ARTÍCULO DE REVISIÓN

Vol. 41 | No. 1 | **ENERO - ABRIL 2020** | pp 43-56

REVISTA MEXICANA DE Ingeniería Biomédica

dx.doi.org/10.17488/RMIB.41.1.3

E-LOCATION ID: 926

A Bootstrapping Method for Improving the Classification Performance of the P300 Speller

A Bootstrapping Method for Improving the Classification Performance in the P300 Speller

J. H. Cristancho-Cuervo, J. F. Delgado-Saa

Universidad del Norte

ABSTRACT

In this paper, we present a novel approach to training classifiers in a speller based on P300 potentials. The method, based on bootstrapping, is a known strategy for generating new samples, but it is rarely used in neurosciences. The study first demonstrates how the performance of the classification task (detecting P300 and Non-P300 classes) could be sub-optimal in the traditional approach. Then, a new method for taking new samples from the training data is proposed. Each classifier is re-trained using balanced sub-groups of individual P300 and non-P300 samples. Data were collected from 14 healthy subjects, using 16 electroencephalography channels. These were filtered in bandpass and decimated. Subsequently, four linear classifiers were trained using the traditional method followed by the proposed one, with 1000, 2000 and 3000 samples per class. Results indicate an improvement in the accuracy and discrimination capacity of discriminative classifiers with the proposed method, maintaining the same statistical properties between the training and test data. By contrast, for generative classifiers, there is no significant difference in the results. Therefore, the proposed method is highly recommended for training discriminative classifiers in spell-based P300 potentials.

KEYWORDS: P300 speller, linear classifier, bootstrapping, training, averaging.

RESUMEN

Este artículo presenta un método novedoso para entrenar clasificadores en un deletreador basado en potenciales P300. El método, basado en bootstrapping, es una estrategia conocida para generar nuevas muestras pero escasamente implementado en neurociencias. El estudio muestra cómo el rendimiento de la detección de P300 (frente a No-P300) puede resultar sub-óptimo usando el método tradicional. Luego, se propone un nuevo método donde se toman nuevas muestras a partir de los datos de entrenamiento. Con ellas, se re-entrena al clasificador usando sub-grupos equilibrados de muestras individuales P300 y No-P300. Los datos se recolectaron de 14 sujetos sanos, usando 16 canales de electroencefalografía. Estos fueron filtrados en pasa-banda y diezmados. Posteriormente, cuatro clasificadores lineales fueron entrenados, usando primero el método tradicional y después el método propuesto, con 1000, 2000 y 3000 muestras por clase. Los resultados muestran una mejoría en la precisión y la capacidad de discriminación de clasificadores discriminativos con el método propuesto, manteniendo las mismas propiedades estadísticas entre los datos de entrenamiento y los de prueba. En contraste, para los clasificadores generativos, no existe una diferencia significativa en los resultados. Por consiguiente, el método propuesto es altamente recomendado para entrenar clasificadores discriminativos en deletreadores basados en potenciales P300.

PALABRAS CLAVE: Deletreador P300, clasificador lineal, bootstrapping, entrenamiento, promediado.

Correspondencia

DESTINATARIO: **Jaime Fernando Delgado Saa**
INSTITUCIÓN: **Universidad del Norte**
DIRECCIÓN: **Km. 5 Vía Puerto Colombia, Barranquilla, Atlántico, Colombia**
CORREO ELECTRÓNICO: **jadelgado@uninorte.edu.co**

Fecha de recepción:

15 de febrero de 2019

Fecha de aceptación:

4 de marzo de 2020

INTRODUCTION

One of the most interesting applications for Brain-Computer Interfaces (BCIs) is the P300 speller, proposed in 1988 by Farwell and Donchin [1] and re-invented and improved in many other studies [2] [3] [4] [5]. A commonly used speller consists of an arrangement of characters uniformly distributed in rows and columns, displayed in a screen. Rather than displaying a single character, the speller randomly highlights some characters organized in rows or columns. When the user watches the desired character in a highlighted row or column, the brain generates a *P300* signal, related to memory and the attention processes in the brain [6].

A typical P300 speller reads signals from the brain, using electroencephalography (EEG), and tries to discriminate between P300 and non-P300 signals. When a P300 signal is detected in a specific row and column, the speller takes the corresponding character and displays it on the screen. The described speller has been used for developing online BCI applications [5] [7] [8] [9]. Note that the target of the classification is to identify the row and the column that corresponds to a character from P300 signals rather than to classify P300 and non-P300 signals.

As the P300 speller is based in the oddball paradigm, the number of events is unbalanced; that is, the number of non-P300 trials is larger than the number of P300 trials [10]. Both unbalanced classes and small datasets could affect the performance measurement of a classifier [11] [12]. To get a more confident performance, the number of samples by class should be balanced.

Some researchers have proposed discarding samples randomly from the class with more members to reach the desired 1:1 proportion [13] [14] [15] [16] [17], trying to preserve as many samples as possible in the training stage [18]. This solves the problem of unbalanced classes at the expense of decreasing the number of available samples.

