of features simultaneously, with more emphasis in the accuracy [48]. The selection of parents was implemented using tournament selection with competitions between two randomly chosen individuals [27]. This process is repeated 20 times in order to select 20 individuals from the population as progenitors for the new generation [23]. Simple crossover with a 0.95 probability and single-bit mutation with a probability of 0.05 were employed [11].

Recursive Feature Elimination

The RFE is an iterative process consisting of three main steps: training the classifier, establishing the feature ranking and removing the feature from the lowest ranking position [12]. In this work, the RFE algorithm uses the absolute value of the classifier weights as the criterion for feature ranking. The pseudocode of this process is presented in Algorithm 2. Note that the features selected using RFE are not necessarily the same for all channels, in contrast with the other methods proposed in which the same coefficients are selected for each channel. For this particular implementation of the RFE algorithm, only one feature was removed at every iteration and the loop ended when only one feature remained in the pattern. The criteria used to choose the subset of features that represents the best solution was the classifier accuracy.

Algorithm 2 Recursive Feature Elimination pseudocode

```
procedure RFE($\mathfrak{T}_{\mathbf{Y}}$)                                   ▷ Input: the feature vector
    Vectorization. Reshapes the matrices $\mathbf{Y} \in \mathbb{R}^{N \times S}$ to vectors $\mathbf{z} \in \mathbb{R}^K : K = N \times S$
    The training set $\mathfrak{T}_{\mathbf{z}}$ is divided into training ($\mathfrak{Tt}_{\mathbf{z}}^1$) and validation ($\mathfrak{Tv}_{\mathbf{z}}^1$) subsets.
    for i=1:K do
        Classifier training. Train the $i^{\text{th}}$ classifier $C_i$ using $\mathfrak{Tt}_{\mathbf{z}}^i$.
        Storage of accuracy. Evaluate the $C_i$ using $\mathfrak{Tv}_{\mathbf{z}}^i$ and storage the .
        Determination of the feature to remove. Find the classifier weight $w_n$ with the lowest absolute value.
        Removal of worst remaining feature. Set $\mathfrak{Tt}_{\mathbf{z}}^{i+1}$ and $\mathfrak{Tv}_{\mathbf{z}}^{i+1}$ as $\mathfrak{Tt}_{\mathbf{z}}^i$ and $\mathfrak{Tv}_{\mathbf{z}}^i$ without the $n^{\text{th}}$ feature.
return Subset of features with highest accuracy.                      ▷ Output: selected features
```

2.4.1 Classifier

The Fisher Linear Discriminant Analysis (LDA) was chosen for the classification stage. Even though the LDA has a simple implementation and low computational cost, it delivers similar performance in comparison with more complex methods for the P300 classification problem [3].

The LDA, as any binary linear classifier, can be characterized by a function $g(\mathbf{z}) = \mathbf{w}^T \mathbf{z} + b$, where $\mathbf{w} = [w_1, ..., w_K]$ is the projection vector, $\mathbf{z} \in \mathbb{R}^K$ is the input vector and b is the bias term.

The classification function assigns the class label c to each pattern $\mathbf{z}$ depending on the sign of the function $g(\mathbf{z})$. It is assumed that the probability distributions of each class are Gaussian.

To obtain the optimal projection vector $\mathbf{w}$ cost function that measures the ratio of the between-class variance to the within-class variance is maximized. Defining as $\mathbf{m}_i$ the mean of all $\mathbf{z} \in \mathfrak{T}_{\mathbf{z}}^i$, where $\mathfrak{T}_{\mathbf{z}}^i$ is the set containing the training patterns belonging to class i, the between-class covariance matrix is defined as

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T, \tag{8}$$

and the within-class covariance matrix as

$$\mathbf{S}_W = \sum_{\mathbf{z} \in \mathfrak{T}_{\mathbf{z}}^i} (\mathbf{z} - \mathbf{m}_1)(\mathbf{z} - \mathbf{m}_1)^T + \sum_{\mathbf{z} \in \mathfrak{T}_{\mathbf{z}}^i} (\mathbf{z} - \mathbf{m}_2)(\mathbf{z} - \mathbf{m}_2)^T. \tag{9}$$

Using these matrices the cost function for the LDA can be written as

$$F(\mathbf{w}; \mathfrak{T}_{\mathbf{z}}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}. \tag{10}$$

The direction of the $\mathbf{w}$ that maximizes F can be found by

$$\mathbf{w} \propto S_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1). \tag{11}$$

The optimal bias term b is chosen to be between the mean of each projected class

$$b = \frac{1}{2}\mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1). \tag{12}$$

It is assumed that the probability distributions of feature vectors from each class are Gaussian. It is important to highlight the effect of working with balanced data on the LDA performance. The imbalanced data sets will not have effects on the projection vector $\mathbf{w}$ if the two covariance matrices are identical, which is not true in most cases [47]. For this reason, balanced data was used as described in the database description section.

