A new multi-objective wrapper method for feature selection – Accuracy and stability analysis for BCI[☆],[☆☆]

Jesús González[a,*], Julio Ortega[a], Miguel Damas[a], Pedro Martín-Smith[a], John Q. Gan[b]

[a] *Department of Computer Architecture and Technology, CITIC, University of Granada, Granada, Spain*
[b] *School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK*

Abstract

Feature selection is an important step in building classifiers for high-dimensional data problems, such as EEG classification for BCI applications. This paper proposes a new wrapper method for feature selection, based on a multi-objective evolutionary algorithm, where the representation of the individuals or potential solutions, along with the breeding operators and objective functions, have been carefully designed to select a small subset of features that has good generalization capability, trying to avoid the over-fitting problems that wrapper methods usually suffer. A novel feature ranking procedure is also proposed in order to analyze the stability of the proposed wrapper method.

Four different classification schemes have been applied within the proposed wrapper method in order to evaluate its accuracy and stability for feature selection on a real motor imagery dataset. Experimental results show that the wrapper method presented in this paper is able to obtain very small subsets of features, which are quite stable and also achieve high classification accuracy, regardless of the classifiers used.

Keywords: BCI, EEG, motor imagery, feature selection, multi-objective problem, evolutionary algorithm, classification, stability, ensemble

1. Introduction

As the research on signal processing and machine learning advances, Brain Computer Interfaces (BCI) are becoming a real solution for people who suffer from some physical disabilities, such as tetraplegia, amyotrophic lateral sclerosis, spinal cord injury or limb amputation. BCI systems translate the brain signals into some control commands for an external device, allowing people with disabilities to interact with the world by simply thinking about it [1]. In fact, one of the most popular approaches to BCI applications is Motor Imagery (MI), which is based on the evidence that the imagination of motor actions has a similar brain response to the motor action itself. MI causes a series of amplifications and attenuations of short duration, the so-called event related desynchronization (ERD) and event related synchronization (ERS). Although the task of ERD/ERS analysis is quite complex, it is currently a hot topic because a BCI system properly trained with MI data of an impaired person could allow this person to control a robotic limb prosthesis [2, 3].

The first step to build a BCI system is the acquisition of the brain signals. Although there are many possibilities, such as Electrocorticography (ECoG), functional Magnetic Resonance Imaging (fMRI) or Magnetoencephalography (MEG), Electroencephalography (EEG) is the most studied technology, mainly because EEG equipment is inexpensive compared to other methods, and because it is portable and non-invasive, that is, no surgery is required to place the electrodes. This allows the recording of signals easily from any person, in order to obtain sufficient data to test the typical steps needed to implement the BCI application, which are pre-processing, feature extraction, feature selection, and finally, classification of the mental task. Once the mental task has been identified, control signals are generated to order the proper action to the external device.

The pre-processing step is mainly for denoising and extracting the components of interest from the original data. As ERD/ERS signals have random statistical characteristics and a great deal of variability, even between trials of the same subject, multiresolution analysis (MRA) has been widely used to overcome these issues [4–7]. Since ERD/ERS signals are weak and noisy, and occur at different locations of the cortex, at different instants within a trial, and in different frequency bands, the outcome of MRA usually suffers from the curse of dimensionality [8], as the number of samples is rather small because it is limited by the number of subjects and trials realized to obtain the data, but the samples are usually of very high dimensionality. Thus, the application of dimensionality reduction techniques is mandatory to enhance the discrimination capacity of the dataset.

[☆] This work was supported by project TIN2015-67020-P (Spanish "Ministerio de Economía y Competitividad" and European Regional Development Funds (ERDF))

[*] Corresponding author

Email addresses: jesusgonzalez@ugr.es (Jesús González), jortega@ugr.es (Julio Ortega), mdamas@ugr.es (Miguel Damas), pmartin@ugr.es (Pedro Martín-Smith), jqgan@essex.ac.uk (John Q. Gan)

The feature extraction step makes use of several techniques, such as statistical or spectral, in order to generate a set of features from the pre-processed data. Although this step may reduce the dimensionality of the problem to some extent, a feature selection process is also necessary to remove irrelevant, noisy or redundant features, which will improve the accuracy and interpretability of the final classifier, and also will decrease its computational complexity, allowing even the implementation of real-time systems.

Currently, there are three well known approaches to solve the feature selection problem: filter, wrapper and embedded methods, with wrapper methods being the most widely used [9]. Filtering techniques are based on the application of an evaluation criterion independent of the final classifier, such as measures of dependency, distance, graph-based learning or consistency over the input data, to make the selection [10], and although they have some advantages, like scalability or high speed, they usually result in a low accuracy in the final classification [11]. Some examples of filter techniques are [12–17] and [18], which has been designed to solve time-evolving feature selection problems.

On the contrary, wrapper techniques depend on a classification algorithm which is used to evaluate the candidate solutions (subsets of features) generated by a search algorithm, and thus are more computationally expensive. In spite of this drawback, they often provide better results, and should be applied whenever possible [19], although some care should be taken to prevent over-fitting, since the classifier used within the wrapper procedure evaluates solutions according to their performance for the training data. Lastly, embedded methods are applied during the classifier learning process to remove features based on the prediction errors of training data [20, 21].

The results of the wrapper approach to feature selection highly depend on the classification and searching algorithms applied. Regarding the search algorithm, as the search space grows exponentially when the number of features is increased, an exhaustive search for the best features subset is not practical when the number of available features is large. Thus, the use of metaheuristics [22] has been widely applied to solve this kind of problems. One of the first works using this approach was [23], where a wrapper method composed of a Genetic Algorithm (GA) [24] as searching heuristic and a Support Vector Machine (SVM) [25] as the classifier was presented to automatically select the most relevant features for an EEG-based BCI system. Later on, Khushaba *et al.* presented in [26] a filter/wrapper mixed approach where Ant Colony Optimizacion (ACO) [27] and a Linear Discriminant Analysis (LDA) [28] classifier were applied to make the feature selection. GAs were also used in [29], in combination with a Naïve Bayesian Classifier (NBC) [30], for the feature selection of an MI application, and in [31], where the performance of GAs and Simulated Annealing (SA) [32] was compared for a feature selection problem, both using a NBC as classifier.

The feature selection problem has also been treated as a Multi-Objective Optimization Problem (MOOP) [33–35], trying to maximize the classifier performance while the number of features is minimized, as larger feature sets could produce over-fitting and lower generalization capability. One possible approach is to use Particle Swarm Optimization (PSO) based algorithms. Although PSO was initially formulated for single-objective problems, it has been widely applied recently to solve feature selection problems [36], and in recent years, there have been several multi-objective implementations of PSO. For example, in [37] two multi-objective adaptations of PSO, one introducing the idea of non-dominated sorting, and the other applying concepts such as crowding, mutation and dominance to PSO, were proposed. Another possibility is suggested in [38] where a combination of LDA and a multi-objective hybrid real-binary Particle Swarm Optimization (MHPSO) algorithm is applied for EEG channel selection. However, Evolutionary Algorithms (EA) [39] are particularly well suited to these problems [40, 41], and have also been extensively applied to BCI applications. In [42] a many-objective approach, which optimizes five different objectives related to the accuracy and precision of the classifier in the wrapper procedure, is proposed. In [43] a wrapper method is proposed where the training accuracy and cross-validation error of a LDA classifier are used as objectives to be optimized by a parallel version of the NSGA-II [44] algorithm, one of the most popular implementations of the Multi-Objective Evolutionary Algorithm (MOEA) concept. The NSGA-II algorithm is also applied in [45] to optimize Deep Belief Networks (DBN) in an EEG classification problem, and in [46], where the accuracy of a wrapper procedure optimized by NSGA-II is compared with that obtained by GAAM [47], an adaptation of GAs to solve feature selection problems. Thus, this paper proposes the application of the NSGA-II algorithm to implement the search of potential solutions within the wrapper procedure. As NSGA-II highly depends on the representation of the potential solutions, breeding operators, and objective functions designed to guide the search, this paper proposes new alternatives for these key aspects of every MOEA, which have been carefully designed to achieve small subsets of features that will try to optimize both the classification accuracy and the generalization capability of the classifier, trying to avoid the over-fitting problems that wrapper procedures usually suffer.

On the other hand, the classification accuracy of the proposed wrapper procedure is not the only objective studied in this work. Although it is clear that the main purpose of a feature selection procedure is to identify a reduced subset of features that allow the classification of a high-dimensional dataset with a reasonably good accuracy, it has been pointed out recently that the robustness or stability of the selected feature subset is also important [48–51], since domain experts prefer feature selection algorithms that perform stably to small changes in the dataset. The stability concept is also valid for feature selection methods based on stochastic procedures, like the wrapper method proposed in this work, since different runs of the algorithm with different random seeds and/or initial populations may result in different feature selection results. Thus, a stability test of its results could help us evaluate whether the selected features are consistent from multiple runs of the proposed procedure.

One of the most widely used metrics to assess stability for feature selection methods is the Spearman correlation index [50, 51], which relies on full ranked lists of the set of features being analyzed by the wrapper method. Thus, another contribution of this paper is a novel procedure to obtain a full ranked list of features from the results obtained by each execution of the proposed wrapper method, in order to allow a stability analysis of the results obtained.