By contrast, there are mainly three approaches to processing the input features to a classifier for a P300 speller. The first one consists of training and evaluating the classifier in single trials [3] [8] [19]. The second approach makes use of averaged data over a fixed number of trials, for training and testing the system [5] [7] [9] [13] [16] [20] [21] [22]. The third approach consists of training the classifier in single trials, and evaluating the classifier with averaged trials [14] [17] [23] [24].

The last approach (called the *traditional* approach in this work) is commonly used in the literature. It suffers from an important problem of statistical properties of signals during the training stage being different from those of signals used during the testing stage. This violates the assumption that training and testing data should come from the same population, for any classification problem [25]. Consequently, the classifier has reduced capacity to differentiate between P300 and non-P300 trials.

Different statistical properties of training and testing signals carry another problem. The estimation of the posterior probabilities from probabilistic classification approaches would not be correct.

This issue is critical for P300 applications that make use of language models [7] [26] [27] [28] since the posterior probability of the output of the classifier is typically combined with the probability of letters in a particular language to determine the most likely sequence of letters.

In addition, since P300 and non-P300 classes are unbalanced, performance measures, such as the accuracy, tend to be biased. This happens because the classifier assigns most of the samples to the class with higher prior probability [12]. Some studies have proposed use of the *Cohen's kappa index* κ as an alternative measure of performance that does not suffer from the issues previously described [29] [30] [31].

The classification problem could be seen from one of two possible points of view. The first one establishes that the classification problem is typically divided into two stages. The *inference stage* tries to learn a probabilistic model of the data given the class. Then, the *decision stage* implements the theorem of Bayes to determine the class of each data. A classifier implemented in this manner is known as a *generative classifier* [25]. *Linear discriminant analysis* (LDA) classifiers are generative because they mostly assume Gaussian distributions in the data [25] [32].

The second point of view determines that a class could be directly mapped from the data. The model comes from either a probabilistic *discriminant model* of the class, given the data, or a deterministic *discriminant function* that directly maps the data to the class. A classifier that uses the latter approach is a *discriminative classifier* [25]. *Logistic regression* and the *support-vector machine* (SVM) are examples of discriminative classifiers that use a probabilistic model and a discriminant function, respectively.

In this work, a method for training linear classifiers in the identification of P300 potentials is presented. First, we demonstrate that the traditional approach could lead to misinterpretation of the actual performance of these classifiers, as the performance metric based on accuracy is not well suited for the cases of unbalanced classes. Second, a bootstrapping approach is presented as a method for obtaining effective training of linear classifiers. Results indicate a significant improvement using the proposed method for detection of P300 potentials.

METHODS

Experiment and dataset description

The experiment consists of declaring one of 36 possible characters (26 letters and 10 digits). Each subject observed a 6 × 6 matrix of characters in a screen, focusing the attention on the character that was prescribed above the matrix speller. For each character, the matrix was displayed for a 2.5 s period, and all characters had the same intensity. Afterward, each column and each row were randomly intensified for 100 ms, followed by a blank period of 75 ms after each intensification step. There were 12 different row/column stimuli by round and 15 rounds of intensifications by character, for a total of 31.5 s. Each subject spelled 32 characters in total. Fourteen healthy subjects participated in the study.

The dataset contains EEG signals that were recorded using a cap embedded with 64 electrodes, according to the modified 10-20 system [33]. All electrodes were referenced to the right earlobe and grounded to the right mastoid. The raw EEG signal was bandpass-filtered between 0.1 and 60 Hz and amplified with a 20000X SA Electronics amplifier [23]. Each experiment took into account only 16 EEG channels, motivated by the study presented by Krusienski et al. [23]: F3, Fz, F4, FCz, C3, Cz, C4, CPz, P3, Pz, P4, PO7, PO8, O1, O2, and Oz. Each channel is sampled at a rate of 240 Hz for one subject and 256 Hz for the others. All aspects of the data collection and experimental control were controlled by the BCI2000 system [22]. Two datasets were acquired for each subject: One was used for training, and the other was implemented for testing. Both databases were taken on different days. All datasets were obtained from the Wadsworth Center, NYS Department of Health.

Data processing

Data were pre-processed using bandpass filtering, separation in trials and decimation. Then, all channels were concatenated in a single vector. Depending on the type of training, data were taken from either the input of a classifier or a new population for obtaining new samples. In the latter case, a determined number of N averaged samples were taken. Afterward, the training datasets were inputs of a linear discriminant classifier. Details are explained in the following subsections.

Pre-processing

For each subject, data were bandpass-filtered between 1 and 20 Hz using a fourth-order Butterworth filter. The chosen bandwidth eliminates the trend of each channel and allows decimation of the signal later, by preventing the aliasing. Afterward, data were separated in trials by taking a window of 600 ms after the presentation of each visual stimulus (the highlight of one row or column), as proposed in a previous work [26]. Signals from all channels were decimated by a factor of 4 and concatenated in a single feature vector. The factor was chosen because frequencies higher than that of the beta band reflect unrelated neural processes to P300 in awareness [34]. In addition, the maximum analog frequency of the EEG signal is 60 Hz, as seen before [23]. For the averaged process, signal segments were averaged across repetitions, up to the maximum number of repetitions by character (15). Concatenated channels were used as the inputs of the classifier since they are used in the traditional method, as described in [23].

