U.P.B. Sci. Bull., Series C, Vol. 75, Iss. 2, 2013 ISSN 2286 – 3540

A GENETIC ALGORITHM AND ENSEMBLE OF SVM APPROACH FOR BCI P300 CLASSIFICATION PROBLEM

Salim GHOGGALI[1], Mihaela UNGUREANU[2], Rodica STRUNGARU[3]

The latest BCI literature shows that support vector machine (SVM) methods generally outperform traditional statistical and neural methods in classification problems involving P300 speller. However, there are still open issues that, if suitably addressed, could allow further improvement of their performances in terms of classification accuracy. Especially two issues are to be considered: 1) accurate detection of the user targeted characters and 2) choosing the best feature subspace where to carry out the classification task. In this study, the first issue is addressed through an ensemble of classifier approach, each classifier consisting of linear support vector machine (SVM) trained on a few set of the available data. The second issue is solve by a genetic optimization formulated in such a way that the best discriminative features are automatically detected, without requiring any a priori information. The effectiveness of the proposed classification method is assessed on the real dataset II of the BCI Competition III.

Keywords: Feature detection/ selection, BCI P300 speller, genetic algorithms (GAs), support vector machines (SVMs), Ensemble of SVMs classifiers.

1. Introduction

Brain-Computer Interfaces (BCIs) have gained increasing interest in the field of signal and image processing. The BCI is a cooperation between a subject/patient brain and a computer that decodes signals from the brain to direct some external tasks. During the past few years, research in the field of BCI has witnessed a fabulous development ([1], [2]) and is nowadays regarded as one of the most successful applications of the neurosciences. Indeed, such systems can provide a significant improvement of the quality of life for subjects suffering from some form of disability.

Two famous techniques for BCIs control can be found in the literature, i.e., the invasive and non-invasive one. In invasive BCIs technique, a micro-

[1] PhD Student, Dept. of Applied Electronics and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: ghoggalisalim@yahoo.com

[2] Prof., Dept. of Applied Electronics and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: mickyungureanu@yahoo.de

[3] Prof., Dept. of Applied Electronics and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: strungar@elmed.pub.ro

electrode array is implanted in the brain (mainly in the motor or premotor frontal areas or into the parietal cortex, [3]), while in non-invasive BCIs technique, mostly electroencephalograms (EEGs) are recorded from the scalp. Many types of EEG-based BCIs can be found; for instance some are based on Steady State Visually Evoked Potential (SSVEP, [4]); they are based on the detection of the activity of the brain at a specific frequency corresponding to the intensifying frequency of a visual stimulus ([5], [6]). Another type of BCIs relies on the detection of mental tasks (imagination of right/left hand movements, subtraction, word association, etc.) which are detected through slow co-cortical potentials (SCP) [7], readiness potential [8] and event-related desynchronization (ERD) [9].

The BCI control strategy considered in this paper belongs to another category; it is based on the detection of the P300 Event-Related Potential (ERP: stereotyped electrophysiological response to an internal or external stimulus, [10]), which allows subjects to write a text on a computer. It is based on the oddball paradigm: the task is to discriminate between epochs containing a positive-going potential evoked at a latency of about 300 ms by target stimuli (rarely presented) and epochs associated with the non-target (frequently present) stimuli. In order to achieve a good generalization capability for the classifier, it is necessary to collect enough training samples. For that, in real application scenarios, each symbol/stimulus is flashed several times. This is done at the expense of decreasing the speed of the system: e.g., when considering 15 repetitions, about two symbols can be spelt per minute only [11]. Moreover, the signal-to-noise ratio (SNR) of the P300 ERP signals is very low, which makes this potential almost undetectable in single trial. To increase SNR and hence classification accuracy, the sequence of intensifications are repeated several times, in order to average the EEG responses and increase the signal-to-noise ratio.