The implementation of LDA provided in the PRTools library was used to perform the training and evaluation of the patterns [9].

2.5 Performance indexes

The indexes selected to evaluate the performance are the Accuracy (Acc), Sensitivity (Sen), and Specificity (Spe), defined as:

$$\text{Acc} = \frac{TP + TN}{NT}, \quad \text{Sen} = \frac{TP}{FN + TP}, \quad \text{Spe} = \frac{TN}{FP + TN}. \tag{13}$$

where TP, TN, FP and FN are the number of true positives (correctly classified patterns with P300), true negatives (correctly classified patterns without P300), false positives (incorrectly classified patterns without P300) and false negatives (incorrectly classified patterns with P300), respectively, and NT is the total number of patterns in the test partition. Since in the case of the P300-Speller the available data is imbalanced the balanced accuracy (BalAcc), was also computed. This metric is designed to be insensitive to unbalanced data and can facilitate the analysis of the results. It is defined as:

$$\text{BalAcc} = \frac{\text{Spe} + \text{Sens}}{2}. \tag{14}$$

The number of features obtained with each method is also an important index to evaluate, particularly in potential scenarios considering low-cost BCI hardware implementations with limited processing power. In case of similar classification performance, methods resulting in fewer features are preferable.

2.6 Data analysis and statistics

A first analysis was performed within the different type of methods (i.e. feature selection from temporal patterns, feature generation from DDWT patterns, feature generation and selection from DDWT patterns), in order to determine which of the alternatives within that family provided the best performance. A final comparison using the best performing methods was carried out afterwards. To determine if there were statistically significant differences in performance indexes among strategies, the normality of the performance index distributions was first evaluated using the Shapiro-Wilk test. In most cases, data distributions failed the test, so the non-parametric Friedman test was used to evaluate differences in performance between methods. In case that this test detected a significant difference between groups, the Tukey test on mean ranks for *post hoc* comparisons between pairs of methods was used. The level of statistical significance was set at 0.05[4].

3 Results

Performance indexes obtained by applying each of the methods are shown in Table 1, grouped according to the experiments described below.

Table 1: Performance indexes means and standard deviations.

Patterns	Sensitivity	Specificity	Accuracy	# of features
Temporal	0.696 ± 0.05	0.695 ± 0.06	0.695 ± 0.05	640
Temporal and GAs	0.766 ± 0.07	0.774 ± 0.07	0.770 ± 0.05	131 ± 29
Temporal and RFE	0.768 ± 0.08	0.786 ± 0.08	0.777 ± 0.07	120 ± 74
DDWT	0.695 ± 0.05	0.696 ± 0.06	0.695 ± 0.05	640
DDWT-D1	0.770 ± 0.06	0.767 ± 0.06	0.768 ± 0.06	320
DDWT-D1D2	0.758 ± 0.06	0.739 ± 0.05	0.747 ± 0.05	160
DDWT-D1 and GAs	0.777 ± 0.06	0.786 ± 0.07	0.781 ± 0.06	163 ± 41
DDWT-D1D2 and GAs	0.732 ± 0.07	0.763 ± 0.07	0.748 ± 0.06	140 ± 16
DDWT-D1 and RFE	0.766 ± 0.07	0.789 ± 0.07	0.778 ± 0.06	111 ± 58
DDWT-D1D2 and RFE	0.727 ± 0.07	0.749 ± 0.08	0.738 ± 0.06	63 ± 34
WPT and LDB	0.785 ± 0.08	0.785 ± 0.08	0.785 ± 0.08	180

3.1 Feature generation using DDWT and smoothing (DDWT, DDWT-D1 and DDWT-D1D2)

Significant differences in performance indexes were found for the proposed methods and the reference method (Friedman, $p < 0.001$ for all variables). The DDWT-D1 and DDWT-D1D2 methods resulted in significantly higher sensitivity than the reference method (Tukey, $p < 0.05$), and there were no significant differences between them (Tukey, $p > 0.05$). Furthermore, the DDWT-D1 resulted in significantly higher specificity than the reference and DDWT-D1D2 methods (Tukey, $p < 0.05$). The DDWT-D1 and DDWT-D1D2 methods resulted in significantly better accuracy than the reference method (Tukey, $p < 0.05$), and there were no significant differences between them (Tukey, $p > 0.05$). In summary, the DDWT-D1 method was selected for the final analysis, since its sensitivity and accuracy rates were comparable to DDWT-D1-D2, but it demonstrated better specificity.