Re-sampling of training samples

In the traditional approach, the classifier is trained with single trials and tested on averaged trials, to increase the signal-to-noise ratio. Note that besides the issue of having unbalanced data, the statistical properties of the training data do not match those of the testing data.

To avoid these problems, we implement an approach based on bootstrap re-sampling (bootstrapping) [32] [35]. From the training trials, a new dataset is obtained by re-sampling N trials with replacement, where N is the number of trials used to get an averaged sample. The process is repeated M times by class, to get M averaged samples by class. The new dataset is used to train a classifier such that 1) the number of samples is equal for each class in the training set, and 2) the statistical properties of training and testing data remain comparable. It is worth noting that in practical scenarios, the number of averaged trials may not be defined *a priori*. However, this procedure can be followed for any value of *N*. Additionally, it does not imply any additional significant computational load, as the re-sampling is computationally inexpensive.

In this work, we used the training dataset as a new population to implement the re-sampling. We varied the number of repetitions (single trials) used to get a new averaged trial, with $N = \{2, 3, \ldots, 14, 15\}$, because 15 is the maximum number of repetitions available by character. Then, we repeated the process *M* times by class. Single trials were not used because re-sampling only allows obtaining repeated samples, decreasing the variability of the training samples in the mentioned case.

We tested a classifier trained with one of the following kinds of samples: unbalanced classes with single trials and balanced classes by re-sampling $M = \{1000, 2000, 3000\}$ averaged trials by class. The value of M is chosen according to the statistical significance obtained in the results, for all the classifying algorithms. For all cases, averaged trials were used as testing data.

Classifiers

In the literature, the classification problem involves identifying the row and the column that corresponds to a character of the speller. In the present study, the target of the classification is to determine whether a signal is P300 or not. For aiming to the goal, we implemented four classifying algorithms in the study: *stepwise linear discriminant analysis* (SWLDA), B*ayesian linear discriminant analysis* (BLDA), *support-vector machine with a linear kernel* (LSVM) and logistic regression (Log Reg). While LDA-based algorithms are generative classifiers, LSVM and Log Reg lies in the category of linear discriminative classifiers [25] [36]. The results presented are based on the test datasets, which are not seen by any of the implemented classifiers during the training procedure.

For discriminative classifiers, it is necessary to choose the value of a regularization factor *C*. A four-fold cross-validation process is implemented with the training dataset, to get the best value of *C*. The number of values tested for *C* was 25, all located between 0.01 and 1. After this procedure, the final classifier is trained using the whole training dataset and the chosen value of *C*. The process is repeated by each user and each type of training samples [36].

Stepwise LDA

The traditional approach implements a modified version of LDA as the classifier, where a stepwise regression is used before the classification task [23]. The classifier is known as SWLDA. Unlike other LDA-based classifiers, this classifier chooses the coefficients of the model regression iteratively, according to a statistical criterion [37]. As a result, the model obtained is more compact than the least-squares-based regression. The study implements the stepwise regression included in the Statistical and Machine Learning Toolbox for MATLAB®. Additional details of SWLDA can be found in [38].

Bayesian LDA

When the coefficients of the model implemented for LDA are chosen according to Bayesian criteria, an LDA classifier based on *Bayesian interpolation* (BLDA) is obtained. According to the literature, the algorithm gives better results than the ordinary LDA or, even, SWLDA [39] [40]. Like the SWLDA classifier, the coefficients are obtained by iteration. However, the statistical criteria for choosing corrections are based on the Bayes Rule and are not added or removed from the model [41]. The algorithm implemented in the study and further details of BLDA can be obtained from [39].

Linear SVM

Support-vector machine (SVM)-based classifiers have been implemented in several previous studies related to BCIs, including P300 spellers [14] [15] [20] [42] [43]. In this work, a linear kernel support-vector machine was implemented as the classifier with the LIBSVM Toolbox for MATLAB® [44], for each subject. The reader is encouraged to see [45] for a wide list of studies implementing SVM in BCIs.

Logistic Regression

Unlike the L-SVM, logistic regression-based classifiers have been implemented in fewer works related to BCIs [46] [47]. It is a member of the family of *log-linear models*, implemented in discriminative classifiers [25] [32]. In the present study, the classifier was implemented with the UGM Toolbox for MATLAB® [48], for each subject. Further details about Logistic Regression are found in [25].

Performance metrics

Accuracy

A common measure of performance used for classification is the accuracy. Accuracy is defined as a metric of the closeness between measured or predicted values and their corresponding true values [49]. A measure commonly used for the accuracy, for classification problems with M_c classes, is defined by using the trace of a confusion matrix H [29] as shown indicated in Equation (1):

$$p_0 = \frac{\text{trace}(H)}{N_s}, \tag{1}$$

Where where N_s is the total number of samples entered to the classifier, and trace(H) is the number of samples correctly classified. Accuracy varies from 0 to 1, where 1 gives represents a perfect classification.