The first spelling paradigm based on the detection of the P300 was introduced in 1988 by Farwell and Donchin [11]. This application is nowadays one of the most studied BCI and the current study considers this very system. In this paradigm, the subject is presented with a 6 by 6 matrix of characters as shown in Fig. 1.a. The subject's task is to spell the word displayed on top of the matrix, one character at time. For the spelling of a single character, each of the 12 rows and columns (6 rows and 6 columns) of this matrix were successively and randomly intensified. Rows/columns are consecutively and randomly intensified. In order to make the spelling procedure more reliable, the sets of 12 intensifications were repeated 15 times for each character sequence (i.e., any specific row/column was intensified 15 times and thus there were 180 total intensifications for each character sequences). The subject focuses on one out of 36 different characters of the matrix. Two out of 12 intensifications of rows or columns contained the desired character. The EEG signals have been acquired using 64- channels. A more detailed description of the dataset can be found in the BCI competition online web site [12].

The problem addressed in this paper is to predict if the post-intensification segments (i.e., the 64-channel signals collected after the intensification of a row or column, named a post-intensification segments) contains P300 ERP or not. This first part is a binary classification problem that applied 15 times corresponding to the number of sequences in each character spelling. The second part of the problem deals with a 36-class classification problem to recognize a character from a 6 by 6 matrix.

In literature, in order to cope with this problem, Hoffman et al. [13] adopted a boosting strategy with orthogonal least square (OLS) as a classifier. Lenhardt et al. and Bostanov [14] used the linear discriminant analysis (LDA) to minimize the overlap between the two classes. Using the same reasoning in LDA, a Bayesian LDA (BLDA) classifier was proposed in [14]. The support vector machine (SVM) [16][17], has been used in BCI researches [14]. Recently, Rakotomamonj [15] utilized an ensemble of SVM classifiers instead of a single SVM, to improve the detection rate.

The current paper proposes a novel approach based on the genetic algorithms and SVMs. The choice of SVMs is motivated by the fact that they seem so far the best classifiers for P300 BCI's, thanks to the margin maximization principle they are based on, which provides them with a higher generalization capability.

The remaining part of this paper is organized as follows. First, we recall the basic concepts of SVM and genetic algorithms (GA) and then we present the proposed classification approach. After that, a description of the datasets and the experiments performed are provided. The Experimental results and the conclusions are provided finally.

2. Support Vector Machines

For simplicity, let us first consider a binary classification problem. Let us assume that the training set consists of N vectors xi ∈ ℜd (i = 1, 2, …, N) from the d-dimensional feature space X. A target, yi ∈ {-1, +1}, is associated to each vector xi. The linear SVM classification approach consists in finding an optimal hyperplane that maximizes the separation (margin) between the two classes in X. In the nonlinear case, data are first mapped with a kernel method in a higher dimensional feature space, i.e., Φ(X) ∈ ℜd' (d'> d). The membership decision rule is based on the function sign[f(x)], where f(x) represents the discriminant function associated to the hyperplane in the transformed space and is defined as:

$$f(x) = w^* \cdot \Phi(x) + b^*, \qquad (1)$$

The optimal hyperplane defined by the weight vector w* ∈ ℜd' and the bias b* ∈ ℜ is the one that minimizes the cost function that relies on two criteria: margin maximization and empirical risk minimization. When adopting a 1-norm measure of the empirical errors, the SVM cost function is defined as:

$$\Psi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i , \tag{2}$$

and is subject to the following functional margin constraints:

$$y_i(w \cdot \Phi(x_i) + b) \geq 1 - \xi_i, \quad \text{i = 1, …, N.} \tag{3}$$

with :

$$\xi_i \geq 0, \ \text{i = 1, …, N.} \tag{4}$$

The ξi's are the so-called slack variables introduced to account for non-separable data. The constant C represents a regularization parameter that allows controlling the trade-off between the model complexity and the empirical risk: large values of C favor the empirical risk minimization leading thus to complex decision boundaries and overfitting problems while small values of C runs into model simplicity and hence underfitting risk. The dual formulation of the above optimization problem is given by:

$$\max_{\alpha}\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) , \tag{5}$$