The rest of this paper is organized as follows: Section 2 describes the proposed multi-objective wrapper method. Section 3 summarizes how the stability of the results is assessed and introduces a novel procedure to obtain full ranked lists of features from the results of the proposed wrapper method to allow the stability assessment. Finally, Section 4 describes the experiments carried out and their results, and Section 5 concludes this work.

2. Proposed multi-objective evolutionary wrapper method

A wrapper procedure to reduce the number of features for a classification problem basically consists of a search algorithm, which explores the search space of all the possible subsets of features and applies a classification algorithm to evaluate some properties of candidate solutions, such as their accuracy, number of selected features, generalization capability, Kappa index [52], etc. As the wrapper procedure uses a training set to learn the most relevant features and a validation set to validate the feature selection, the search algorithm should provide a wide variety of solutions, in order to reject those that suffer from over-fitting (a high training accuracy but poor results with the test set) and keep those with more generalization capability. EAs [39] are quite adequate for this restriction, since they evolve a population of potential solutions, and thus, the probability of finding a solution with better test accuracy is higher. Moreover, EAs are particularly well suited to non-linear problems with extensive search spaces, such as the feature selection problem approached in this paper, where other optimization techniques are unable to find adequate solutions in a reasonable time.

Another advantage of using EAs as the search algorithm within the wrapper procedure is that they are well-suited to solve MOOPs, allowing the search of solutions that, having a high training accuracy, meet other objectives too, such as a small number of selected features, a high generalization capability, or even anatomical and functional relevance of EEG channels [53]. Although there are currently several well known implementations of the MOEA concept, such as MOEA/D [54], PAES [55], SPEA2 [56] or NSGA-II [44], since the aim of this paper is not to analyze the characteristics of the a specific MOEA but rather applying a MOEA as the search algorithm for the proposed wrapper method, NSGA-II has been used because it is quite well known and widely used.

Nevertheless, there are several aspects that must be adapted to complete the wrapper procedure, such as the most appropriate representation for the individuals, the breeding operators to perform the crossover and mutation of potential solutions for the problem, and the objectives to be optimized in order to select the most relevant features in a given dataset. The accuracy of the search algorithm and stability of the feature selection will highly depend on how these characteristics are mapped to the problem being solved, since NSGA-II is a generic multi-objective optimization technique. Fig. 1 shows the flowchart of the proposed wrapper method. The contributions of this paper have been highlighted, indicating also the section that describes each one of them. The remaining steps of the method are taken from the original NSGA-II algorithm.

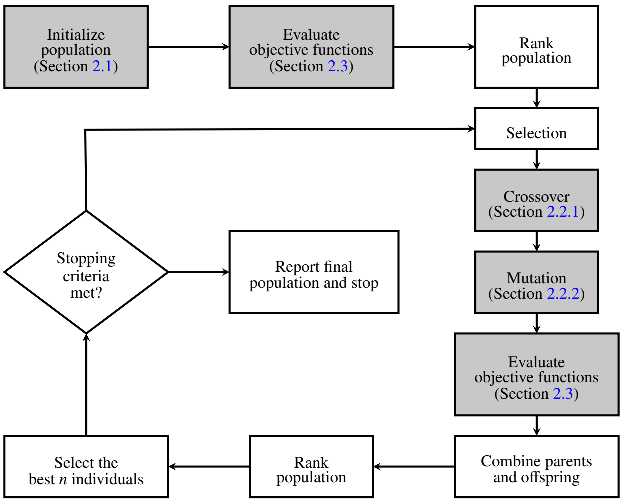
Initialize population (Section 2.1)
Evaluate objective functions (Section 2.3)
Rank population
Selection
Crossover (Section 2.2.1)
Mutation (Section 2.2.2)
Evaluate objective functions (Section 2.3)
Combine parents and offspring
Rank population
Select the best n individuals
Stopping criteria met?
Report final population and stop

Figure 1: Flowchart of the proposed wrapper method. The steps that are not highlighted are taken from the original NSGA-II algorithm

2.1. Individual representation

As the feature selection problem deals with the choice of the best subset of features, several representations are possible for the individuals. The most direct one is to use a binary vector $\boldsymbol{i}$ of size n, with n being the total number of available features, where a value of true for a given bit i_k means that the k-th feature is chosen and a value of false means that it is discarded. Thus, an individual would be coded as:

$$\boldsymbol{i} = [i_0, i_1, \ldots, i_{n-1}]^T \in \mathbb{R}^n : i_k \in \{0, 1\}, \quad k \in [0, n) \qquad (1)$$

Nevertheless, this representation has several drawbacks. First of all, each individual must store a value for all the features in the original data (selected or not), which implies a large memory size to store the whole population, especially since the potential solutions for the problem will be sparse vectors having mostly zero values, as the objective is to find a small subset of features to characterize the input patterns. Besides, the boolean type wastes a minimum of a byte in programming languages, when a boolean value could be stored only with one bit. In fact, it would be possible to use bit fields to use exactly one bit per feature, but this solution would imply a high computational overhead due to the shifts and boolean operations needed to extract and modify single bits inside memory words.

This problem was also faced in [47], which proposes another alternative representation. Given that it is possible to classify BCI patterns with a quite small number of features, it is not necessary to store thousands of boolean values (one value per feature) in every individual in the population. Instead it is more convenient to codify individuals as a small list of indexes of the selected features with a limited size $s < n$. However, this representation has also one disadvantage. Lists may contain repeated elements, while features can be selected once at most. Thus, this paper proposes the use of subsets of features to avoid this drawback. An individual will be represented as a subset I of indexes of the selected features with a limited size. That is:

$$I \subset \{x \in [0, n) \cap \mathbb{N}\}, \quad |I| \leq s \tag{2}$$

with $\mathbb{N}$ being the set of natural numbers. This representation will save a great amount of memory for the population storage and will also speed up the evaluation of individuals, as the loop that selects the features of the training data for the evaluation will process only a limited number of features. Thus, our wrapper procedure will use this representation for the individuals processed by the NSGA-II algorithm.

2.2. Breeding operators

The proposed individual representation implies special consideration needed in the design of custom breeding operators to generate the offspring. As individuals are coded as a subset of feature indexes with a limited size, the design of the breeding operators must take into account the total number of features (n) and the maximum number of selected features (s) to generate valid descendants.

2.2.1. Crossover operator

Given a couple of progenitors, P_i and P_j, the two offspring, O_i and O_j, will be generated as follows.

Let C_{ij} be the subset of features that have been selected by both P_i and P_j, that is, their common features:

$$C_{ij} = P_i \cap P_j \tag{3}$$

Let also R_{ij} be the remaining features in P_i and P_j, once the common features are removed:

$$R_{ij} = \left(P_i \setminus C_{ij}\right) \cup \left(P_j \setminus C_{ij}\right) \tag{4}$$

The offspring O_i and O_j will be of the form:

$$O_i = C_{ij} \cup R_i, \quad O_j = C_{ij} \cup R_j \tag{5}$$

provided that:

$$R_i \cup R_j = R_{ij} \quad \text{and} \quad R_i \cap R_j = \emptyset \tag{6}$$

$$|O_i| = |P_i| \quad \text{and} \quad |O_j| = |P_j| \tag{7}$$

All the common selected features in P_i and P_j will also be common in O_i and O_j, since $O_i \cap O_j = C_{ij}$. The remaining selected features R_{ij}, which are not common in P_i and P_j, will be randomly distributed between R_i and R_j (6) in a way that O_i and O_j will be of the same size of P_i and P_j respectively (7). This crossover procedure will always generate valid solutions that meet the constraints stated above.

2.2.2. Mutation operator

This operator affects each individual gene or selected feature within an individual separately. Given an individual I, a gene mutation probability of p_m, and a random variable X following a standard uniform distribution ($X \sim \mathcal{U}(0, 1)$), let M be defined as the random subset of features in I that will be mutated:

$$M = \{i \in I : X(i) \leq p_m\} \tag{8}$$

where $X(i)$ denotes the probability that feature i is mutated.

Once M is obtained, two possibilities exist to mutate all its elements. Each one of them could be modified or removed. Thus, M will be randomly split into two new subsets, M_s and M_r, the elements of M that will be substituted and those that will be removed respectively:

$$M_s = \{m \in M : X(m) \leq 0.5\}, \quad M_r = M \setminus M_s \tag{9}$$

The mutated individual I' will be obtained as:

$$I' = (I \setminus M) \cup N_s \cup N_a, \quad |I'| < s \tag{10}$$

where N_s is the subset of new features that will substitute those belonging M_s:

$$N_s \subset \{x \in [0, n) \cap \mathbb{N}\}, \ |N_s| = |M_s| \text{ and } N_s \cap I = \emptyset \tag{11}$$

and N_a is a subset of at most one new feature that will be added to I', in order to make possible the increase of features in I' respect to the original I.

$$N_a \subset \{x \in (I \setminus M) \setminus N_s\}, \quad |N_a| \in \{0, 1\} \tag{12}$$

2.3. Objective functions

The objective functions are responsible to guide the search towards optimum solutions, thus it is clear that the classification error of the wrapper procedure should be taken into account as one of the objectives. In our case, instead of the training accuracy, the training Kappa index [52] is preferred, as it not only takes into account the accuracy of the classifier, but also the per class error distribution. It is defined as follows:

$$\kappa(\mathscr{C}, D) = \frac{p_o(\mathscr{C}, D) - p_e(\mathscr{C}, D)}{1 - p_e(\mathscr{C}, D)} \tag{13}$$

where $p_o(\mathscr{C}, D)$ is the relative observed agreement between the classifier $\mathscr{C}$ and the labeled data in the dataset D, (identical to accuracy), and $p_e(\mathscr{C}, D)$ is the hypothetical probability of chance agreement between the classifier $\mathscr{C}$ and the labeled data in D.

