A Principal Component Analysis Ensemble Classifier for P300 Speller Applications

Amr S. Elsawy, Seif Eldawlatly, Mohamed Taher and Gamal M. Aly
Computer and Systems Engineering Department
Faculty of Engineering, Ain Shams University
Cairo, Egypt
aselsawy@eng.asu.edu.eg

***Abstract*—Recent advances in developing Brain-Computer Interfaces (BCIs) have opened up a new realm for designing efficient systems that could enable disabled people to communicate. The P300 speller is one important BCI application that allows the selection of characters on a virtual keyboard by analyzing recorded electroencephalography (EEG) activity. In this work, we propose an ensemble classifier that uses Principal Component Analysis (PCA) features to identify evoked P300 signals from EEG recordings. We examine the performance of the proposed method, using different linear classifiers, on the datasets provided by the BCI competition III. Results demonstrate a classification accuracy of 91% using the proposed method. In addition, our results indicate a significant improvement in classification accuracy compared to traditional feature extraction and classification approaches. The proposed method results in low across-subjects variability compared to other methods with minimal parameter tuning required which could be useful in mobile platform P300 applications.**

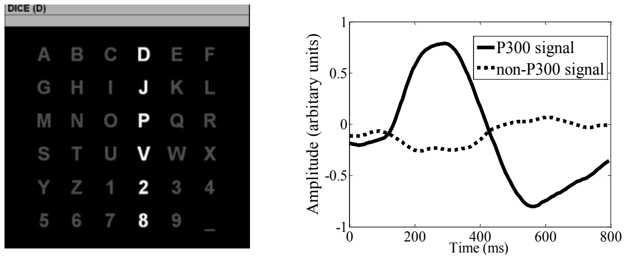
Keywords—BCI; P300 speller; PCA; ensemble classifier

I. INTRODUCTION

A brain-computer interface (BCI) is a device that provides a communication channel between the brain and the computer to compensate for the impairment of normal peripheral nerves or muscles [1]. While various successful applications of BCIs have been recently introduced [2], one of the early and widely studied applications is the P300 speller which was first introduced by Farwell and Donchin [3]. In this paradigm, an event-related potential (ERP), termed P300, is evoked in scalp-recorded electroencephalography (EEG) ~300 ms post rare stimulus presentation. A typical interface for a P300 speller application consists of a 6 by 6 grid of characters as illustrated by Fig. 1a. The rows and columns of the grid are intensified randomly one at a time while the user focuses on a specific target character. When the row or the column containing the target character is intensified, a P300 signal is evoked as shown in Fig. 1b. The target character is uniquely determined by the identity of the row and column upon which their intensification results in eliciting the P300 signal.

From a machine learning perspective, this problem can be considered as a binary classification problem where the classifier discriminates among two classes: P300 versus non-P300. However, dimensionality reduction is one basic step that is normally executed before applying the classifier to the data. This is a vital step given the limited number of training samples typically available. Common methods used in dimensionality reduction include decimation [4] and Principal Component Analysis (PCA) [5, 6]. Decimation is computationally effective and results in acceptable classification accuracy [4, 7]. However, it requires a limited number of channels. When a large number of recorded channels is used in the analysis, a larger decimation factor is required in order to keep the number of features low which reduces the quality of the analyzed signal and consequently leads to low classification accuracy [4]. As a result, a channel selection algorithm is typically used to select the most relevant channels [8]. As an alternative, PCA can be used with a subset or all channels which has the advantage of selecting more relevant features compared to decimation. However, PCA is computationally more expensive given the large size of input feature vectors [6]. As a result, a combination of decimation and PCA has been proposed which reduces the computational complexity by reducing the size of the input data while selecting the most relevant features for subsequent analysis [9].

In this paper, we introduce a novel approach for feature extraction that is subsequently used with an ensemble classifier to efficiently discriminate among the two signals: P300 versus non-P300. In this approach, PCA is first applied to the recorded filtered EEG of each channel individually. A number of classifiers that is equal to the number of extracted principal components are then trained where a classifier is constructed for the i^{th} principal component of all channels combined. The outputs of the classifiers are then fused based on the



(a) (b)

Fig. 1. (a) P300 Speller Interface [4]. (b) Mean responses of Cz channel for P300 and non-P300 signals for subject B of the BCI competition III.

significance of their corresponding principal components. We compare the performance of the proposed approach to that of traditional feature extraction and classification algorithms that use PCA or PCA plus decimation. We demonstrate an improved performance compared to traditional methods when using large number of channels obtained with minimal tuning.