under the constraints :

$$\alpha_i \geq 0 \text{, i = 1, …, N.} \tag{6}$$

and :

$$\sum_{i=1}^{N}\alpha_i y_i = 0 , \tag{7}$$

where α=[α1, α2,…, αN] is a vector of Lagrange multipliers. The final result is a discriminant function conveniently expressed as a function of the data in the original (lower) dimensional feature space X:

$$f(\mathbf{x}) = \sum_{i \in S}\alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^* , \tag{8}$$

where K(·,·) is a kernel function. The set S is a subset of the indices {1, 2, …, N} corresponding to the non-zero Lagrange multipliers αi's which define the so-called support vectors. The kernel K(·,·) must satisfy the condition stated in

Mercer’s theorem so as to correspond to some type of inner product in the transformed (higher) dimensional feature space Φ(X). A typical example of such kernels is represented by the following Gaussian function:

$$K(\mathbf{x}_i, \mathbf{x}) = exp\left(-\gamma \left\| \mathbf{x}_i \text{ - } \mathbf{x} \right\|^2\right), \tag{9}$$

where γ represents a parameter inversely proportional to the width of the Gaussian kernel.

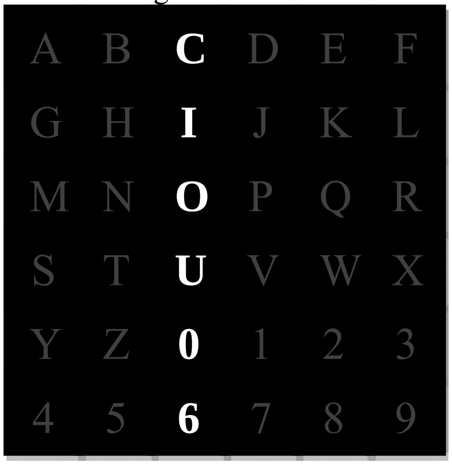
3. Genetic Algorithms

Genetic algorithms (GAs) are optimization techniques which exploit principles inspired from the principle of evolution and natural genetic [18], [19]. A genetic optimization algorithm performs a search by evolving a population of candidate solutions (individuals) modeled with “chromosomes”. From one generation to the next, the population is improved by mechanisms derived from genetics. The most common form of GAs involves the following steps. First, an initial population of chromosomes is randomly generated. Then, the goodness of each chromosome is evaluated according to a predefined fitness function, Selection operation selects the best and reject the worst chromosome among the population by using an appropriate selection rule based on the principle that the better the fitness, the higher the chance of being selected through some fitness function. This fitness evaluation step allows keeping the best chromosomes. Once the selection process is over, the next step is devoted to reproducing a new population. This is done by applying crossover and mutation operators. The entire process is iterated until a user-defined convergence criterion is reached.

4. Problem formulation

The problem analyzed in the current study is described as following: given the 64-channel (see Fig. 1.b for electrode configuration) signals collected after the intensification of a row or column, named a post-stimulus signal, it is desired to find out if the recorded signals contains P300 ERP component. Therefore, a discrimination between the epochs which have P300 ERP component and the epochs which have no P300 ERP component, is accomplished (i.e., the first part of the problem is a binary classification problem). Afterwards, according to the previous classification of each post-stimulus signal, the aim is to correctly predict which row or column is related to desired character. Hence, this second part of the problem deals with a multiclass classification problem since the aim is to recognize a symbol from the 6 × 6 matrix as given in Fig. 1.a.

To solve this problem, the study considers a training set consisting of signals recorded using a 64-electrode cap (Fig. 1 b), for which the target characters are labeled. Note that these characters come from the spelling of word, but the chronology of spelled characters has been lost because it was decided to scramble them. For each subject, the training set is made of 85 spelling characters which correspond to 15300 = 12 × 15 × 85 post-stimulus labeled signals from which 2550=15x2x85=2550 are targets.