Since datasets in BCI problems usually have a very high dimensionality and also a quite small number of samples, the wrapper method should count on a mechanism to avoid overfitting. Usually cross-validation is used to address this issue. However, applying cross-validation to evaluate all the solutions in the population in all the generations of a MOEA can be quite expensive in terms of computing time and computing power. So, this paper proposes a multi-objective approach to achieve similar results with much less computation. For each potential solution (subset of input features) to be evaluated, the original training dataset D is randomly split into two subsets, D_{tr} and D_{val} according to parameter p_{val}, which indicates the percentage of solutions used to validate the solution. Although D is split into two different random subsets for each individual evaluation, the division procedure always assures that a percentage p_{val} of samples of each class in D are included in D_{val}. Then, a classifier $\mathscr{C}$ is trained only with the samples belonging to D_{tr}, and later the following two objective functions o_1 and o_2 are calculated. Since the Kappa index may have values from 0 to 1, with 0 indicating no agreement at all between the classifier and the training data and 1 a total agreement, the objective functions have been defined as

$$o_1 = \kappa(\mathscr{C}, D_{tr}), \quad o_2 = \kappa(\mathscr{C}, D_{val}), \tag{14}$$

as the NSGA-II algorithm will try to maximize all the objective functions. While the first objective function guides the search towards solutions with high training accuracy, the second one prevents over-fitting, and since D is randomly split for the evaluation of each individual in the population, a sort of cross-validation, distributed over all the generations of the MOEA, is finally carried out, with the advantage that each solution is evaluated only twice (for D_{tr} and D_{val}) instead of five or ten times, which are typical values to apply cross-validation. Another advantage of this procedure is that the final solutions are not biased due to the way D is split into D_{tr} and D_{val}, since different subsets D_{tr} and D_{val} are randomly generated for each individual evaluation.

3. Proposed feature ranking procedure for stability assessment

Since the proposed feature selection method relies on a stochastic search algorithm, runs with distinct random seeds may obtain different results, as the initial population of the algorithm and the application of the breeding and selection operators depend on the random seed used to initialize the random number generator. Thus, the stability of the results should be assessed to validate the proposed method. To assess stability, it is common to perform a pairwise comparison of the results of each run with the remaining runs and average the assessment with respect to the number of comparisons [51]. That is:

$$\bar{S} = \frac{2}{l(l-1)} \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} S(\mathscr{R}_i, \mathscr{R}_j) \tag{15}$$

where $\bar{S}$ denotes the assessment score of the stability from the averaged pairwise comparisons, $S(\mathscr{R}_i, \mathscr{R}_j)$ is a stability score for the two full ranked lists $\mathscr{R}_i$ and $\mathscr{R}_j$, obtained by the *i-th* and *j-th* runs of the wrapper method, and l denotes the number of executions of the wrapper procedure, each one with a distinct random seed. One of the most widely used metrics for stability assessment is the Spearman correlation index [50, 51], which is applied as follows:

$$S(\mathscr{R}_i, \mathscr{R}_j) = 1 - \sum_{k=1}^{n} \frac{6(r_i^k - r_j^k)^2}{n(n^2-1)} \tag{16}$$

where n is the total number of features, and r_i^k denotes the rank of the *k-th* feature in the *i-th* ranked list.

Spearman correlation index ranges from -1 to 1, with 0 indicating no correlation at all, and 1 or -1 indicating a perfect positive or negative correlation, respectively. Thus, for feature selection stability assessment, the higher (positive) value of the Spearman index, the more consistency for the two ranked lists being compared, and therefore, the more stability of the wrapper procedure.

To apply the Spearman correlation index to our proposed multi-objective wrapper method it is necessary to obtain a full ranked list, from each execution, of all the features in the dataset. This paper proposes a novel procedure to generate a full ranked list of features from the Pareto front obtained after the execution of the wrapper procedure, which is based on the concept of the *relevance* of each feature. Let $\mathscr{P}_i$ be the Pareto front of the *i-th* run of the wrapper procedure:

$$\mathscr{P}_i = \{\Theta_i^j : j = 1, ..., m_i\} \tag{17}$$

where Θ_i^j denotes each one of the m_i Pareto-optimal solutions, which are subsets of features as defined in (2), found by the *i-th* run of the wrapper procedure. The relevance of each single feature f_k for the *i-th* execution of the wrapper method is defined as the probability of that feature appears in any of the m_i Pareto-optimal solutions:

$$R_i(f_k) = \frac{\sum_{j=1}^{m_i} \mu(f_k, \Theta_i^j)}{m_i} \tag{18}$$

where $\mu(f_k, \Theta_i^j)$ denotes the membership of feature f_k to the Pareto-optimal solution Θ_i^j, that is:

$$\mu(f_k, \Theta_i^j) = \begin{cases} 1 & \text{if} \quad f_k \in \Theta_i^j \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Once $R_i(f_k)$ has been calculated for all the n features in the dataset, they can be sorted by its relevance value and a rank can be assigned to each one of them according to their position within the list, obtaining the ranked list $\mathscr{R}_i$. In case of several features sharing the same relevance value, their rank is calculated as the mean of their positions in the list.

On the other hand, as proposed in [51], all the runs of a stochastic wrapper method can be used as an ensemble classifier to improve the accuracy or the stability of the results. In this case it is necessary to obtain a full ranked list combining the information of all the solutions found in the different Pareto-optimal fronts. Thus, instead of averaging the rank of each feature across the different runs, it is preferred to average the relevance of features:

$$\bar{R}(f_k) = \frac{\sum_{i=1}^{l} R_i(f_k)}{l} \tag{20}$$

Once the average relevance is obtained for each feature, the ranking can be obtained as detailed above.

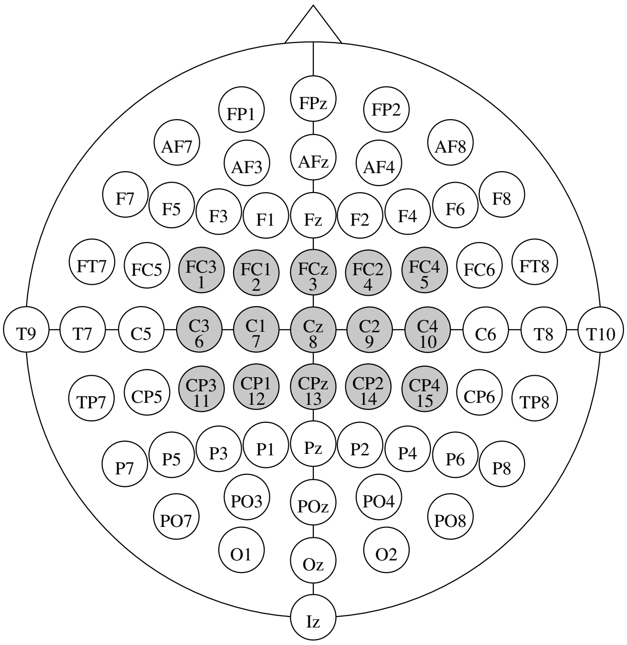

Figure 2: Numbering of the selected electrodes for the dataset

4. Experimentation

This section describes all the details related to the experimentation process carried out to evaluate the proposed wrapper method. First, the dataset used in all the experiments is described. Then, a baseline approach, which uses the same dataset, is briefly described, as the results obtained by the proposed wrapper procedure will be compared with those obtained by this baseline approach. A filter approach for multi-objective feature selection is briefly described too, since its results on the proposed dataset will be also compared with those obtained by the proposed wrapper method. After that, all the implementation details of the proposed wrapper method are presented.

Although a review of the most often used classifiers for BCI problems is presented in [57, 58], along with some recommendations about which classifier should be used for different kinds of BCI problems, it is necessary to test different classifiers for the dataset, in order to choose one with better results.

4.1. Dataset description

The MI dataset was recordered in the BCI laboratory at the University of Essex, UK, and contains three different classes: imaginary movements of right hand, left hand and feet. The BCI data were obtained applying the 10-20 international placement system [59], a standardized method to describe and apply the location of scalp electrodes in the context of an EEG test. The main motivation for this method is to ensure that a subject's experimental outcomes are able to be compiled, reproduced, and effectively analyzed and compared using the scientific method.

From the set of electrodes recommended by the 10-20 system, only 15 electrodes were used (FC3, FC1, FCz, FC2, FC4, C3, C1, Cz, C2, C4, CP3, CP1, CPz, CP2, CP4), as shown in Fig. 2, which cover the major area of the motor cortex. Samples were obtained from 12 healthy subjects (58% female, 50% naïve to BCI, with ages ranging from 24 to 50), recordered with a sampling frequency of 256 Hz during four different runs of 30 trials per class, producing a total of 120 trials per class for each subject. For experimental purposes, data collected from the first two runs has been dedicated to training and validating the wrapper procedures and the remaining has been used to test the accuracy of the classifiers with the resulting selected features. More details about this dataset can be found in [4].