[4] Statistics were carried out using SigmaPlot 12 from Systat Software, Inc., San Jose California USA, www.systatsoftware.com.

3.2 Feature selection in temporal patterns(Temporal+RFE and Temporal+GAs)

Significant differences in performance indexes were found when feature selection methods were applied to raw temporal patterns (reference method) (Friedman, $p < 0.001$ for all variables). Post hoc analysis revealed that both alternatives for feature selection provided sensitivity, specificity, and accuracy rates significantly higher the reference method with a lower number of features (Tukey, $p < 0.05$). Furthermore, there were no significant differences between feature selection methods for any of the indexes (Tukey, $p > 0.05$), so that both methods were considered for the final analysis.

3.3 Feature generation and selection in DDWT patterns (DDWT-D1+GAs, DDWT-D1+RFE, DDWT-D1D2+GAs and DDWT-D1D2+RFE)

Significant differences were found in sensitivity (Friedman, $p < 0.001$), specificity (Friedman, $p < 0.002$), accuracy (Friedman, $p < 0.001$), and number of features (Friedman, $p < 0.001$). All feature selection and generation alternatives performed better than the reference method (Tukey, $p < 0.001$ for all indexes). In particular, DDWT-D1D2+RFE and DDWT-D1D2+GAs resulted in significantly lower sensitivity, specificity and accuracy than the other proposed alternatives (Tukey, $p < 0.05$). Furthermore, DDWT-D1+ RFE presented similar values of sensitivity, specificity and accuracy compared to DDWT-D1+GAs and DDWT-D1, but using significantly fewer features (Tukey, $p < 0.05$), so it was selected for the final analysis and the latter is discarded.

3.4 Final analysis (Temporal+RFE, Temporal+GAs, DDWT-D1+RFE and WPT+LDB)

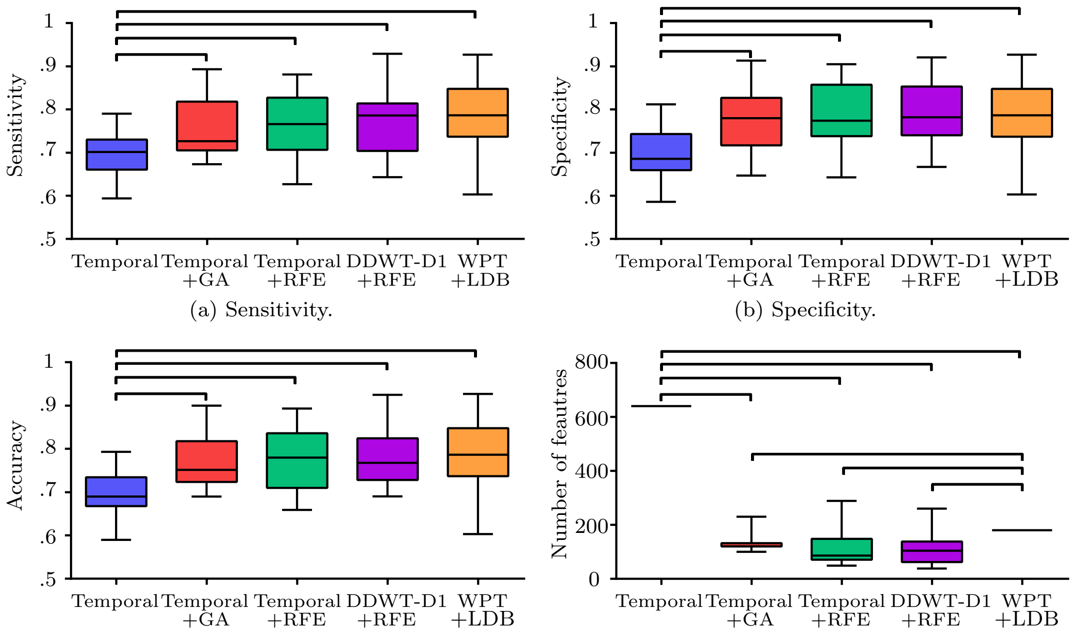
Performance indexes for the methods selected for the final analysis are shown in Fig. 2. No significant differences were found between the selected methods in terms of sensitivity (Friedman, $p < 0.615$), specificity (Friedman, $p < 0.376$) or accuracy (Friedman, $p < 0.457$). However, there were significant differences between in terms of number of features (Friedman, $p < 0.001$). In particular, Temporal+RFE, Temporal+GAs and DDWT-D1+RFE use significantly fewer features than WPT+LDB (Tukey, $p < 0.05$), without any other significant differences between methods.