II. Datasets

A. Data Description

We used the dataset of the BCI competition III which was recorded for two subjects and divided into four sets: two labeled sets and two unlabeled sets [10]. The training set of each subject consists of 85 characters and the test set consists of 100 characters. Each character epoch (i.e. trial) consists of a sequence of 12 intensifications representing 6 rows and 6 columns repeated 15 times resulting in a total of 180 (i.e. 12*15) intensifications per character epoch. The signals were recorded from 64 ear-referenced channels, bandpass filtered from 0.1-60Hz and sampled at 240Hz. More details on the datasets are available in [10].

B. Data Preprocessing

First, all the recorded channels signals were filtered using the common average reference spatial filter [11]. This is done by computing the mean of all channels within the considered character epoch and subtracting this mean value from each channel

$$r_i(j) = s_i(j) - \frac{1}{N}\sum_{k=1}^{N} s_k(j) \qquad (1)$$

where $s_i(j)$ represents the raw signal recorded on electrode i at time j, $r_i(j)$ represents the filtered signal and N is the total number of channels. A moving-average filter was then applied with a window of 25 samples to reduce the noise [7]. The training data were scaled using z-score

$$x_{ij} = \left(r_i(j) - \mu_i\right)/\sigma_i \qquad (2)$$

where x_{ij} denotes feature j of channel i (i.e. $x_i(j)$), μ_i is the average of the signals recorded on channel i and σ_i is the corresponding standard deviation. The test data were scaled according to the scaling parameters obtained from the training data. Finally, when decimation is applied, a decimation factor of 12 was used [4].

III. Feature Extraction

A data segment of 800 ms (i.e. 192 samples) post stimulus presentation was extracted for each channel [4]. Two sets of channels were used in our analysis. The first set comprised 8 channels in which channels Fz, Cz, Pz, P3, P4, PO7, PO8 and Oz were selected [4, 7]. We also examined the performance when all 64 channels available in the dataset were used in the analysis.

A. Concatenated Feature Vector

The concatenated feature vector refers to the standard feature vector that is provided by the BCI competition III data. In this feature vector, the data segments obtained from all channels for one character epoch are concatenated to constitute one feature vector as illustrated by Fig. 2. We investigate the performance in two cases: when decimation is applied prior to concatenation and when it is not applied. The feature vector size is then reduced using PCA both in the decimated and undecimated cases. PCA reduces the features dimensionality by projecting the data into reduced dimensions that preserve most of data variance [12]. This is done by, first, subtracting the mean of each feature to center the data. The covariance matrix is then computed based on the centered data. The eigenvectors of the covariance matrix are then obtained and sorted based on their corresponding eigenvalues.

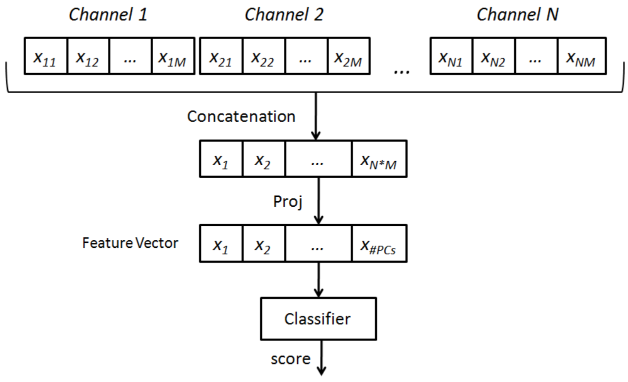

Channel 1
Channel 2
Channel N
x_{11} x_{12} … x_{1M}
x_{21} x_{22} … x_{2M}
…
x_{N1} x_{N2} … x_{NM}
Concatenation
x_1 x_2 … x_{N*M}
Proj
Feature Vector
x_1 x_2 … $x_{\#PCs}$
Classifier
score

Fig. 2. Concatenated feature vector in which features from each channel are concatenated into one vector. The concatenated feature vector is projected on the principal components to give the final feature vector that is fed to classifier.

In our analysis, principal components that account for 99.9% of variance in training data are selected in case of 8 channels [6] and 99% of variance are selected in case of 64 channels. Lower variance is used in the case of 64 channels to decrease the number of selected principal components given the limited number of training samples. The training dataset is then projected using the selected principal components and used to train the classifiers. The test datasets are projected on the same principal components obtained from the training datasets.