The original data were filtered from 8 to 30 Hz in order to attenuate external noise and artifacts. Then, the filtered data were zero-centered, scaled, and segmented using a one-second window with an overlap of one fifth of a second, resulting in $n_s = 20$ segments per sample and per electrode or channel ($n_e = 15$ electrodes). Each segment was processed with the MRA approach described in [4], where a sequence of successive decomposition levels ($n_l = 6$) is applied to the signal, obtaining two sets of wavelet coefficients per level, one set for approximation (coefficients from the low-pass filter) and another for details (coefficients from the high-pass filter). With respect to the family of wavelets used, in [4] wavelet lifting (also known as second generation wavelets) is considered to build a set of wavelets adequate to cope with the temporal, spectral and spatial domains present in the MI signals analysis. The lifting scheme proposed in [4] is based on a graph representation of a motor imagery trial and builds the applied wavelets family more straightforward and with low resource consumption. Thus, the pre-procesed data were composed of a total of $n_e \times n_s \times n_l \times 2 = 3\,600$ sets of coefficients per sample (see Fig. 3), which implies a total of 151 200 coefficients to characterize each sample, since the number of coefficients in a set for the i^{th} level is 2^{8-i}.

After the pre-processing step, a quite simple feature extraction approach was applied in [4], which extracted one feature for each one of the 3 600 coefficient sets obtained by the MRA analysis. This feature was calculated by the second moment (variance) of the coefficient set and normalizing the value between 0 and 1. Therefore, the training and test datasets were finally composed of approximately 180 samples of $n = 3\,600$ features to characterize three different classes. Observing the great difference between the number of features and the number of samples in each dataset, it is clear that an efficient feature selection mechanism is required to avoid the existing curse of dimensionality problem and classify the samples properly.

4.2. Baseline classification approach

The baseline classification approach, described in [4], does not implement a feature selection step. Instead, a different LDA classifier is applied to each coefficient set of each level and each segment to avoid the curse of dimensionality problem. Thus, a total of $n_s \times n_l \times 2 = 240$ LDA classifiers, each with $n_e = 15$ inputs are applied to each input pattern, and the final classification is obtained with the majority voting of all the LDA outputs, as shown in Fig 4.

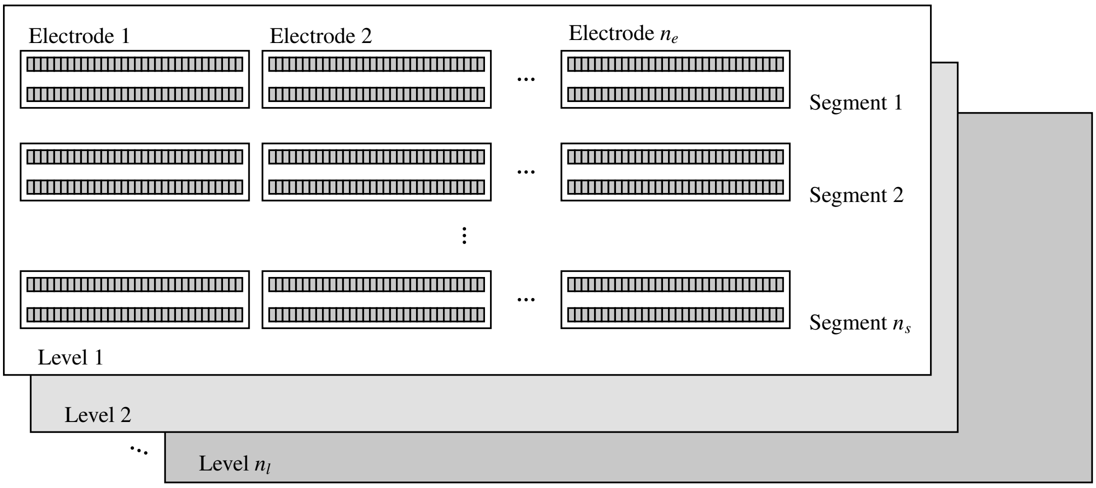

Figure 3: Results of the pre-processing step. For each imaginary movement recorded in the laboratory, the original signal is transformed into $n_e \times n_s \times n_l \times 2$ sets of coefficients. For each electrode, decomposition level and segment, two sets of coefficients are obtained, one for approximation and another for the details

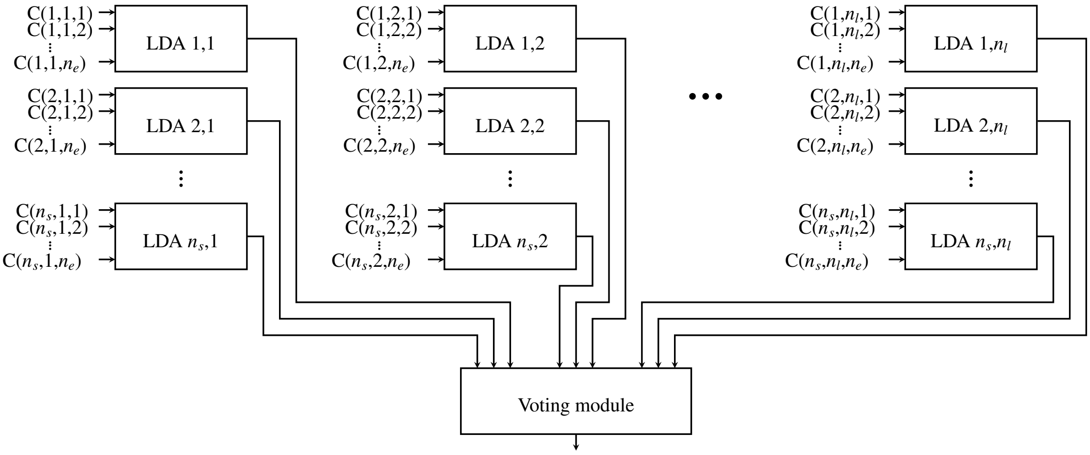

Figure 4: EEG classification with multiple LDA classifiers based on majority voting, with one LDA per segment and level

Table 1: Parameters for the wrapper method

Parameter	Value
Number of generations	200
Population size	3000
Total number of features (n)	3600
Maximum number of selected features (s)	30
Mutation probability (p_m)	0.01
Rate of training samples used for validation (p_{val})	0.33
Number of executions of the wrapper method (l)	50

4.3. *Supervised filter method for multi-objective feature selection*

Experiment has also been conducted on the same dataset using the FOPT* algorithm, a filter method for multi-objective feature selection in EEG classification [60]. In this case, the filter method is based on a set of label-aided utility functions that do not require the accuracy or the generalization of the classifier. The procedure defines a function for each label in the classification problem which will be used as an objective (or fitness) function by the multi-objective evolutionary algorithm NSGA-II.

4.4. *Sparse representation of data*

The datasets described in Section 4.1 were also used in [43], where several classification models were evaluated. Two of them, SR-LDA and SR-SVC, conducted the classification of samples in the datasets after transforming their input features into a sparse representation. To obtain this representation, a dictionary of 30 atoms per class was formerly generated with the K-SVD algorithm [61]. Then, the sparse representation of each sample was computed by means of the dictionary using the OMP algorithm [62], and finally, the sparse representation for each segment was summed up to obtain a histogram of sparse features that was used to train and test two different classifiers, LDA and SVC.

4.5. *Implementation details and parameterization of the proposed wrapper method*

The implementation of the NSGA-II algorithm has been taken from ECJ [63], a research Evolutionary Computation (EC) system written in *Java* and developed within the *Evolutionary Computation Laboratory* at the George Mason University, VA, USA. Moreover, the implementation of the direct LDA algorithm has been possible with the aid of *Apache commons Math* library [64], also written in *Java*. The rest of the code has been written by the authors of this work.

Since the wrapper method presented in this paper relies on the NSGA-II, which is an EA, there are some parameters that must be chosen to make it work. Table 1 shows the values used for all the experiments presented in this section. It is worth mentioning that no work on fine tuning these parameters has been considered, with the exception of the maximum number of selected features (s), as our aim here is to analyze if the proposed wrapper procedure provides comparable results to those of the baseline approach proposed in [4], but overcoming the curse of dimensionality by means of the feature selection based on the training data and the use of only one classifier. Thus, the values of this parameters have been obtained by means of several trial and error tests until good results were obtained. Regarding the maximum number of features, it has been fixed to 30 as in [60], where this parameter was also fixed for the FOPT* filter method after several trial and error tests in order to find a balance between the accuracy and the computation time of the filter method.

4.6. *Tested classifiers*

Linear Discriminant Analysis (LDA) [28] has been widely used for the problem of EEG motor imagery classification [4, 6, 43, 65–67]. Assuming a normal distribution and the same variance for all the input features, the objective of LDA is to project the input data into the $c-1$ dimensional space that maximizes the linear separability of the c classes in the dataset. If the data are linearly separable, hyperplanes can be used to separate the different classes, but if the data are not linearly separable, or the normality and homoscedasticity of the input data is not assured, the classification accuracy of LDA will be degraded. Nevertheless, since the input data is commonly preprocessed and scaled, it usually performs quite well with BCI data. However, when the number of input features is very large, as in the case of high-dimensional feature selection problems, LDA encounters several difficulties. The first one is the size of the matrices it has to handle to perform the projection. For a n-dimensional problem, it must process scatter matrices of size $n \times n$ and calculate complex operations on such matrices, such as inversion, multiplication, or eigenvalues computation. The second one is that these matrices are almost always singular, as the number of available patterns is much smaller than the number of features. Thus LDA, as proposed in [28], is usually not applicable in high-dimensional problems. Fortunately, [68] presents a direct method to perform an exact calculation of the LDA projection for high-dimensional problems, where all these difficulties are avoided.

k-Nearest Neighbors (KNN) [69] has also been applied to BCI systems [70–72]. This is a quite simple classifier that assigns a class to an unknown input pattern based on the dominant class among its k nearest neighbors [69]. With a sufficiently high value of k and enough training patterns, KNN can approximate any function, which enables it to produce nonlinear decision boundaries. However, KNN is very sensitive to the curse of dimensionality, which makes it fail with high-dimensional training data, although it performs quite well with low dimensional vectors [57]. Thus, as LDA is able to reduce the dimensionality of the input patterns and KNN has proven efficient in BCI applications with a small number of input features, the combination of LDA + KNN could achieve a good accuracy for feature selection problems.