A	B	**C**	D	E	F
G	H	**I**	J	K	L
M	N	**O**	P	Q	R
S	T	**U**	V	W	X
Y	Z	**0**	1	2	3
4	5	**6**	7	8	9

Fig. 1a. Example of 6x6 user display in P300 speller.

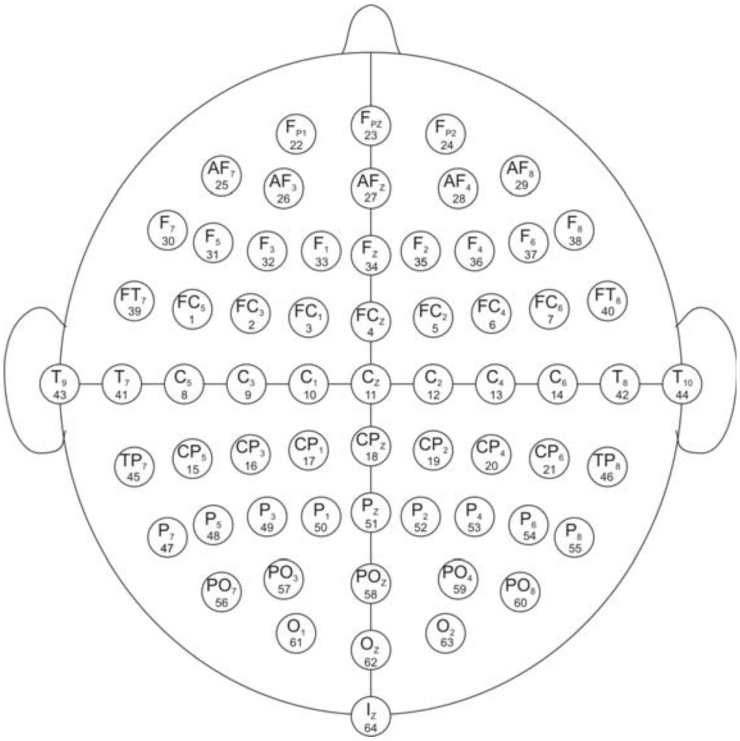


Fig. 1b. EEG electrode placement system.

4. Data preprocessing and feature extraction

Firstly, all data samples between 0 to 667 ms, following the beginning of intensification, are extracted from each channel. Considering that P300 ERP component appears about 300 ms after the stimulus, this window is large enough to capture all the time features demanded by an efficient classification process. Afterwards, each extracted signal is bandpass filtered in the frequency range 0.1-10 Hz by applying an 8-order Chebyshev bandpass filter and is downsampled, while considering the high cutoff frequency, in order to reduce the signal size to 14 samples.

After this preprocessing stage, post-stimulus signals are transformed into a feature vector by concatenating the 14 samples of all the 64 channels. Thus, for a single subject, the training set is composed of 15300 post-stimulus vectors xi of dimension 896 = 14×64, having the labels yi = {1,−1}.

- ***Phase 1: SVM Classifier Design***

  Step 1.1) Generate 17 different partitions based on the available training set. Each training partition is composed of 5 characters and thus 900 (i.e.: Single character spelling corresponds to 180 post-stimulus signals) training samples of dimension 896 are provided by considering all the 64 channels.

  Step 1.2) Train a binary SVM classifier with a linear kernel using just the training samples, after estimating its best parameter values with the *k*-fold cross-validation procedure (e.g., partitioning a sample of data into *k* subsets, performing the analysis on one subset the training set), and validate the analysis on the other subset (validation set). In order to reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.). The goodness of this classifier is evaluated according to $C_{cs} = \frac{tp}{tp + fp + fn}$, tp, fp, fn where are respectively the number of true positive, false positive and false negative as stated in A. Rakotomamonj [15]. It is worth noting that the number of true negative is ignored in the C_{cs} criterion for the reason that positive examples are fewer than negative examples.

- ***Phase 2: Root Chromosome Generation***

  Step 2.1) Build a root chromosome $C_1 = [ch_1, ch_2, ..., ch_{64}]$ composed of 64 binary genes, each gene being either 1, meaning that the corresponding channel is selected, or 0, meaning the corresponding channel is omitted.