4 Discussion

In general terms, it can be said that performance indexes were higher than 0.70 in most of the proposed strategies, and above 0.75 in more than half; which is an important achievement considering that with these values it would be possible to establish the communication of a person through a BCI and eventually control a device [18,19]. Taking into account potential hardware implementations of BCIs, strategies using temporal patterns and feature selection have advantages over strategies based on DDWT and WPT, as they are simpler and generate patterns with smaller dimensionality.

These results were contrasted with those reported by researchers who used the same UAM database. Among these are included Saavedra *et al.*, Lindig-León *et al.* and Kindermans *et al.*, that report the P300 classification rate [17,20,35]. In Table 2 we summarize the described results. In general terms, the results obtained in this study are similar or better (in the case of single-trial classification) than those reported in the literature, with the additional advantage of the reduced computational cost for the methods that use temporal patterns with feature selection (GA or RFE). Although Kindermans *et al* obtains a higher accuracy rate, they use patterns built using the average of 3 trials, which represents an advantage compared to single trial patterns.

Finally, in order to test the robustness of the proposed methodology, we selected a few representative methods from each strategy and we tested them using the BCI competition III database



(a) Sensitivity.

(b) Specificity.

(c) Accuracy.

(d) Number of features.

Fig. 2: Performance of best feature generation and selection methods for the UAM database.

Table 2: Comparison of the results obtained for other authors using the UAM database.

Authors	Accuracy	# of trials
This work	0.78	Single trial
Saavedra *et al* [35]	0.56	Single trial
Lindig *et al* [20]	0.75	2 trials
Kindermans *et al* [17]	0.89	3 trials

[26]. This database is widely used in the field, but it has the disadvantage that it contains recording from 2 subjects only. A total of 64 channels were acquired in positions of the 10-20 system and it includes a pre-specified train/test data partition. To closely resemble the conditions of the UAM database the signals, the patterns in this database were downsampled with a sampling frequency of 64 Hz and segmented to obtain 64-sample epochs. Only ten of the original 64 channels were used (Fz, C3, Cz, C4, P3, Pz, P4, PO7, PO8 and Oz). Due to the high class imbalance of this datasets the balanced accuracy was also computed. The detailed results are presented in Table 3. The average balanced accuracy for both subjects of evaluated methods were: temporal patterns achieved 0.67, temporal+GAs patterns achieved 0.72, DDWT-D1 patterns achieved 0.68 and WPT patterns achieved 0.67. The results corroborate the notion that all the proposed algorithms in this article provide acceptable and comparable performance.

5 Conclusions

In this study, we compared different orthogonal decompositions based on the Wavelet Transform for feature extraction, as well as with different filter, wrapper and embedded alternatives for feature selection. To the best of our knowledge, there are no previous experimental studies comparing feature

Table 3: Performances indexes obtained over the test patterns with the Wadsworth Center database for each of the subjects (A or B).

Patterns	Sensitivity		Specificity		Accuracy		Bal. Accuracy		# of features	
	A	B	A	B	A	B	A	B	A	B
Temporal	0.74	0.76	0.56	0.65	0.59	0.66	0.65	0.70	640	640
WPT + LDB	0.73	0.76	0.55	0.65	0.58	0.67	0.64	0.70	180	180
DDWT sin D1	0.76	0.77	0.55	0.65	0.58	0.67	0.65	0.71	320	320
Temporal + GAs	0.79	0.85	0.62	0.62	0.67	0.68	0.71	0.73	270	280

generation strategies using wavelet dictionaries combined with different dimensionality reduction techniques for BCIs based on single-trial classification of P300.