Another widely used classifier is Naïve Bayesian Classifier (NBC), a probabilistic classifier based on the application of the Bayes' theorem with strong (naïve) independence assumptions among the input features [30]. It aims to assign an input pattern the class it belongs to with highest *a posteriori* probability, which is estimated from the class *a priori* probability and the

Table 2: Kappa values (avg ± std) for the training patterns of subjects 104, 107 and 110 of the University of Essex BCI data files, averaged over 50 executions

	Subject		
Classifier	104	107	110
Base	0.899	0.857	0.883
FOPT*	0.837 ± 0.019	0.829 ± 0.021	0.834 ± 0.018
KNN	0.957 ± 0.007	0.891 ± 0.013	0.898 ± 0.015
NBC	0.999 ± 0.001	0.983 ± 0.008	0.998 ± 0.006
LDA + KNN	0.890 ± 0.031	0.962 ± 0.009	0.937 ± 0.014
LDA + NBC	0.988 ± 0.007	0.967 ± 0.008	0.946 ± 0.009

Table 3: Kappa values (avg ± std) for the test patterns of subjects 104, 107 and 110 of the University of Essex BCI data files, averaged over 50 executions

	Subject		
Classifier	104	107	110
Base	0.564	0.631	0.648
FOPT*	0.661 ± 0.015	0.622 ± 0.029	0.575 ± 0.027
SR-LDA	0.607 ± 0.057	0.464 ± 0.048	0.443 ± 0.056
SR-SVC	0.614 ± 0.043	0.507 ± 0.046	0.456 ± 0.069
KNN	0.704 ± 0.031	0.550 ± 0.033	0.590 ± 0.031
NBC	0.642 ± 0.029	0.521 ± 0.030	0.515 ± 0.038
LDA + KNN	0.647 ± 0.053	0.584 ± 0.035	0.580 ± 0.039
LDA + NBC	0.677 ± 0.047	0.550 ± 0.028	0.574 ± 0.055

likelihood of each feature to belong to that class, which can be estimated with different kernel functions, such as Gaussian, multinomial, Bernouilli, etc. Its main application field is text classification, although it also has been applied to BCI and motor imagery applications [29, 31, 73–75].

As the LDA classifier assumes that classes are linearly separable, while KNN and NBC do not, the latter should perform better. However, LDA reduces the dimensionality of the dataset before performing the classification, and this could influence in the classification accuracy. Thus, it would be interesting to test whether NBC and KNN improve their classification accuracy when the dimensionality of the dataset has been previously reduced with LDA.

4.7. Results

NSGA-II, like all metaheuristics, performs a stochastic optimization, and thus, the global optimum is not guaranteed, specially in very complex problems like the one faced in this work. So, the wrapper method should be executed several times to have more possibilities to find a reasonably good solution. Another issue related to multi-objective problems is the way a final solution is selected from the Pareto front of each execution, since all the solutions could be equally valid. In this case, the whole training dataset is used to calculate an external ten-fold cross-validation error for all the solutions in the final Pareto front, and the one achieving the lowest error is chosen as the best solution. In case that several solutions achieve the same cross-validation error, the one selecting the smallest number of features is finally chosen as the best solution for the wrapper procedure. This procedure avoids the feature selection bias, as suggested in [76, 77].

Tables 2 and 3 present the average and standard deviation, over 50 executions, of the Kappa values obtained using the subsets of features selected by the proposed wrapper method for the training and test datasets, respectively. These results are also compared with the baseline (Base), the FOPT* filter, and the SR-LDA and SR-SCV approaches. Only subjects 104, 107 and 110 of the Essex BCI dataset have been evaluated, as these subjects are the ones which achieved better classification accuracy in [4]. For each subject, four different alternatives have been tested as the classifier used in the wrapper method: KNN and NBC applied directly to the features selected by the wrapper method, and both classifiers applied to the LDA projection of these selected features. For the KNN classifier, the value of parameter k has been fixed to the odd number closest to the squared root of the number of samples in the dataset, as it is a widely used heuristic to fix this parameter.

Table 4: p-values obtained from the Kolmogorov-Smirnov statistical tests performed to the results presented in Table 3

	Subject		
Classifier	104	107	110
FOPT*	0.447	0.298	0.921
SR-LDA	0.548	0.973	0.217
SR-SVC	0.062	0.072	0.349
KNN	0.365	0.726	0.701
NBC	0.678	0.798	0.626
LDA + KNN	0.622	0.738	0.219
LDA + NBC	0.539	0.374	0.772

Table 5: p-values obtained from the Bartlett and Kruskal-Wallis tests performed to the results presented in Table 3

	Subject		
Test	104	107	110
Bartlett	0.000	0.000	0.000
Kruskal-Wallis	0.000	0.000	0.000

As can be observed in Table 3, which shows the Kappa values for the test patterns, the proposed approach outperformed the baseline and filter approaches for subject 104, independently of the classifier. Regarding subject 107, the proposed wrapper procedure using LDA + KNN as classifier seems to be the best option, as its results are closer to those obtained by the baseline approach and the FOPT* method, and finally for subject 110 all the classifiers tested within the wrapper method provide results similar to those obtained by FOPT*. Nevertheless, all the results are quite similar for each subject, so statistical analysis is needed in order to check if there are significant differences among the results [78].

First of all, a Kolmogorov-Smirnov test has been applied to the obtained results in order to check their normality. As can be seen in Table 4, the p-values obtained for all the experiments are much larger than the typical significance level of 0.05, which means that the test has failed to reject the null hypothesis that the results follow a normal distribution, that is, all the results follow a normal distribution. Then, a Bartlett test has been applied to test the homoscedasticity of the results obtained by the different feature selection alternatives for each subject. In this case, Table 5 shows that all the p-values obtained are zero, revealing that the null hypothesis that the results of the differ-

Table 6: p-values obtained from multiple pairwise comparison of the different feature selection alternatives. Values lower than a significance level of 0.05 have been highlighted

		Subject		
Classifier 1	Classifier 2	104	107	110
Base	FOPT*	**0.000**	1.000	**0.007**
Base	SR-LDA	0.505	**0.000**	**0.000**
Base	SR-SVC	0.516	**0.000**	**0.000**
Base	KNN	**0.000**	**0.000**	**0.027**
Base	LDA + KNN	**0.000**	0.217	**0.004**
Base	NBC	**0.001**	**0.000**	**0.000**
Base	LDA + NBC	**0.000**	**0.000**	**0.001**
FOPT*	SR-LDA	0.137	**0.000**	**0.000**
FOPT*	SR-SVC	0.111	**0.000**	**0.002**
FOPT*	KNN	0.097	**0.000**	0.913
FOPT*	LDA + KNN	0.981	0.477	0.998
FOPT*	NBC	0.848	**0.000**	**0.035**
FOPT*	LDA + NBC	0.995	**0.000**	1.000
SR-LDA	SR-SCV	1.000	0.831	1.000
SR-LDA	KNN	**0.000**	**0.000**	**0.000**
SR-LDA	LDA + KNN	0.268	**0.000**	**0.000**
SR-LDA	NBC	0.574	0.306	0.347
SR-LDA	LDA + NBC	**0.001**	**0.000**	**0.000**
SR-SVC	KNN	**0.000**	**0.031**	**0.000**
SR-SVC	LDA + KNN	0.214	**0.000**	**0.000**
SR-SVC	NBC	0.505	1.000	0.644
SR-SVC	LDA + NBC	**0.001**	0.055	**0.000**
KNN	LDA + KNN	**0.000**	**0.011**	0.990
KNN	NBC	**0.000**	**0.005**	**0.000**
KNN	LDA + NBC	0.059	1.000	0.859
LDA + KNN	NBC	0.997	**0.000**	**0.000**
LDA + KNN	LDA + NBC	0.180	**0.004**	0.999
NBC	LDA + NBC	**0.028**	**0.012**	**0.000**

ent feature selection methods have homogeneous variances is rejected. Thus, given that results present heteroscedasticity, a Kruskal-Wallis test should be applied to check if there exist significant differences among the proposed feature selection methods. Table 5 also shows a p-value of 0 for the Kruskal-Wallis test for all the subjects, which means that the results are statistically different.