- ***Phase 3: Genetic Algorithm Initialization***

Step 3.1) Generate an initial chromosome population $P^{(t)}$ (t=0) of size D by randomly disturbing D-1 times the root chromosome. The perturbation process is done as follows: for each chromosome gene, the real number p_g is generated, having a random uniformly distribution in the interval [0, 1]. If p_g is less than a user-defined probability then it is assigned to the gene either 0 or 1. The process is repeated till an initial population $P^{(t)}$ of D chromosomes is built.

- ***Phase 4: Genetic Optimization***

Step 4.1) For each chromosome $C_1 = [ch_1, ch_2, ..., ch_{64}]$ of the population $P^{(t)}$, a new training set is constructed, in which the channels corresponding to the genes value set to 0 are discarded. Then, a cross-validation is run, in order to estimate the fitness function C_{cs}, as mentioned in step 1.2.

Step 4.2) ***T***wo chromosomes are selected from the current population for the reproduction, using the roulette wheel spinning method. In this way, the best chromosomes will be more likely to be selected than the worst ones and thus will contribute in the generation of the new offspring.

Step 4.3) Two children are generated, by sequentially applying the crossover and mutation operators, to the two selected chromosomes. The former operation is accomplished first by generating a real random value between [0, 1]. If this value is smaller than a user-defined crossover probability, the two candidate chromosomes undergo a crossover operation; otherwise, the children are taken as the exact copy of their parents. After crossover, each of the two obtained children is subject to mutation in order to introduce randomness in the search process. This is done by flipping each bit with a user-defined mutation probability; the objective of the mutation is to prevent the genetic process to get trapped in a local optimum. Then, the two resulting offspring are then inserted into the new population.

Step 4.4) Return to Step 4.3), unless the new population reaches the size D.

Step 4.6) If the maximal number of generations is not reached go to *Step 4.2*.

- ***Phase 5*: Test Classification**

Step 5.1) At the end of the genetic optimization process, pick up the best chromosome from the final population in terms the fitness function C_{cs}**.**

Step 5.2) Using the best chromosome, ***c***onstruct the resulting new training set. This will be latter fed to our classifier as a training set, to predict the test samples associated to a given row or column.

Step 5.3) In order to predict which character correspond the given test post stimulus vector $x_{r|c}$ associated to a given row or column. After *J* sequences, the most probable row or column that maximizes the score: $S_{r|c} = \frac{1}{J}\frac{1}{K}\sum_{j=1}^{J}\sum_{k=1}^{K} f_k(x_{r|c}^{(j)})$ is chosen. In other words, the scores of all the *K* classifiers (*K* = 17) are fused together to produce a single predicted character, which leads to a more robust classification scheme since the worst classifier in term of score will be corrected by the remaining classifiers.

5. Data description

The P300-speller was considered for the first time in 1988 by Farwell and Donchin [11], who developed a procedure whereby a subject is presented a 6 × 6 character matrix as illustrated in Fig. 1. For the spelling of a single character, each of the 12 rows and columns of the matrix is then intensified according to a random sequence (a sequence is considered such a set of 12 intensifications). The subject is asked to focus its attention on the character he wants to spell, and then a P300 evoked potential appears in the recorded EEG, in response to the intensification of a row or column containing the desired character. In order to make the spelling procedure more reliable, this sequence of intensifications is repeated 15 times for each character to spell. The motivation behind choosing such dataset is that the results of this paper can be compared with the ones of other works which were presented in different papers.