B. Proposed Feature Vector

In the proposed method, first, PCA is performed on each channel of the training data separately. The principal components are then sorted according to the variance in the data accounted for by each principal component. When the feature vector is formed for decimated data, all the principal components are used in the analysis (i.e. 16 principal components for each channel since the decimation factor was 12 and so 192 samples/12 = 16). For the undecimated data, we select the principal components by, first, computing the average of the variance accounted for by each principal component across channels. We then select the principal components that account for 99.9% of the average variance in the 8 channels case and 99% of the average variance in the 64 channels case. The training data are then projected using the selected principal components. Each feature vector is formed

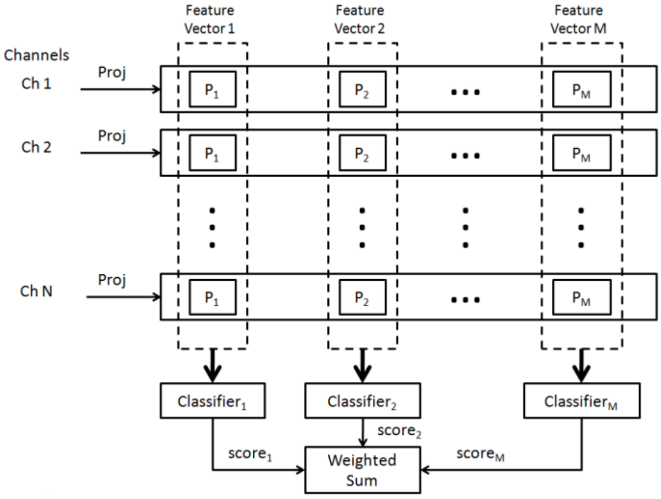


Fig. 3. Proposed feature extraction and ensemble classifier. Each channel is projected using its principal components. The corresponding projections are concatenated to constitute a feature vector. A classifier is then trained for each principal component. The final score is a weighted sum of individual classifiers scores.

by concatenating corresponding projections from the channel set as illustrated in Fig. 3. A classifier is then applied to the feature vector of each principal component as detailed in the next section. The test data are projected on the same principal components obtained from the training data.

IV. Classification Methods

The problem of identifying P300 signals in EEG activity is a binary classification problem where the goal is to classify the signals recorded post stimulus presentation as P300 or non-P300. In this study, linear classifiers were used. A linear decision boundary takes the form

$$w^T x + b = 0 \tag{3}$$

where *x* is the feature vector, *w* is the weight vector and *b* is the bias term.

A. Linear Classifiers

We examined the performance of the linear classification methods given below to demonstrate the efficacy of the proposed approach compared to the concatenated feature vector independent of the used classification algorithm. Linear classifiers have the advantages of low computational complexity and need no tuning.

1) Linear Discriminant Analysis (LDA): is the simplest linear classifier that is based on least squared error. The solution weight vector is given by [12]

$$w = \left(X^T X\right)^{-1} X^T y \tag{4}$$

where *X* is a matrix whose rows are the feature vectors and y is a vector that contains the labels of the feature vectors.

2) Fisher's Linear Discriminant (FLD): is used to find the optimal separating decision boundary between the two classes. For linear problems, its solution is the same as LDA. However, if the bias term is dropped, its solution need not be the same as LDA [12]. The weight vector is given by

$$w = S_W{}^{-1}\left(m_1 - m_2\right) \tag{5}$$

where m_1 and m_2 are the means for each class and S_W is the within-class scatter matrix and is given by

$$S_W = \mathrm{cov}(X_{P300}) + \mathrm{cov}(X_{non-P300}) \tag{6}$$

where cov(.) denotes the covariance, X_{P300} is a matrix whose rows are all P300-labeled feature vectors and $X_{non\text{-}P300}$ is a matrix whose rows are all non-P300-labeled feature vectors.

3) Linear Support Vector Machine (LSVM): the SVM is a powerful tool for separating data with high dimensions. Linear SVM objective function is given by [13]

$$J = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j x_i^T x_j \tag{7}$$

where α_i are Lagrange multipliers and t_i is the label of each feature vector and the objective function is subject to the constraints

$$\sum_{i=1}^{N} \alpha_i t_i = 0 \tag{8}$$

$$\alpha_i \geq 0, for\ i = 1,2,...,N \tag{9}$$

Using quadratic programming solvers, Lagrange multipliers that maximize the objective function can be found.