Table 6 presents a multiple pairwise comparison of the different feature selection alternatives where those p-values lower than a significance level of 0.05 have been highlighted. A marked value for a pair of feature selection methods indicates that their results are statistically different. Thus, relating p-values of Table 6 with the results presented in Table 3, all the analyzed methods can be compared for each subject, as shown in Table 7. Thus, it can be stated that all the proposed feature selection alternatives produce results similar to those produced by FOPT* and better results than the baseline and SR-LDA and SR-SVC approaches for subject 104, with KNN and LDA + NBC being the best ones while NBC and LDA +KNN preforming slightly worse. However, for subject 107 the baseline, FOPT*, along with LDA + KNN, and for subject 110, FOPT*, KNN, LDA + KNN and LDA + NBC achieve comparable results, better than NBC and worse than the baseline approach. Regarding the proposed four classification alternatives,

Table 7: Comparison of all the analyzed methods for the different subjects in the University of Essex BCI data files. Relation $a \prec b$ means that method a has achieved statistically significant better results than method b

Subject	Achieved performance				
104	KNN LDA + NBC	$\leq$	LDA + KNN NBC FOPT$*$	$\prec$ Base $\prec$	SR-LDA SR-SVC
107	Base FOPT$*$ LDA + KNN	$\prec$	KNN LDA + NBC	$\prec$	NBC SR-LDA SR-SVC
110	Base	$\prec$	FOPT$*$ KNN LDA + KNN LDA + NBC	$\prec$	NBC SR-LDA SR-SVC

Table 8: Number of features (avg ± std) selected for subjects 104, 107 and 110 of the University of Essex BCI data files averaged over 50 executions

	Subject		
Classifier	104	107	110
Base	3 600	3 600	3 600
FOPT*	29.770 ± 0.342	29.763 ± 0.224	29.899 ± 0.122
KNN	28.260 ± 1.209	28.840 ± 0.866	29.080 ± 0.829
NBC	27.220 ± 1.112	29.360 ± 0.921	29.380 ± 0.725
LDA + KNN	28.760 ± 1.117	29.850 ± 0.670	29.680 ± 0.695
LDA + NBC	28.480 ± 1.249	29.860 ± 0.756	29.720 ± 0.757

Table 9: Stability scores achieved by the proposed wrapper method using different classifiers for subjects 104, 107 and 110 of the University of Essex BCI data files, averaged over 50 executions

	Subject		
Classifier	104	107	110
KNN	0.948	0.959	0.963
NBC	0.679	0.928	0.793
LDA + KNN	0.694	0.834	0.920
LDA + NBC	0.721	0.859	0.879

Table 10: Execution times in seconds (avg ± std) of the proposed wrapper method using different classifiers for the test patterns of subjects 104, 107 and 110 of the University of Essex BCI data files, averaged over 50 executions

	Subject		
Classifier	104	107	110
KNN	439.054 ± 4.916	448.368 ± 3.496	432.984 ± 4.511
NBC	202.111 ± 6.245	190.421 ± 3.063	186.757 ± 4.919
LDA + KNN	235.715 ± 16.479	213.922 ± 11.835	192.190 ± 3.329
LDA + NBC	239.475 ± 17.524	207.530 ± 8.633	190.392 ± 4.677

NBC has always performed worse than the others, whereas KNN, LDA + KNN and LDA + NBC have achieved similar results, with KNN being better for subject 104 and LDA + KNN obtaining better results for subject 107.

On the other hand, the proposed wrapper method also reduce significantly the number of features needed to classify the input patterns, independently of the classifier algorithm applied. As Table 8 shows, while the baseline approach uses the 3 600 input features of the dataset, the proposed wrapper procedure is able to find subsets of up to 30 features, as FOPT* does, that can be used to achieve comparable classification results, as has been discussed above.

Regarding the stability of the proposed wrapper method, Table 9 shows the stability scores obtained for the different classifiers tested, according to (15) and the full ranked lists of features obtained from the different executions of our wrapper procedure based on our feature relevance criterion (18). As can be seen, most stability scores are close to one, which means a perfect correlation, that is, a quite stable method, specifically for KNN, independently of the subject. Thus, given that KNN has also been one of the proposed methods that has achieved the best results, it seems that KNN is a good option to be considered for this kind of problems.

With respect to the execution time of the proposed wrapper method, it heavily depends on the classifier used within it. As Table 10 shows, NBC obtains the best times while KNN the worst ones. The computation time of KNN also depends on the value of k, which has been fixed heuristically to the odd number closest to the squared root of the number of samples in the dataset. Higher or lower values for this parameter will increase or decrease the execution time respectively. On the other hand, as the application of LDA reduces the dimensionality of the input data drastically, both LDA + KNN and LDA + NBC achieve similar execution times, slightly superior to NBC. The running times also depend on the execution platform. In this case, all the executions of the wrapper method have been performed by means of a master-slave version of NSGA-II running in a cluster composed of 16 nodes, each one containing 2 *Intel Xeon E5520* CPUs running at 2.27 GHz.

Finally, as proposed in [51], the 50 different runs of the wrapper method with a fixed classifier for each subject could be treated as an ensemble classifier to improve the stability of the results. In this case, the relevance of each feature has been obtained as the average of the relevance values for this feature in the different runs of the wrapper procedure for each subject, as explained in Section 3. Table 11 shows the Kappa value obtained for the test patterns and the number of features used when

Table 11: Kappa value and number of selected features for the test patterns using only the features with an average relevance value greater than or equal to a given threshold for the ensemble classifiers. For each classifier and subject, values comparable to those presented in Table 3 (within the range mean ± std) or better have been highlighted

		Subject					
		104		107		110	
Th.	Classifier	Kappa	(N)	Kappa	(N)	Kappa	(N)
1	KNN	0.048	(1)	0.016	(1)	0.212	(1)
	NBC	0.355	(1)	0.175	(1)	0.076	(1)
	LDA + KNN	0.048	(1)	0.000	(1)	0.212	(1)
	LDA + NBC	0.000	(1)	0.147	(1)	0.222	(1)
0.9	KNN	0.601	(8)	0.500	(6)	**0.564**	**(7)**
	NBC	0.387	(8)	0.416	(6)	0.213	(7)
	LDA + KNN	0.246	(5)	0.358	(3)	0.439	(4)
	LDA + NBC	0.384	(5)	0.266	(2)	0.413	(5)
0.8	KNN	**0.705**	**(12)**	**0.542**	**(10)**	**0.564**	**(11)**
	NBC	0.480	(10)	0.483	(9)	0.313	(8)
	LDA + KNN	0.479	(8)	0.432	(6)	0.497	(6)
	LDA + NBC	0.485	(8)	0.500	(5)	0.413	(8)
0.7	KNN	**0.688**	**(15)**	0.508	(11)	**0.564**	**(14)**
	NBC	0.455	(12)	0.474	(12)	0.372	(11)
	LDA + KNN	0.504	(9)	0.491	(7)	0.447	(9)
	LDA + NBC	0.555	(9)	0.499	(8)	0.413	(9)
0.6	KNN	**0.712**	**(18)**	**0.542**	**(13)**	0.556	(18)
	NBC	**0.648**	**(16)**	0.490	(18)	0.456	(15)
	LDA + KNN	**0.671**	**(16)**	0.482	(15)	**0.548**	**(13)**
	LDA + NBC	0.588	(12)	0.491	(11)	0.489	(11)
0.5	KNN	0.670	(20)	**0.533**	**(16)**	**0.581**	**(20)**
	NBC	**0.640**	**(20)**	**0.532**	**(21)**	**0.539**	**(20)**
	LDA + KNN	**0.662**	**(19)**	0.524	(18)	**0.623**	**(15)**
	LDA + NBC	0.581	(15)	0.482	(13)	**0.539**	**(18)**
0.4	KNN	0.652	(26)	**0.525**	**(21)**	**0.606**	**(23)**
	NBC	**0.657**	**(23)**	**0.499**	**(24)**	**0.497**	**(25)**
	LDA + KNN	**0.730**	**(22)**	**0.566**	**(23)**	**0.606**	**(19)**
	LDA + NBC	0.623	(20)	0.466	(19)	**0.589**	**(26)**
0.3	KNN	**0.696**	**(30)**	**0.567**	**(28)**	**0.589**	**(26)**
	NBC	**0.648**	**(27)**	**0.499**	**(32)**	**0.539**	**(30)**
	LDA + KNN	**0.653**	**(27)**	**0.583**	**(31)**	0.531	(30)
	LDA + NBC	**0.673**	**(26)**	0.516	(29)	**0.573**	**(30)**

Table 12: Spearman correlation index between the different classifiers according to their average ranked list of selected features

		Subject		
Classifier 1	Classifier 2	104	107	110
KNN	LDA + KNN	0.314	0.475	0.426
KNN	NBC	0.246	0.378	0.318
KNN	LDA + NBC	0.284	0.480	0.426
LDA + KNN	NBC	0.229	0.201	0.259
LDA + KNN	LDA + NBC	0.474	0.562	0.515
NBC	LDA + NBC	0.231	0.221	0.241

the datasets are classified only with a subset of features whose relevance is greater than or equal to a given relevance threshold. Only results with a relevance value greater than 0.3 are displayed, since subsets with lower relevance values are composed of a relative high number of features (compared with the number of samples in the datasets). Those feature subsets that are comparable or better than those presented in Table 3 have been highlighted. As can be seen, results obtained by the ensemble are comparable with those obtained by each separate execution of the wrapper algorithm. Specifically, in the last row of Table 11, which shows the subsets of features evaluated with a relevance close to 0.3, all the subsets have a number of features that is near to 30 and almost all of them have resulted in a Kappa value comparable or even better than the values presented in Table 3. These results are completely coherent with the stability indexes presented in Table 9 and demonstrate that the wrapper method presented in this paper is able to find quite stable subsets of features regardless of the classifier, as the most relevant features for each subject tend to appear in the Pareto front of all the executions of the wrapper algorithm, obtaining a rather high relevance index.