Since the definition of accuracy is closely related to the definition of the binomial distribution $\mathcal{B}(N_s, p_0)$ with success probability p_0 and number of trials N_s, p_0 can be approximated to a normal distribution with the standard deviation defined in Equation (2):

$$S_e(p_0) = \sqrt{\frac{p_0(1-p_0)}{N_s}} \quad (2)$$

However, high accuracy does not always mean the classifier has high performance. This is true when the number of classes is highly unbalanced, as the classifier tends to be biased toward the class with the highest occurrence in the dataset. This is known as the *accuracy paradox* [50].

Cohen's kappa index

A commonly used measure of precision is the *Cohen's kappa index* κ [29] [51] [52]. It is an alternative way of measuring the predictive power of a classifier that relates the accuracy with the probability to classify by chance, as expressed by Equation (3):

$$\kappa = \frac{p_0 - p_e}{1 - p_e}. \quad (3)$$

The numerator is the difference between the accuracy and the expected probability to classify correctly by chance (p_e). The denominator is the difference between the maximum accuracy and p_e. Consequently, κ is defined as the rate of the difference between accuracy and p_e, and the maximum value of this difference is used to determine the difference. The possible values for κ are within the range of −1 to 1 [53]. A value of 1 means perfect classification, whereas a value of 0 indicates random assignments between true classes and the predicted values. Finally, a value of −1 indicates an opposite relationship between the real and predicted values. The expected probability p_e is defined in the Equation (4):

$$p_e = \frac{1}{N_s^2}\sum_{i=1}^{M_c} n_{i:}n_{:i}, \quad (4)$$

Where the sum of all elements for an the *i*-th row $n_{i:}$ and the sum of all elements for an the *i*-th column $n_{:i}$ are expressed in Equations (5) and (6) [30]:

$$p_e = \sum_{j=1}^{M_c} H_{ij} \quad (5)$$

$$p_e = \sum_{j=1}^{M_c} H_{ji} \quad (6)$$

The standard deviation of κ is calculated using Equation (7):

$$S_e(\kappa) = \frac{\sqrt{p_0 + p_e^2 - \sum_{i=1}^{M_c} n_{i:}n_{:i}(n_{i:}+n_{:i})/N_s^3}}{1 - p_e\sqrt{N_s}} \quad (7)$$

The standard error can be used to build confidence intervals and calculate statistical significance when accuracy or kappa values are compared.

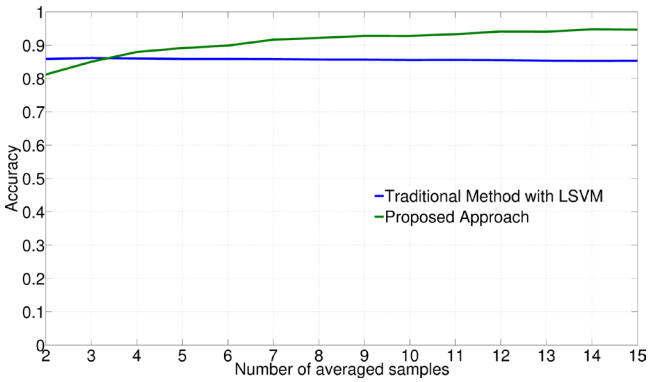
RESULTS

Results presented here refer to the average performance obtained by each classifier, in terms of the accuracy and Cohen's kappa index metrics. All metrics were obtained from the testing dataset of each subject.

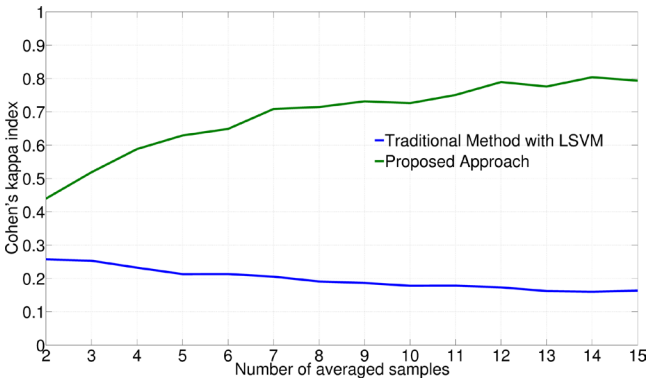
Number of bootstrapped samples

The statistical significance of differences among the numbers of bootstrapped samples for averaged training data was tested by a one-way randomized blocks ANOVA using a performance index and a classifier. ANOVA was chosen rather than a Student's t-test because ANOVA does not take into account the random effects of the number of averages and subjects, whereas ANOVA does. The number of training data (M) was taken as the design variable, and each performance index was the output variable. Subjects and the number of averaged samples by a testing trial were taken as randomized blocks. The numbers of samples used were M = {1,000, 2,000, 3,000}.