The experimental validation was conducted on the basis of the dataset from the BCI 2003 competition, which is still available in the competition page [20]; it has been recorded from two different subjects and 5 different spelling sessions. Each session is composed of runs, and for each run, a subject is asked to spell a word. For a given acquisition session, all EEG signals of a 64-channel scalp recording (see Fig. 1) have been continuously collected. Before digitization at a sample rate of 240 Hz, the signals have been bandpass filtered from 0.1 Hz to

60 Hz [12]. A more detailed description of the dataset can be found in the BCI competition paper [21].

6. Experimental results

This section presents the classification results achieved for two different datasets, ‘A’ and ‘B’, using two different scenarios. For the first scenario, the root chromosome is initialized randomly, without any a priory information about the channels. In contrast to the first scenario, in the second one, the root chromosome is initialized with the channels obtained by Rokotomamonj algorithm. Finally, we carried out an experimental comparison with the 3 best algorithm submitted to the competition, namely the Rokotomamonj, Yandong and Zongtan algorithms. In all the experiments, we used the following standard parameters for the genetic inflation process: population size D= 20; crossover probability pc=0.9; mutation probability pm=0.01; and maximum number of generations set to 50. The SVM classifier parameter C is varied in the range [0.01, 0.05, 0.1, 0.5, 1].

The test sets have been processed in the same way as the training set and then are fed to our proposed classifiers. For the competition, performances have been evaluated based on the accuracy (i.e., the ratio of the total number of test samples correctly classified to the total number of test samples) of predicted characters in the test sets composed of 200 spelling characters.

Table 1

Percentage accuracy achieved by our algorithm on the test set

	Number of sequences			
	Scenario 1		Scenario 2	
Subject	5	15	5	15
A	50	89	71	96
B	65	92.5	74	97
Mean	57.5	90.75	72.5	96.5

Hence, the 36-class classification problem is considered, and each predicted character is obtained according to the above described method.

Table 1 depicts the performance achieved on the test sets, with respect to the number of sequences. Concerning the scenario 1, for 5 and 15 sequences we obtained accuracy of 57.5% and 90.75% respectively. In scenario 2, we obtained accuracy of 70% and 94.5% for 5 and 15 sequences respectively.

Interestingly, the first ranked (Rokotomamonj) and the second ranked (Yandong) algorithms have results similar to ones obtained by applying the proposed algorithm. Indeed, they use a multiple classifier strategy where each single classifier is a SVM. In the first ranked, each classifier is composed of a linear Support Vector Machine trained on a small part of the available data and for which recursive channel elimination is used to select a subset of channels. The score of these classifiers are fused together in order to produce a single result. However, in the second ranked algorithm, instead of summing the score of each single classifier, a voting strategy is used. The algorithm also uses a different procedure for channel selection. It has been decided to use a fixed number of channels, chosen in an ad-hoc way. These channels are different for each subject. Zongtan utilizes the T-statistic criterion. After decorrelating the signals by applying the principal component analysis (PCA), the optimized weighted sum of EEG signal is computed to construct new features that feed to the classifier.

Table 2 present the results obtained for the proposed algorithm and those of the mentioned algorithms. These latter have been obtained from BCI competition website. From this table, the following observations can be drawn.

Table 2

Comparison with the average accuracy of the three best competitors algorithms

	Number of sequences	
Algorithm	5	15
The proposed algorithm scenario 2	72.5	96.5
1st ranked algorithm	73.5	96.5
2nd ranked algorithm	55	90.5
3rd ranked algorithm	59.5	90

For 5 character sequences, the proposed algorithm is adopted, when the best case (scenario 2) scenario, provided a decrease of 1% as compared to the 1st ranked algorithm; this can be explained by the fact that the genetic optimization have been trapped into the local minimum (i.e.: after 20 generations it converges to a suboptimal solution). A gain of 17.5%, and 13% is obtained for the proposed algorithm as compared to the 2nd and 3rd ranked algorithms, respectively (see Fig. 2).