B. Classifying Concatenated Feature Vectors

For each character, a total of 12 feature vectors that correspond to the intensification of 6 rows and 6 columns are classified. However, since more than one row or column might be classified by the classifier as the target choice, we determine the target row and column as those that maximize $w^T x + b$. As a result, the bias term *b* is dropped since it is constant among all rows/columns. The classifier score thus takes the form

$$score = w^T x \tag{10}$$

The predicted row *r* for the target character is then determined by

$$r = \arg\max_{row}\left(w^T x_{row}\right) \tag{11}$$

and the predicted column *c* for the target character is determined by

$$c = \arg\max_{col}\left(w^T x_{col}\right) \tag{12}$$

C. PCA Ensemble Classifier

In the proposed method, we use an ensemble classifier in which a classifier for each principal component is trained and the total score is a weighted sum of the classifiers scores

$$total\_score = \sum_{i=1}^{M} \left(\eta_i * score_i \right) \tag{13}$$

where M is the number of classifiers and the score of each classifier is

$$score_i = w_i^T x_i \tag{14}$$

and the proposed weight η_i is given by

$$\eta_i = N \Big/ \left(\sum_{j=1}^{i} \sum_{k=1}^{N} \lambda_{jk} \right) \tag{15}$$

where λ_{jk} represents eigenvalue j of channel k obtained from the PCA analysis and N denotes the number of channels (i.e. 8 or 64). This measure assigns a weight to each score that is proportional to the significance of the corresponding principal component. Similar to the concatenated feature vector case, the predicted row for the target character is the row with maximum total score across all rows

$$r = \arg \max_{row} \left(total\_score_i \right) \tag{16}$$

and the predicted column for the target character is the column with maximum total score across all columns

$$c = \arg \max_{col} \left(total\_score_i \right) \tag{17}$$

V. Results

A. Cross Validation

We first examined the performance of different classifiers on the training data when four different feature extraction methods were used: the first is using decimation of the signal of each channel with a factor of 12 with PCA applied to the concatenated signals after decimation. The second is applying PCA to the concatenated feature vectors without decimation. In both cases, principal components that account for 99.9% of the variance were selected for the 8-channel analysis while, for the 64-channel analysis, principal components that account for 99% of variance were selected. The third method involves the proposed ensemble classifier where the signals of each channel are first decimated with a decimation factor of 12 with PCA applied to each channel separately and then all principal components projections are grouped as previously illustrated in Fig. 3. The fourth method involves the proposed ensemble classifier without decimation selecting the principal components that account for 99.9% of the variance for the 8-channel analysis or 99% of variance for the 64-channel analysis.

For each of the two subjects (A and B) provided by the BCI competition III data, a training dataset consisting of 85 characters was used. In the cross validation of each subject, multiple datasets of 70 characters were used to train the classifiers and the other remaining 15 characters were used for validation. We formed 11 overlapped datasets (with 70 training characters and 15 validation characters) with an overlap of 7 characters. Overlapping was used in order to have sufficient data for statistical significance tests. Each of the four feature extraction methods was tested on the data recorded from 8 channels in addition to all 64 channels. It is worth mentioning that cross validation was only used to compare the performance of different methods and not to tune the parameters of the proposed approach.

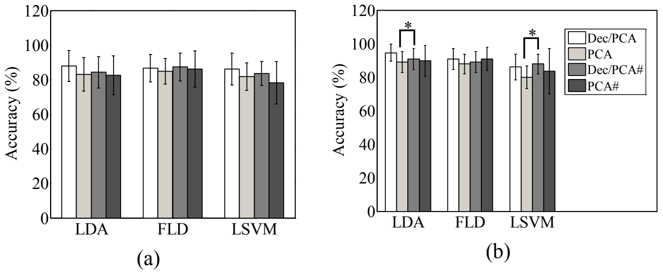


Fig. 4. (a) Subject A and (b) Subject B classification accuracy for different approaches using 8 channels subset. The PCA ensemble approach is denoted by #. *P < 0.05, two-sample t-test.

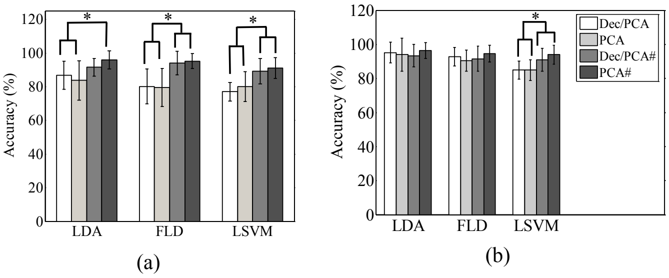


Fig. 5. (a) Subject A and (b) Subject B classification accuracy for different approaches using 64 channels subset. The PCA ensemble approach is denoted by #. *P < 0.05, two-sample t-test.