For Log Reg, the ANOVA test does not give any significant differences among the number of bootstrapped samples for neither accuracy nor Cohen's kappa index (accuracy: F = 1.72, p = 0.18; Cohen's kappa index: F =

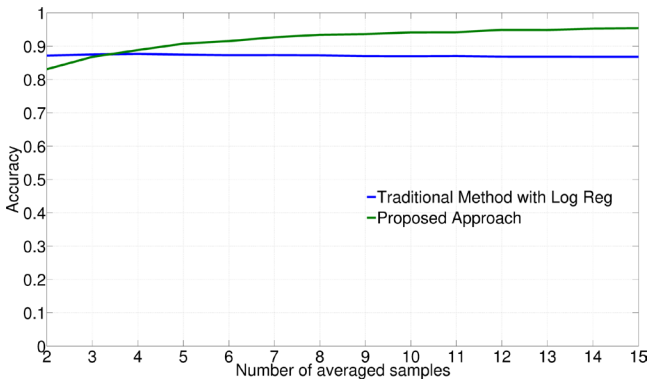


a) Accuracy

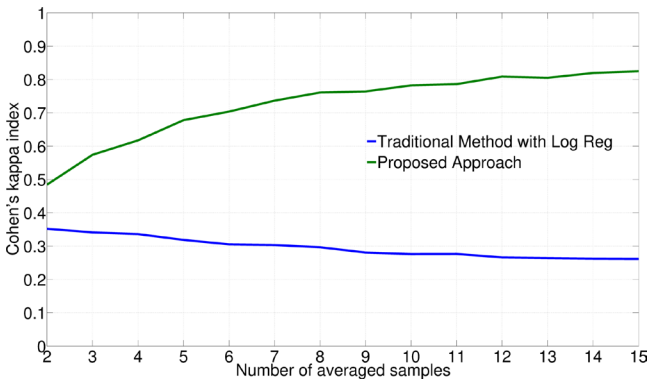


b) Cohen's kappa index

FIGURE 1. Averaged results of all subjects, for LSVM.



a) Accuracy



b) Cohen's kappa index

FIGURE 2. Averaged results of all subjects, for Log Reg.

0.51, p = 0.60). In the case of LSVM, there is no statistical difference in the number of samples (accuracy: F = 0.92, p = 0.40; Cohen's kappa index: F = 0.02, p = 0.98). Similar conclusions are obtained by analyzing the results of ANOVA tests for SWLDA (accuracy: F = 0.61, p = 0.55; Cohen's kappa index: F = 0.13, p = 0.88) and BLDA (accuracy: F = 0.67, p = 0.51; Cohen's kappa index: F = 0.39, p = 0.25). Although there is no significant difference, most of highest results were obtained with M = 2,000 averaged bootstrapped samples. Therefore, the chosen number of samples is M = 2,000 in the remaining sections of the paper.

Type of training samples

The statistical significance of differences among the previously described types of training data was tested by two procedures. First, a one-way randomized blocks ANOVA was performed by metric and classifier. The type of training data (traditional or proposed) was taken as the design variable. Other parameters are the same as those in the previous subsection. Then, a Student's t-test was performed individually for each subject, by pooling the metrics and their corresponding standard deviations. The purpose was to estimate the significance of the differences between the traditional method and the proposed one, both at a general level and by each subject.

Linear SVM

Figure 1 illustrates the average performance obtained by employing the LSVM classifier on each subject and type of training data. The ANOVA test gives significant differences for both metrics (accuracy: F = 216.92, p < 0.01; Cohen's kappa index: F = 1380.10, p < 0.01).

TABLE 1. Averaged metrics by subject, for LSVM.

Subject	Traditional Approach		Proposed Method	
	Accuracy	Kappa	Accuracy	Kappa
1	0,95 ± 0.11	0.78 ± 0.33	**0.98 ± 0.08***	**0.93 ± 0.33***
2	0.88 ± 0.12	0.43 ± 0.27	**0.95 ± 0.10***	**0.83 ± 0.33***
3	0.84 ± 0.13	0.01 ± 0.07	**0.90 ± 0.12***	**0.55 ± 0.33***
4	0.84 ± 0.13	0.06 ± 0.18	**0.95 ± 0.10***	**0.81 ± 0.30***
5	0.88 ± 0.12	0.36 ± 0.26	**0.97 ± 0.08***	**0.90 ± 0.30***
6	0.83 ± 0.13	0.00 ± 0.00	**0.84 ± 0.13**	**0.52 ± 0.27***
7	0.83 ± 0.13	0.00 ± 0.00	**0.87 ± 0.12***	**0.48 ± 0.27***
8	0.83 ± 0.13	0.00 ± 0.00	**0.87 ± 0.12***	**0.59 ± 0.28***
9	0.89 ± 0.12	0.43 ± 0.27	**0.94 ± 0.10***	**0.81 ± 0.29***
10	0.84 ± 0.12	0.06 ± 0.15	**0.94 ± 0.10***	**0.79 ± 0.30***
11	0.85 ± 0.13	0.18 ± 0.22	**0.94 ± 0.10***	**0.80 ± 0.29***
12	0.83 ± 0.13	0.00 ± 0.00	**0.90 ± 0.12***	**0.53 ± 0.28***
13	**0.88 ± 0.12**	0.44 ± 0.27	0.81 ± 0.13	**0.50 ± 0.26***
14	0.83 ± 0.13	0.02 ± 0.27	**0.90 ± 0.11***	**0.59 ± 0.29***
Average	0.86 ± 0.12	0.20 ± 0.19	**0.91 ± 0.11**	**0.69 ± 0.29**

*The difference is highly significant, with a Student's t-test ($p < 0.01$).
Number of samples: 372 for subject 1,504 for the rest.