The use of strategies to improve the signal-to-noise ratio of single-trial EEG in the wavelet domain, either through smoothing of dyadic discrete wavelet transform or wavelet packet transform, proved to be valid alternatives to detect the P300 wave, since acceptable performance indexes were obtained. On the other hand, feature selection methods showed similar performances, both in wavelet and temporal patterns. However, the latter have an advantage when considering hardware implementation of BCIs given their reduced computational cost, since it is not necessary to transform the EEG signal.

Acknowledgements We thank the support of Universidad Nacional de Entre Ríos (UNER), Universidad Nacional del Litoral (UNL) and Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

References

1. Z. Amini, V. Abootalebi, and M. Sadeghi. Comparison of performance of different feature extraction methods in detection of P300. *Biocybernetics and Biomedical Engineering*, 33(1):3–20, 2013.
2. A. Bashashati, M. Fatourechi, R. Ward, and G. Birch. A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Journal of Neural Engineering*, 4(2):32–57, 2007.
3. B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller. Single-trial analysis and classification of ERP components — A tutorial. *NeuroImage*, 56(2):814–825, May 2011.
4. V. Bostanov. BCI competition 2003-data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. *IEEE Transactions on Biomedical engineering*, 51(6):1057–1061, 2004.
5. G. Chandrashekar and F. Sahin. A survey on feature selection methods.
6. R. Coifman and M. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.
7. B. Dal Seno, M. Matteucci, and L. Mainardi. A genetic algorithm for automatic feature extraction in P300 detection. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3145–3152, June 2008.
8. B. Dal Seno, M. Matteucci, and L. Mainardi. Online detection of P300 and error potentials in a BCI speller. *Computational intelligence and neuroscience*, 2010:11, 2010.
9. R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, and D. Tax. *PRTools4 - A Matlab Toolbox for Pattern Recognition*, 2004.
10. L. Farwell and E. Donchin. Talking off the top of your head: toward a metal prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical neurophysiology*, 70:510–523, 1988.
11. D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing, 1989.
12. I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature Extraction, Foundations and Applications. Series Studies in Fuzziness and Soft Computing.* Springer Verlag, 2006.
13. S. Herrmann, S. Rach, J. Vosskuhl, and D. Struber. Time–frequency analysis of event-related potentials: A brief tutorial. *Brain topography*, 27:438–450, 2014.
14. B. Jansen, A. Allam, P. Kota, K. Lachance, A. Osho, and K. Sundaresan. An exploratory study of factors affecting single trial p300 detection. *IEEE Transactions on Biomedical Engineering*, 51(6):975–978, 6 2004.
15. M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter. Bci competition 2003-data set iib: support vector machines for the p300 speller paradigm. *IEEE Transactions on Biomedical Engineering*, 51(6):1073–1076, 2004.