Fig. 4a illustrates the classification accuracy obtained for subject A using the 8 channels subset averaged across the 11 overlapped datasets (mean ± SD). As can be seen, all classifiers and feature extraction methods performed equally well with an average classification accuracy across all methods of 84.3± 9.1%. Similar performance was obtained for subject B using 8 channels as illustrated by Fig. 4b with average classification accuracy across all methods of 88.4±7.1%.

When all 64 channels are used in the analysis, the PCA ensemble approach results in a significant improvement in classification accuracy. Fig. 5a illustrates the classification accuracy obtained for subject A using 64 channels. Using the proposed ensemble approach without decimation (i.e. last column in the figures) resulted in a classification accuracy of 95.8± 5.4% for LDA, 95.2± 4.3% for FLD and 90.9± 6.2% for LSVM. In addition, the classification accuracy obtained using this method was significantly different compared to that obtained using the concatenated feature vector (P < 0.05, two-sample t-test). Similar performance was achieved with subject B using 64 channels as shown in Fig. 5b (Classification accuracy of 96.4±4.6% for LDA, 94.5±5% for FLD and 93.9±5.5% for LSVM) although not as significantly different as in subject A. These results indicate the efficacy of the proposed approach compared to the concatenated feature vector and traditional classification approach.

TABLE I. 8 CHANNELS RESULTS USING 15 TRIALS

Classification Method	Feature Reduction Method	A	B	Average
LDA	PCA	81	93	87±6%
	Dec/PCA	77	92	84.5±7.5%
	PCA#	83	83	83±0%
	Dec/PCA#	81	84	82.5±1.5%
FLD	PCA	81	88	84.5±3.5%
	Dec/PCA	76	87	81.5±5.5%
	PCA#	86	86	86±0%
	Dec/PCA#	83	88	85.5±2.5%
LSVM	PCA	74	84	79±5%
	Dec/PCA	70	77	73.5±3.5%
	PCA#	78	76	77±1%
	Dec/PCA#	82	82	82±0%

TABLE II. 64 CHANNELS RESULTS USING 15 TRIALS

Classification Method	Feature Reduction Method	A	B	Average
LDA	PCA	87	90	88.5±1.5%
	Dec/PCA	87	93	90±3%
	PCA#	88	89	88.5±0.5%
	Dec/PCA#	90	91	90.5±0.5%
FLD	PCA	83	90	86.5±3.5%
	Dec/PCA	81	88	84.5±3.5%
	PCA#	92	90	91±1%
	Dec/PCA#	93	90	91.5±1.5%
LSVM	PCA	80	77	78.5±1.5%
	Dec/PCA	75	77	76±1%
	PCA#	91	90	90.5±0.5%
	Dec/PCA#	92	90	91±1%

B. *Test Data Analysis*

Each classification method was examined using the same feature extraction methods investigated in the cross validation above. The test dataset consisted of 100 characters for each subject with 12 intensifications for each character epoch repeated 15 times. The classification accuracy was calculated based on the percentage of correct characters not on the percentage of correct classification of each individual feature vector. The results obtained using 8 channels are given in Table I. As in the cross validation, all methods performed equally well. However, the proposed approach results in the least variance in classification accuracy across subjects. The results obtained using 64 channels are given in Table II. The results indicate that the methods involving the proposed PCA feature extraction approach (denoted by PCA# in the tables) significantly outperform other methods for all cases with the least across-subject variability.

All methods were also tested using the first 5 trials as was required by the BCI III competition [10]. The results are shown in Table III and Table IV using 8 and 64 channels, respectively. The results show that our approach has in general a lower across-subjects variability for the same classifier. Results also demonstrate that using the proposed approach with FLD and SVM is better than using the concatenated feature vector in the 64-channel case.