TABLE 2. Averaged metrics by subject, for Log Reg.

Subject	Traditional Approach		Proposed Method	
	Accuracy	Kappa	Accuracy	Kappa
1	0.97 ± 0.09	0.88 ± 0.33	**0.98 ± 0.09**	**0.92 ± 0.33**
2	0.92 ± 0.11	0.65 ± 0.29	**0.96 ± 0.09***	**0.86 ± 0.30***
3	0.84 ± 0.13	0.03 ± 0.14	**0.93 ± 0.11***	**0.68 ± 0.29***
4	0.85 ± 0.13	0.15 ± 0.22	**0.95 ± 0.10***	**0.84 ± 0.30***
5	0.91 ± 0.11	0.56 ± 0.29	**0.97 ± 0.08***	**0.91 ± 0.30***
6	0.84 ± 0.13	0.05 ± 0.13	**0.86 ± 0.12***	**0.59 ± 0.27***
7	0.83 ± 0.13	0.00 ± 0.07	**0.88 ± 0.12***	**0.51 ± 0.28***
8	0.83 ± 0.13	0.00 ± 0.07	**0.88 ± 0.12***	**0.62 ± 0.28***
9	0.92 ± 0.11	0.64 ± 0.29	**0.95 ± 0.09***	**0.84 ± 0.29***
10	0.85 ± 0.13	0.15 ± 0.21	**0.95 ± 0.09***	**0.83 ± 0.30***
11	0.88 ± 0.12	0.41 ± 0.27	**0.95 ± 0.09***	**0.82 ± 0.30***
12	0.83 ± 0.13	0.00 ± 0.04	**0.90 ± 0.11***	**0.55 ± 0.29***
13	**0.91 ± 0.11**	**0.60 ± 0.29**	0.84 ± 0.13	0.57 ± 0.26
14	0.84 ± 0.13	0.03 ± 0.13	**0.91 ± 0.11***	**0.61 ± 0.29***
Average	0.87 ± 0.12	0.30 ± 0.22	**0.92 ± 0.11**	**0.72 ± 0.29**

*The difference is highly significant, with a Student's t-test ($p < 0.01$).
Number of samples: 372 for subject 1,504 for the rest.

According to the results, when the classifier is trained with 2,000 averaged trials by class, the performance is significantly better than that of the traditional approach.

Table 1 contrasts the average of results and pooled standard deviations obtained by subject, for accuracy and kappa. A Student's t-test was performed to get the statistical significance of the difference between the methods. Results indicate that the difference is highly significant ($p < 0.01$), for most of metrics and subjects. In most cases, the difference is in favor of the proposed method.

Logistic Regression

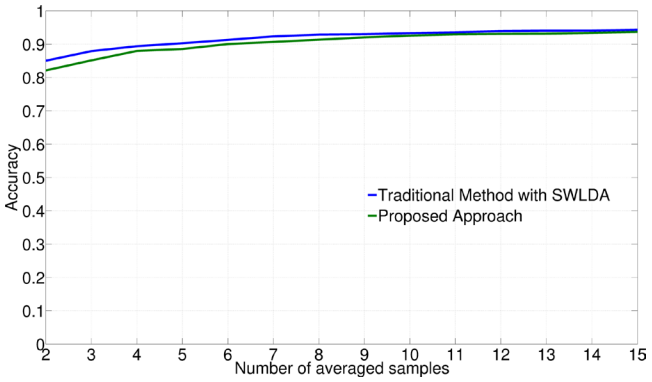
Figure 2 illustrates the average performance obtained by employing the Logistic Regression classifier on each subject and type of training data. The ANOVA test gives significant differences for both metrics (accuracy: $F = 215.10$, $p < 0.01$; Cohen's kappa index: $F = 843.29$, $p < 0.01$). Results indicate that the performance of the classifier is higher with the proposed method for training.

Table 2 compares the pooled results and standard deviations obtained by subject, for accuracy and kappa. Results of the Student's t-test, by subject and metric, indicate that the difference is highly significant ($p < 0.01$) for most of metrics and subjects, in favor of the proposed method. The consistency in the statistical analyses for Log Reg supports the improvement in the results with our method. It also applies to the LSVM results.

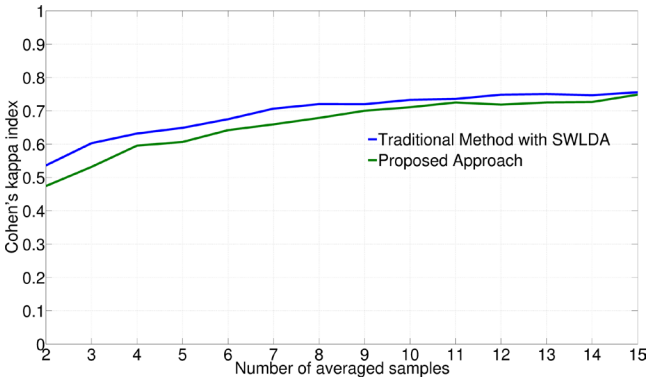
Stepwise and Bayesian LDA

Figure 3 illustrates the average performance obtained by employing the SWLDA classifier on each subject and type of training data.