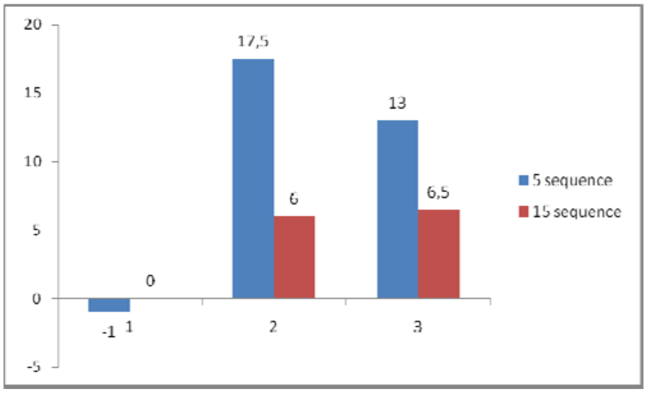


Fig. 2. Overall error obtained by proposed and the competitors algorithms.

For 15 character sequences (i.e., in greater details), the proposed algorithm provides a gain of 6% and 6.5% compared to the 2nd and 3rd ranked algorithms respectively, when the second scenario is adopted. However, it acts as well as the 1st ranked algorithm (i.e.: the same accuracy).

Furthermore, the advantage of the ensemble SVM classifiers technique is investigated. By comparing the results of those with the ones obtained by a single SVM trained on original hyperdimentional feature space (i.e.,without feature reduction) (see Table III). It is clear that the single SVM without channel selection for 15 character sequences yields to a performance similar to the one of the proposed algorithm, when all the channels are used. However, for only 5 sequences, ensemble SVM performs better than a single SVM (about 5% more). Also, it can be seen that the gap between the single and ensemble SVM, in terms of accuracy, is more significant when considering 5 sequences. Another interesting point, as seen from this table, is that when only 8 fixed channels are utilized, the ensemble of SVM classifier gives a gain about 9%. Note that these 8 prefixed channels are selected according to Huffman et al. [22]. Turning back to the results obtained by the proposed algorithm, the worst scenario (scenario 1) provides a decrease of 17% and 12% for Ensemble SVM without channel selection and Single SVM without channel selection, respectively; the rusults are still much better than those obtained for the Single SVM with 8 prefixed channels and Ensemble SVM with 8 prefixed channels. For scenario 2, much better results are obtained for the proposed algorithm as compared to the other methods, excepting the last method which shows an about 2% higher average accuracy. Finally, the proposed algorithm yields to the best results for the best scenario, like the Single SVM without channel selection (table 3).

Table 3

Comparison with the average accuracy of the three best competitors algorithms

	Number of sequences	
Algorithm	5	15
The proposed algorithm scenario 1	57.5	90.75
The proposed algorithm scenario 2	72.5	96.5
Single SVM without channel selection, C = 0.01	69.5	96.5
Single SVM with 8 prefixed channels, C = 1e-5	31	70
Ensemble SVM with 8 prefixed channels	40	80
Ensemble SVM without prefixed channels	74.5	95.5

7. Conclusions and discussions

In this paper, a novel solution is presented for tackling the problem of classifying BCI P300 ERP in the very low signal-to-noise ratio EEG. It relies on genetic algorithm and ensemble of classifiers. The novelty of the approach consists in using a genetic optimization algorithm to indentify the best features and discard the useless. However, the main drawback is the computational load which can be significant high (up to several days). In general, the experimental results confirm the usefulness of proposed approach. However, as observed in the obtained results, the best accuracy is obtained with the 1st ranked algorithm for 5 sequences, its value being slightly higher than that obtained for the proposed algorithm. This can be explained by considering the convergence to a local (suboptimal) solution. It is reasonable to assume that the proposed approach can be much improved by pushing further the genetic optimization process (augment the number of the population and the number of the generation).

The detection of P300 waves in BCI P300 Speller remains a very challenging problem, since it has a large variability over subjects. As its presence is unclear, it presents high potential of outliers for the classification, both for the machine learning algorithms and neuroscience experts; in addition, many trials are demanded due the low SNR ratio. In order to enhance it, more efforts have to be involved.

REFERENCES

[1]. "Is this the bionic man?" Nature, **vol. 442**, 2006, p. 109.