16. C.-Y. Kee, S. Ponnambalam, and C.-K. Loo. Multi-objective genetic algorithm as channel selection method for P300 and motor imagery data set. *Neurocomputing*, 161:120–131, 2015.
17. P.-J. Kindermans, H. Verschore, D. Verstraeten, and B. Schrauwen. A P300 BCI for the masses: Prior information enables instant unsupervised spelling. In *Advances in Neural Information Processing Systems*, pages 710–718, 2012.
18. A. Kubler, V. Mushahwar, L. Hochberg, and J. Donoghue. BCI meeting 2005-workshop on clinical issues and applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):131–134, June 2006.
19. K. Li, V. Narayan Raju, R. Sankar, Y. Arbel, and E. Donchin. *Advances and Challenges in Signal Analysis for Single Trial P300-BCI*, pages 87–94. Springer Berlin Heidelberg, 2011.
20. C. Lindig León and O. Yáñez Suárez. Optimized detection of the infrequent response in P300-based brain-computer interfaces. *Revista Mexicana de Ingeniería Biomédica*, 34(1):53–70, 2013.
21. F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 4(2), June 2007.
22. J. N. Mak, Y. Arbel, J. W. Minett, L. M. McCane, B. Yuksel, D. Ryan, D. Thompson, L. Bianchi, and D. Erdogmus. Optimizing the P300-based brain-computer interface: current status, limitations and future directions. *Journal of Neural Engineering*, 8(2):025003, 2011.
23. D. Milone, L. Rufiner, R. Acevedo, L. Di Persia, and H. Torres. *Introducción a las Señales y a los Sistemas Discretos.* EDUNER, 2006.
24. M. Mitchell. *An introduction to genetic algorithms 5ed.* MIT Press Cambridge, 1999.
25. M. R. Mowla, J. E. Huggins, and D. E. Thompson. Enhancing P300-BCI performance using latency estimation. *Brain-Computer Interfaces*, 4(3):137–145, 2017.
26. New York State Department of Health. BCI laboratory of the wadsworth center, Junio 2006.
27. M. Pacheco, Y. Atum, R. Acevedo, and L. Rufiner. Evaluation of different parents selection methods in a genetic algorithm wrapper for P300 BCI. In *XXV Congresso Brasileiro de Engenharia Biomédica (SBEB 2016)*, 2016.
28. B. Perseh and A. Sharafat. An efficient P300-based BCI using wavelet features and IBPSO-based channel selection. *Journal of Medical Signals and Sensors*, 2(3):128, 2012.
29. V. Peterson, R. Acevedo, H. L. Rufiner, and R. Spies. Local discriminant wavelet packet basis for signal classification in brain computer interface. In *VI Latin American Congress on Biomedical Engineering CLAIB 2014, Paraná, Argentina*, pages 584–587, Cham, 2015. Springer International Publishing.
30. V. Peterson, Y. Atum, F. Jauregui, I. Gareis, R. Acevedo, and L. Rufiner. Detección de potenciales evocados relacionados a eventos en interfaces cerebro-computadora mediante transformada wavelet. *Revista Ingeniería Biomédica*, 7(14):51–59, 2013.
31. T. W. Picton. The P300 wave of the human event-related potential. *Journal of Clinical Neurophysiology*, 9(4):456–479, 1992.
32. H. Qi, M. Xu, W. Li, D. Yuan, W. Zhu, X. An, D. Ming, B. Wan, and W. Wang. Feature selection study of P300 speller using support vector machine. In *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pages 1331–1334. IEEE, 2010.
33. A. Rakotomamonjy and V. Guigue. BCI Competition III: Dataset II- Ensemble of SVMs for BCI P300 Speller. *IEEE Transactions on Biomedical Engineering*, 55(3):1147–1154, Mar. 2008.
34. L. Rufiner. *Análisis y modelado digital de la voz. Técnicas recientes y aplicaciones.* Ediciones UNL, Colecci'on Ciencia y Técnica, 1a. ed. edition, 2006.
35. C. Saavedra and L. Bougrain. Wavelet-based semblance for P300 single-trial detection. In *International Conference on Bio-Inspired Systems and Signal Processing BIOSIGNAL 2013*, 2013.
36. N. Saito. Local feature extraction and its applications using a library of bases. In *Topics in Analysis and Its Applications: Selected Theses*, pages 269–451. World Scientific, 2000.
37. N. Saito and R. Coifman. Local discriminant bases and their applications. *Journal of Mathematical Imaging and Vision*, 5:337–358, 1995.
38. V. Samar. Wavelet analysis of neuroelectric waveforms: A conceptual tutorial. *Brain and Language*, 66:7–60, 1999.
39. G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw. BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004.
40. E. W. Sellers and E. Donchin. A P300-based brain–computer interface: initial tests by ALS patients. *Clinical neurophysiology*, 117(3):538–548, 2006.
41. H. Serby, E. Yom-Tov, and G. F. Inbar. An improved p300-based brain-computer interface. *IEEE Transactions on neural systems and rehabilitation engineering*, 13(1):89–98, 2005.
42. E. Smith and M. Delargy. Locked-in syndrome. *Bmj*, 330(7488):406–409, 2005.
43. A. Turnip, Haryadi, D. Soetraprawata, and D. Kusumandari. A comparison of extraction techniques for the rapid electroencephalogram-P300 signals. *Advanced Science Letters*, 20(1):80–85, 2014.
44. P. Wang and J. Shen. Research of P300 feature extraction algorithm based on wavelet transform and fisher distance. *International Journal of Education and Management Engineering*, 1(6):36–43, 2011.
45. A. Webb and A. Copsey. *Statistical pattern recognition 3ed.* Wiley Chichester, 2011.

46. J. Wolpaw, N. Birbaumer, W. Heetderks, D. McFarland, P. Peckham, G. Schalk, E. Donchin, L. Quatrano, C. Robinson, and T. Vaughan. Brain-computer interface technology: a review of the first international meeting. *IEEE transactions on rehabilitation engineering*, 8(2):164–173, June 2000.
47. J. Xie and Z. Qiu. The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern Recognition*, 40:557–562, 2007.
48. L. Zhuo, J. Zheng, F. Wang, X. Li, B. Ai, and J. Qian. A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:397–402, 2008.