TABLE III. 8 CHANNELS RESULTS USING FIRST 5 TRIALS

Classification Method	Feature Reduction Method	A	B	Average
LDA	PCA	40	70	55±15%
	Dec/PCA	33	70	51.5±18.5%
	PCA#	42	60	51±9%
	Dec/PCA#	40	62	51±11%
FLD	PCA	41	69	55±14%
	Dec/PCA	36	67	51.5±15.5%
	PCA#	46	60	53±7%
	Dec/PCA#	40	61	50.5±10.5%
LSVM	PCA	33	55	44±11%
	Dec/PCA	33	60	46.5±13.5%
	PCA#	37	53	45±8%
	Dec/PCA#	38	61	49.5±11.5%

TABLE IV. 64 CHANNELS RESULTS USING FIRST 5 TRIALS

Classification Method	Feature Reduction Method	A	B	Average
LDA	PCA	47	67	57±10%
	Dec/PCA	54	64	59±5%
	PCA#	43	63	53±10%
	Dec/PCA#	45	64	54.5±9.5%
FLD	PCA	37	63	50±13%
	Dec/PCA	41	57	49±8%
	PCA#	42	63	52.5±10.5%
	Dec/PCA#	51	57	54±3%
LSVM	PCA	32	57	44.5±12.5%
	Dec/PCA	33	48	40.5±7.5%
	PCA#	44	57	50.5±6.5%
	Dec/PCA#	45	56	50.5±5.5%

TABLE V. COMPETITION RESULTS

	Contributor	15 trials	5 trials
1	Alain Rakotomamonjy [8]	96.5%	73.5%
2	Li Yandong	90.5%	55.0%
3	Zhou Zongtan [9]	90.0%	59.5%

C. *Comparison with BCI Competition III Results*

The results of the P300 speller BCI competition III obtained by the contributors are available on the website [10] and the best three results are listed in Table V. In their analysis, all contributors used the concatenated feature vector. Compared to our results, the proposed ensemble classifier results in classification accuracy that is higher than the second place accuracy. It has to be noted though that our approach requires no tuning as demonstrated before in cross validation section and no channel selection is required as we use all the channels in the 64-channel case. In addition, we only used linear classifiers while the top two contributors used non-linear SVM which is expected to result in higher classification accuracy compared to the linear classifiers we examined.

VI. Conclusions

We demonstrated the efficacy of using a novel principal component analysis ensemble classifier for P300 speller applications. Results demonstrate that applying the proposed method with linear classifiers outperforms traditional concatenated feature vector when using all recorded 64 channels. The proposed method reduces the computation complexity as the feature vector size for each classifier is equal to the number of selected channels (i.e. max 64 features). In addition, results demonstrate consistent classification accuracy across the two investigated subjects compared to the case of concatenated feature vector. Moreover, the proposed method does not require any tuning which leads to less training time. This method can be thus employed for mobile platforms that have low processing power.

References

[1] M. v. Gerven, *et al.*, "The brain–computer interface cycle," *Journal of Neural Engineering,* vol. 6, p. 041001, 2009.

[2] J. R. d. Millán, *et al.*, "Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges," *Front. Neurosci.,* vol. 4, p. 161, 2010.

[3] 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, pp. 510-523, 1988.

[4] D. J. Krusienski, E. W. Sellers, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "Toward enhanced P300 speller performance," *Journal of neuroscience methods,* vol. 167, p. 15, 2008.

[5] Z. Cashero and C. Anderson, "Comparison of EEG blind source separation techniques to improve the classification of P300 trials," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, 2011, pp. 7183-7186.

[6] A. Finke, A. Lenhardt, and H. Ritter, "The MindGame: a P300-based brain–computer interface game," *Neural Networks,* vol. 22, pp. 1329-1333, 2009.

[7] D. J. Krusienski, *et al.*, "A comparison of classification techniques for the P300 Speller," *Journal of neural engineering,* vol. 3, p. 299, 2006.

[8] A. Rakotomamonjy and V. Guigue, "BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller," *Biomedical Engineering, IEEE Transactions on,* vol. 55, pp. 1147-1154, 2008.

[9] Y. Liu, Z. Zhou, D. Hu, and G. Dong, "T-weighted approach for neural information processing in P300 based brain-computer interface," in *Neural Networks and Brain, 2005. ICNN&B'05. International Conference on*, 2005, pp. 1535-1539.

[10] B. Blankertz, *et al.*, "The BCI competition III: Validating alternative approaches to actual BCI problems," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on,* vol. 14, pp. 153-159, 2006.

[11] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw, "Spatial filter selection for EEG-based communication," *Electroencephalography and clinical Neurophysiology,* vol. 103, pp. 386-394, 1997.

[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*: Wiley-interscience, 2012.

[13] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery,* vol. 2, pp. 121-167, 1998.