[2]. *P. Sajda, K. Muller, and K. Shenoy*, "Brain-Computer Interfaces [from the guest editors]", IEEE Signal Processing Magazine, **vol. 25**, 2008, pp. 16-17.prosthetics," Current biology: CB, **vol. 16**, 2006, pp. R77-80.

[3]. *B. Pesaran, S. Musallam, and R.A. Andersen*, "Cognitive neural prosthetics", Current biology: CB, **vol. 16**, 2006, pp. R77-80.

[4]. *J.J. Vidal*, "Toward direct brain-computer communication", Annual review of biophysics and bioengineering, **vol. 2**, 1973, pp. 157-80.

[5]. *P. Martinez, H. Bakardjian, and A. Cichocki*, "Fully Online Multicommand Brain-Computer Interface with Visual Neurofeedback Using SSVEP Paradigm", Computational intelligence and neuroscience, **vol. 2007**, 2007, p. 94561.

[6]. *M. Cheng, X. Gao, S. Gao, and D. Xu*, "Design and implementation of a brain-computer interface with high transfer rates", IEEE transactions on bio-medical engineering, **vol. 49**, 2002, pp. 1181-6.

[7]. *N. Birbaumer, et al.*, "The thought translation device (TTD) for completely paralyzed patients", IEEE transactions on rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society, **vol. 8**, 2000, pp. 190-193.

[8]. *B. Blankertz, G. Dornhege, M. Krauledat, K. Müller, and G. Curio*, "The non-invasive Berlin Brain-Computer Interface: fast acquisition of effective performance in untrained subjects", NeuroImage, **vol. 37**, 2007, pp. 539-50.

[9]. *J.R. Wolpaw, D.J. McFarland, and T.M. Vaughan*, "Brain-computer interface research at the Wadsworth Center", IEEE Transactions on Rehabilitation Engineering, **vol. 8**, 2000, pp. 222-226.

[10]. *S.J. Luck*, An Introduction to the Event-Related Potential Technique, MIT Press, 2005.

[11]. *L. A. Farwell and E. Donchin*, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," Electroencephalography and Clinical Neurophysiology, **vol. 70**, no. 6, pp. 510–523, December 1988.

[12]. *G. Schalk, et al.*, "BCI2000 : a general-purpose brain-computer interface (BCI) system", IEEE Transactions on Biomedical Engineering, **vol. 51**, no. 6, pp. 1034–1043, 2004.

[13]. *U. Hoffmann, et al.*, "A boosting approach to p300 detection with application to brain-computer interfaces", in Proc. rd Int. IEEE EMBS Coni. Neural Engineering, 2005.

[14]. *Vi-Hung Liu, et al.*, "An improved SVM-based real-time P300 speller for brain-computer interface", in Proc. Conf. Systems Man and Cybernetics (SMC), pp. 1748–1754, 2010.

[15]. *A. Rakotomamonjy and V Guigue*, "BCI competition III: dataset II-ensemble of SVMs for BCI p300 speller", IEEE Trans. Biomedical Engineering, **vol. 55**, no. 3, pp. 1147-1154, 2008.

[16]. *R. O. Duda, P. E. Hart, and D. G. Stork*, Pattern Classification, New York 2nd Edition.: Wiley, 2001.

[17]. *V. Vapnik*, Statistical Learning Theory. New York: Wiley, 1998.

[18]. *D.E. Goldberg*, Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley, 1989

[19]. *L. Davis*, Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold, 1991.

[20]. *B. Blankertz*, "Bci competition III webpage." [Online]. Available: http://ida.firstfraunhofer.de/projects/bci/competition ii

[21]. *B. Blankertz, et al.* "The BCI competition iii: Validating alternative approaches to actual bci problems", IEEE Transactions on Neural Systems and Rehabilitation Engineering, **vol. 14**, no. 2, pp. 153–159, 2006.

[22]. *Ulrich Hoffmann, et al.*, "An efficient P300-based brain-computer interface for disabled subjects," Journal of neuroscience methods, **vol. 167**, No. 1., pp. 115-125, 2008.