The ANOVA test gives significant differences for both metrics (accuracy: $F = 28.15$, $p < 0.01$; Cohen's kappa index: $F = 20.86$, $p < 0.01$).However, when metrics are contrasted with a Student's t-test, results reveal that there is no statistical significance in most of the cases, as presented in Table 3.

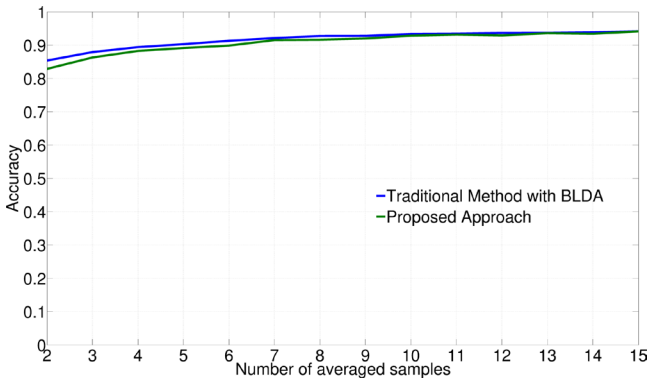


a) Accuracy

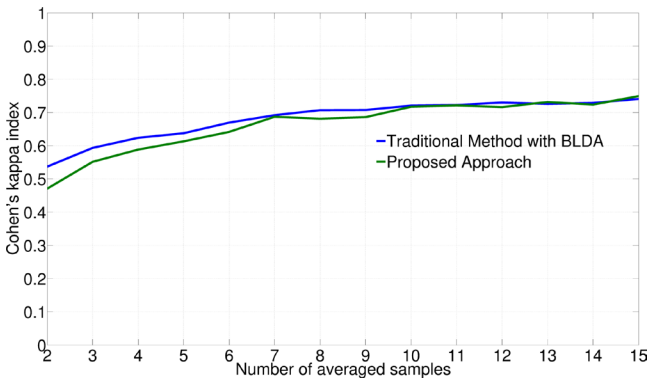


b) Cohen's kappa index

FIGURE 3. Averaged results of all subjects, for SWLDA .



a) Accuracy



b) Cohen's kappa index

FIGURE 4. Averaged results of all subjects, for BLDA.

Similar results were obtained for BLDA, as illustrated in Figure 4. Although the ANOVA test gives a significant difference between the traditional and the proposed methods (accuracy: $F = 13.21$, $p < 0.01$; Cohen's kappa index: $F = 6.21$, $p = 0.013$), the individual Student's t-tests do not reject the null hypothesis of equality of metrics, as presented in Table 4.

DISCUSSION

Results of LSVM and logistic regression are similar. When we trained the classifiers with the traditional approach, Cohen's kappa index decreased as the number of averaged samples was increased in the testing samples. The observed decrement of kappa is due to the probability of classifying by chance p_e, as defined in Equation (3). The value of p_e increases with the increase in the number of averaged samples. Meanwhile, the accuracy only has little changes when the number of averaged samples by trial is increased. As a consequence, p_e is closer to the accuracy as the number of averaged samples is greater, so kappa is decreased.

Tables 3 and 4 indicate that subjects 3, 4, 6, 7, 8, 12 and 14 have averaged kappa values close to or lower than 0.05. It is a strong indicator of the presence of the *accuracy paradox* in these cases. It is an indication that discriminative classifiers try to label most of the samples as non-P300 targets in the traditional approach because in the P300 speller, there are more non-P300 samples than P300 ones. In addition, since each discriminative classifier is trained in single trials, it learns features that averaged trials do not have, because of the different statistical properties of single and averaged data. As a consequence, most of non-P300 features will be learned in this case.

TABLE 3. Averaged metrics by subject, for SWLDA.