Lastly, Table 12 shows the pairwise Spearman correlation index between the different classifiers applied within the proposed wrapper procedure. In this case, since all the executions of the wrapper method for each different classifier have been treated as an ensemble, average relevances (20) have been used to obtain the full ranked lists and the average stability scores. As can be seen, unfortunately there is not much correlation across the classifiers in general, which means that each classifier selects different subsets of features for each one of the datasets.

5. Conclusions

This paper has presented a multi-objective evolutionary wrapper method specifically designed for feature selection from high-dimensional motor imagery data. Although it is based on NSGA-II, a commonly used MOEA, it incorporates a new way of encoding the individuals in order to save memory and speedup their evaluation. It also uses new breeding operators designed for the proposed individual representation and two objective functions that guide the search towards subsets of features that maximize the training Kappa index while also avoid over-fitting. It is worth mentioning that these two objective functions have been designed to avoid applying a cross-validation evaluation for each individual of the population, which would demand much more computation time.

The validity of the proposed algorithm has been tested applying four different classifiers on an EEG classification problem. As Section 4.7 shows, the proposed wrapper method is able to improve the baseline approach and achieve results similar to those achieved by FOPT* for subject 104 of the University of Essex BCI data files. It also achieves results comparable to FOPT* for subjects 107 and 110. For these two subjects the baseline approach obtains the best Kappa values, but it uses the whole 3 600 features, whereas FOPT* and the proposed wrapper method impose a limit of 30 selected features at most.

Regarding the four classifiers tested within the proposed wrapper method, KNN obtains the best Kappa values for subjects 104 and 110 on the test dataset. Besides, it also presents more stability than the other classifiers for the three subjects, which provide relatively different full ranked lists of selected features for each subject. Thus, it seems to be a quite good choice for motor imagery feature selection problems, although its execution time, which depends on the value fixed for *k*, can be notably higher. On the other hand, the observed lack of stability across different classifiers is an interesting issue that is worth being analyzed in depth in our future research.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments and suggestions that have helped improve the quality of the paper.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, Brain-computer interfaces for communication and control, Clinical Neurophysiology 113 (6) (2002) 767–791, doi:10.1016/S1388-2457(02)00057-3.

[2] P. L. Jackson, M. F. Lafleur, F. Malouin, C. Richards, J. Doyon, Potential role of mental practice using motor imagery in neurologic rehabilitation, Archives of Physical Medicine and Rehabilitation 82 (8) (2001) 1133–1141, doi:10.1053/apmr.2001.24286.

[3] A. Zimmermann-Schlatter, C. Schuster, M. A. Puhan, E. Siekierka, J. Steurer, Efficacy of motor imagery in post-stroke rehabilitation: a systematic review, Journal of NeuroEngineering and Rehabilitation 5 (8) (2008) doi:10.1186/1743-0003-5-8.

[4] J. Asensio-Cubero, J. Q. Gan, R. Palaniappan, Multiresolution analysis over simple graphs for brain computer interfaces, Journal of Neural Engineering 10 (4) (2013) 046014, doi:10.1088/1741-2560/10/4/046014.

[5] P. Saidi, G. K. Atia, A. Paris, A. Vosoughi, Motor imagery classification using multiresolution analysis and sparse representation of eeg signals, in: Proceedings of the 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, Orlando, FL, USA, 2015, pp. 815–819, doi:10.1109/GlobalSIP.2015.7418310.

[6] A. Celecia, R. González, M. Vellasco, Feature selection methods applied to motor imagery task classification, in: Proceedings of the 2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI), IEEE, Cartagena, Colombia, 2016, doi:10.1109/LA-CCI.2016.7885731.

[7] J. Luo, Z. Feng, J. Zhang, N. Lu, Dynamic frequency feature selection based approach for classification of motor imageries, Computers in Biology and Medicine 75 (1) (2016) 45–53, doi:10.1016/j.compbiomed.2016.03.004.

[8] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, Princeton, NJ, USA, 1961.

[9] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517, doi:10.1093/bioinformatics/btm344.

[10] X. Zhu, H.-I. Suk, L. Wang, S.-W. Lee, D.Shen, Alzheimer's Disease Neuroimaging Initiative, A novel relational regularization feature selection method for joint regression and classification in ad diagnosis, Medical Image Analysis 38 (2017) 205–214, doi:10.1016/j.media.2015.10.008.

[11] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502, doi:10.1109/TKDE.2005.66.

[12] F. Nie, S. Xiang, Y. Jia, C. Zhang, S. Yan, Trace ratio criterion for feature selection, in: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08), Vol. 2, AAAI Press, Chicago, IL, USA, 2008, pp. 671–676.

[13] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization, in: Proceedings of the 23rd International Conference on Neural Information Processing Systems (NIPS'10), Vol. 2, Curran Associates Inc., Vancouver, British Columbia, Canada, 2010, pp. 1813–1821.

[14] Y. Liu, F. Nie, J. Wu, L. Chen, Efficient semi-supervised feature selection with noise insensitive trace ratio criterion, Neurocomputing 105 (2013) 12–18, doi:10.1016/j.neucom.2012.05.031.

[15] F. Nie, W. Zhu, X. Li, Unsupervised feature selection with structured graph optimization, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16), AAAI Press, Phoenix, AZ, USA, 2016, pp. 1302–1308.

[16] W. He, X. Cheng, R. Hu, Y. Zhu, G. Wen, Feature self-representation based hypergraph unsupervised feature selection via low-rank representation, Neurocomputing 253 (2017) 127–134, doi:10.1016/j.neucom.2016.10.087.

[17] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, S. Zhang, Graph self-representation method for unsupervised feature selection, Neurocomputing 220 (2017) 130–137, doi:10.1016/j.neucom.2016.05.081.

[18] J. Li, X. Hu, L. J. nad H. Liu, Toward time-evolving feature selection on dynamic networks, in: Proceedings of the 16th IEEE International Conference on Data Mining (ICDM'2016), IEEE, Barcelona, Spain, 2016, pp. 1003–1008, doi:10.1109/ICDM.2016.56.

[19] P. Pudil, P. Somol, Identifying the most informative variables for decision-making problems – a survey of recent approaches and accompanying problems, Acta Oeconomica Pragensia 2008 (4) (2008) 37–55, doi:10.18267/j.aop.131.

[20] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: A framework for unsupervised feature selection, IEEE Transactions on Cybernetics 44 (6) (2014) 793–804, doi:10.1109/TCYB.2013.2272642.

[21] X. Zhu, S. Zhang, R. Hu, Y. Zhu, J. Song, Local and global structure preservation for robust unsupervised spectral feature selection, IEEE Transactions on Knowledge and Data Engineering 30 (3) (2018) 517–529, doi:10.1109/TKDE.2017.2763618.

[22] K. Sörensen, Metaheuristics – the metaphor exposed, International Transactions in Operational Research 22 (1) (2013) 3–18, doi:10.1111/itor.12001.

[23] M. Schroder, M. Bogdan, T. Hinterberger, N. Birbaumer, Automated eeg feature selection for brain computer interfaces, in: Proceedings of the 1st International IEEE/EMBS Conference on Neural Engineering, IEEE, Capri, Italy, 2003, pp. 626–629, doi:10.1109/CNE.2003.1196906.

[24] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Professional, Reading, MA, USA, 1989.

[25] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297, doi:10.1007/BF00994018.

[26] R. N. Khushaba, A. AlSukker, A. Al-Ani, A. Al-Jumaily, Intelligent artificial ants based feature extraction from wavelet packet coefficients for biomedical signal classification, in: Proceedings of the 3rd IEEE International Symposium on Control, Communications and Signal Processing (ISCCSP 2008), IEEE, St. Julians, Malta, 2008, pp. 1366–1371, doi:10.1109/ISCCSP.2008.4537439.

[27] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, Cambridge, MA, USA, 2004.

[28] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188, doi:10.1111/j.1469-1809.1936.tb02137.x.

[29] R. Corralejo, R. Hornero, D. Álvarez, Feature selection using a genetic algorithm in a motor imagery-based brain computer interface, in: Proceedings of the 33rd Annual International Conference of the IEEE Engineering-in-Medicine-and-Biology-Society (EMBS), IEEE, Boston, MA, USA, 2011, pp. 7703–7706, doi:10.1109/IEMBS.2011.6091898.

[30] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1973.

[31] S. Basterrech, P. Bobrov, A. Frolov, D. Húsek, Nature-inspired algorithms for selecting eeg sources for motor imagery based bci, in: Proceedings of the 14th International Conference on Artificial Intelligence and Soft Computing (ICAISC 2015), Part II, Lecture Notes in Computer Science, Springer, Zakopane, Poland, 2015, pp. 79–90, doi:10.1007/978-3-319-19369-4_8.

[32] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680, doi:10.1126/science.220.4598.671.

[33] A. Guillén, H. Pomares, J. González, I. Rojas, O. Valenzuela, B. Prieto, Parallel multiobjective memetic rbfnns design and feature selection for function approximation problems, Neurocomputing 72 (16–18) (2009) 3541–3555, doi:10.1016/j.neucom.2008.12.037.

[34] I. Vatolkin, M. Preuss, G. Rudolph, M. Eichhoff, C. Weihs, Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures, Soft Computing 16 (12) (2012) 2027–2047, doi:10.1007/s00500-012-0874-9.

[35] L. D. Vignolo, D. H. Milone, J. Scharcanski, Feature selection for face recognition based on multi-objective evolutionary wrappers, Expert Systems with Applications 40 (13) (2013) 5077–5084, doi:10.1016/j.eswa.2013.03.032.

[36] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (4) (2016) 606–626, doi:10.1109/TEVC.2015.2504420.

[37] B. Xue, M. Zhang, W. N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Transactions on Cybernetics 43 (6) (2013) 1656–1671, doi:10.1109/TSMCB.2012.2227469.

[38] A. Gonzalez, I. Nambu, H. Hokari, Y. Wada, Eeg channel selection using particle swarm optimization for the classification of auditory event-related potentials, Scientific World Journal (2014) 350270, doi:10.1155/2014/350270.

[39] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd Edition, Springer, Berlin, Germany, 1998.

[40] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. Coello Coello, A survey of multiobjective evolutionary algorithms for data mining: Part i, IEEE Transactions on Evolutionary Computation 18 (1) (2014) 4–19, doi:10.1109/TEVC.2013.2290086.

[41] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. Coello Coello, A survey of multiobjective evolutionary algorithms for data mining: Part ii, IEEE Transactions on Evolutionary Computation 18 (1) (2014) 20–35, doi:10.1109/TEVC.2013.2290082.

[42] M. Pal, S. Bandyopadhyay, Many-objective feature selection for motor imagery eeg signals using differential evolution and support vector machine, in: Proceedings of the 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), IEEE, Durgapur, India, 2016, doi:10.1109/MicroCom.2016.7522574.

[43] J. Ortega, J. Asensio-Cubero, J. Q. Gan, A. Ortiz, Classification of motor imagery tasks for bci with multiresolution analysis and multiobjective feature selection, BioMedical Engineering OnLine 15 (Suppl 1) (2016) 73, doi:10.1186/s12938-016-0178-x.

[44] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197, doi:10.1109/4235.996017.

[45] J. Ortega, A. Ortiz, P. Martín-Smith, J. Q. Gan, J. González, Deep belief networks and multiobjective feature selection for bci with multiresolution analysis, in: Proceedings of the 14th International Work-Conference on Artificial Neural Networks (IWANN'2017), Lecture Notes in Computer Science, Springer, Cádiz, Spain, 2017, pp. 28–39, doi:10.1007/978-3-319-59153-7_3.

[46] K. Lorenz, I. Rejer, Feature selection with nsga and gaam in eeg signals domain, in: Proceedings of the 8th International Conference on Hu-

man System Interactions (HSI), IEEE, Warsaw, Poland, 2015, pp. 94–98, doi:10.1109/HSI.2015.7170649.

[47] I. Rejer, Genetic algorithm with aggressive mutation for feature selection in bci feature space, Pattern Analysis and Applications 18 (3) (2015) 485–492, doi:10.1007/s10044-014-0425-3.

[48] D. Dernoncourt, B. Hanczar, J. D. Zucker, Analysis of feature selection stability on high dimension and small sample data, Computational Statistics & Data Analysis 71 (2014) 681–693, doi:10.1016/j.csda.2013.07.012.

[49] Z. He, W. Yu, Stable feature selection for biomarker discovery, Computational Biology and Chemistry 34 (4) (2010) 215–225, doi:10.1016/j.compbiolchem.2010.07.002.

[50] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Knowledge and Information Systems 12 (1) (2007) 95–116, doi:10.1007/s10115-006-0040-8.

[51] P. Yang, B. B. Zhou, J. Y.-H. Yang, A. Y. Zomaya, Stability of feature selection algorithms and ensemble feature selection methods in bioinformatics, in: Biological Knowledge Discovery Handbook: Preprocessing, Mining, and Postprocessing of Biological Data, Wiley Series on Bioinformatics: Computational Techniques and Engineering, John Wiley & Sons, Inc., Hoboken, New Jersey, 2013, pp. 333–352, doi:10.1002/9781118617151.ch14.

[52] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1) (1960) 37–46, doi:10.1037/h0026256.

[53] V. S. Handiru, V. A. Prasad, Optimized bi-objective eeg channel selection and cross-subject generalization with brain-computer interfaces, IEEE Transactions on Human-Machine Systems 46 (6) (2016) 777–786, doi:10.1109/THMS.2016.2573827.

[54] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731, doi:10.1109/TEVC.2007.892759.

[55] J. Knowles, D. Corne, The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation (CEC), IEEE, Washington, DC, USA, 1999, pp. 98–105, doi:10.1109/CEC.1999.781913.

[56] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, in: Proceedings of the 4th International Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2001), International Center for Numerical Methods in Engineering (CIMNE), Athens, Greece, 2001, pp. 95–100.

[57] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, B. Arnaldi, A review of classification algorithms for eeg-based brain-computer interfaces, Journal of Neural Engineering 4 (2) (2007) R1–R13, doi:10.1088/1741-2560/4/2/R01.

[58] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, F. Yger, A review of classification algorithms for eeg-based brain-computer interfaces: a 10-year update, Journal of Neural Engineering (in press) doi:10.1088/1741-2552/aab2f2.

[59] J. S. Ebersole, A. M. Husain, D. R. Nordli, Current Practice of Clinical Electroencephalography, 4th Edition, Wolters Kluwer, Philadelphia, PA, USA, 2014.

[60] P. Martín-Smith, J. Ortega, J. Asensio-Cubero, J. Q. Gan, A. Ortiz, A supervised filter method for multi-objective feature selection in eeg classification based on multi-resolution analysis for bci, Neurocomputing 250 (2017) 45–56, doi:10.1016/j.neucom.2016.09.123.

[61] M. Aharon, M. Elad, A. Bruckstein, K-svd: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322, doi:10.1109/TSP.2006.881199.

[62] T. Tony Cai, L. Wang, Orthogonal matching pursuit for sparse signal recovery with noise, IEEE Transactions on Information Theory 57 (7) (2011) 4680–4688, doi:10.1109/TIT.2011.2146090.

[63] S. Luke, *et al.*, Ecj 24 and 25. a java-based evolutionary computation research system, `https://cs.gmu.edu/~eclab/projects/ecj/`, accessed: 2017-07-27.

[64] Apache, Commons math: The apache commons mathematics library, `http://commons.apache.org/proper/commons-math/`, accessed: 2017-07-27.

[65] R. Boostani, B. Graimann, M. H. Moradi, G. Pfurtscheller, A comparision approach toward finding the best feature and classifier in cue-based bcis, Medical & Biological Engineering & Computing 45 (4) (2007) 403–412, doi:10.1007/s11517-007-0169-y.

[66] R. Masoomi, A. Khadem, Enhancing lda-based discrimination of left and right hand motor imagery: Outperforming the winner of bci competition ii, in: Proceedings of the 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI 2015), IEEE, Tehran, Iran, 2015, pp. 392–398, doi:10.1109/KBEI.2015.7436077.

[67] Y. Wang, K. C. Veluvolu, Evolutionary algorithm based feature optimization for multi-channel eeg classification, Frontiers in Neuroscience 11 (2017) 28, doi:10.3389/fnins.2017.00028.

[68] H. Yu, J. Yang, A direct lda algorithm for high-dimensional data - with application to face recognition, Pattern Recognition 34 (10) (2001) 2067–2070, doi:10.1016/S0031-3203(00)00162-X.

[69] T. M. Cover, P. E. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27, doi:10.1109/TIT.1967.1053964.

[70] B. Blankertz, G. Curio, K. R. Müller, Classifying single trial eeg: Towards brain computer interfacing, in: Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, MA, USA, 2002, pp. 157–164.

[71] J. Borisoff, S. Mason, A. Bashashati, G. Birch, Brain-computer interface design for asynchronous control applications: improvements to the lf-asd asynchronous brain switch, IEEE Transactions on Biomedical Engineering 51 (6) (2004) 985–992, doi:10.1109/TBME.2004.827078.

[72] C. Liu, H. Wang, Z. Lu, Eeg classification for multiclass motor imagery bci, in: Proceedings of the 25th Chinese Control and Decision Conference (CCDC 2013), IEEE, May, 2013, pp. 4450–4453, doi:10.1109/CCDC.2013.6561736.

[73] T. Bassani, J. C. Nievola, Brain-computer interface using wavelet transformation and naïve bayes classifier, in: Brain Inspired Cognitive Systems 2008, Advances in Experimental Medicine and Biology, Springer, New York, NY, USA, 2010, pp. 147–165, doi:10.1007/978-0-387-79100-5_8.

[74] J. Machado, A. Balbinot, A. Schuck, A study of the naive bayes classifier for analyzing imaginary movement eeg signals using the periodogram as spectral estimator, in: Proceedings of the 2013 ISSNIP Biosignals and Biorobotics Conference (BRC 2013), IEEE, Rio de Janerio, Brazil, 2013, doi:10.1109/BRC.2013.6487514.

[75] Siuly, H. Wang, Y. Zhang, Detection of motor imagery eeg signals employing naïve bayes based learning process, Measurement 86 (2016) 148–158, doi:10.1016/j.measurement.2016.02.059.

[76] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324, doi:10.1016/S0004-3702(97)00043-X.

[77] C. Ambroise, G. J. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proceedings of the National Academy of Sciences of the United States of America 99 (19) (2002) 6562–6566, doi:https://doi.org/10.1073/pnas.102102699.

[78] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization, Journal of Heuristics 15 (2009) 617–644, doi:10.1007/s10732-008-9080-4.