Subject	Traditional Approach		Proposed Method	
	Accuracy	Kappa	Accuracy	Kappa
1	0.98 ± 0.08	**0.95 ± 0.33**	0.98 ± 0.08	0.94 ± 0.33
2	**0.96 ± 0.09**	**0.88 ± 0.30**	0.95 ± 0.10	0.84 ± 0.29
3	**0.86 ± 0.12**	0.26 ± 0.25	0.87 ± 0.12	**0.36 ± 0.26***
4	**0.95 ± 0.10**	**0.83 ± 0.30**	0.94 ± 0.10	0.79 ± 0.30
5	0.98 ± 0.07	**0.94 ± 0.31**	0.98 ± 0.08	0.92 ± 0.30
6	**0.88 ± 0.12**	**0.63 ± 0.28**	0.84 ± 0.13	0.53 ± 0.27
7	**0.89 ± 0.12**	**0.54 ± 0.28**	0.86 ± 0.12	0.48 ± 0.27
8	**0.90 ± 0.11**	**0.68 ± 0.28**	0.87 ± 0.12	0.60 ± 0.28
9	**0.94 ± 0.10**	**0.82 ± 0.29**	0.93 ± 0.10	0.80 ± 0.29
10	0.94 ± 0.10	0.78 ± 0.30	0.94 ± 0.10	0.78 ± 0.30
11	**0.95 ± 0.10**	**0.82 ± 0.30**	0.94 ± 0.10	0.79 ± 0.29
12	**0.90 ± 0.12**	**0.50 ± 0.28**	0.89 ± 0.12	0.48 ± 0.28
13	**0.83 ± 0.13**	**0.54 ± 0.26**	0.80 ± 0.13	0.48 ± 0.26
14	**0.90 ± 0.12**	**0.57 ± 0.29**	0.87 ± 0.12	0.47 ± 0.28
Average	**0.92 ± 0.11**	**0.69 ± 0.29**	0.90 ± 0.11	0.66 ± 0.29

*The difference is highly significant, with a Student's t-test ($p < 0.01$).
Number of samples: 372 for subject 1,504 for the rest.

TABLE 4. Averaged metrics by subject, for BLDA.

Subject	Traditional Approach		Proposed Method	
	Accuracy	Kappa	Accuracy	Kappa
1	**0.99 ± 0.08**	**0.95 ± 0.33**	0.98 ± 0.08	0.94 ± 0.33
2	**0.96 ± 0.09**	**0.87 ± 0.30**	0.95 ± 0.09	0.85 ± 0.30
3	0.86 ± 0.12	0.30 ± 0.25	0.86 ± 0.12	**0.33 ± 0.26**
4	0.95 ± 0.10	**0.82 ± 0.30**	0.95 ± 0.10	0.79 ± 0.30
5	0.98 ± 0.07	**0.94 ± 0.31**	0.98 ± 0.10	0.92 ± 0.30
6	**0.86 ± 0.12**	**0.58 ± 0.27**	0.84 ± 0.13	0.53 ± 0.27
7	**0.88 ± 0.12**	0.48 ± 0.28	0.87 ± 0.12	**0.49 ± 0.28**
8	**0.90 ± 0.11**	**0.67 ± 0.28**	0.88 ± 0.12	0.62 ± 0.28
9	**0.95 ± 0.09**	**0.86 ± 0.30**	0.94 ± 0.10	0.82 ± 0.29
10	**0.95 ± 0.10**	**0.79 ± 0.30**	0.94 ± 0.10	0.78 ± 0.30
11	**0.95 ± 0.09**	**0.83 ± 0.30**	0.94 ± 0.10	0.80 ± 0.29
12	0.89 ± 0.12	0.43 ± 0.27	0.89 ± 0.12	**0.45 ± 0.27**
13	**0.85 ± 0.12**	**0.58 ± 0.27**	0.82 ± 0.13	0.52 ± 0.26
14	**0.89 ± 0.12**	0.46 ± 0.28	0.88 ± 0.12	0.46 ± 0.28
Average	**0.92 ± 0.29**	**0.68 ± 0.11**	0.91 ± 0.29	0.66 ± 0.11

*The difference is highly significant, with a Student's t-test ($p < 0.01$).
Number of samples: 372 for subject 1,504 for the rest.

By contrast, when we trained the classifier with the proposed method, the performance improved significantly. The improvement of kappa indicates that the classifiers learn features from both P300 and non-P300 classes. This is because the accuracy gets greater values than p_e when the number of averaged samples by trial is increased. Consequently, the accuracy, and thus kappa, will be higher, as seen in Figures 1 and 2. The reasons for this are balanced data, similar statistical properties for training and test samples, and more statistical variation by class due to an increase in sample size. Here, it is necessary to remark that the inconvenience of using unbalanced classes with single trials for training discriminative classifiers is due to the difference in statistical properties of the data used for testing the classifier. Again, the advantage of training with re-sampled and averaged samples is statistically significant.

By contrast, although results of stepwise and Bayesian LDA were similar between them, they are different from the discriminative classifiers. ANOVA tests give significant differences in the methods, whereas the Student's t-test does not reject the statistical equality of the results, as presented in Tables 5 and 6. The discrepancy of the statistics is due to the origin of the standard deviation in each test. The Student's test employs weighted pooling of the variances, whereas ANOVA uses the *mean square* of the error from the data. Thus, the standard deviation by subject lies between 0,07 and 0,13 for accuracy and between 0,25 and 0,33 for kappa, the mean square errors are around 0,0006 for accuracy and 0,006 for kappa. This means that the differences of magnitude between standard deviations are around 183 for accuracy and 48 for Cohen's kappa index. Therefore, with the same averaged metrics, both types of tests give different results: ANOVA sees a gap between the levels of the design variable, whereas the Student's t-test gives small values of the statistics.

Another issue worth considering is the nature of the LDA-based classifiers. They try to fit the data to a set of Gaussian models, with a mean by class and a common covariance matrix [25]. When new data are presented to the

classifier, they are compared with each model. Later, a class is assigned to the data when the highest score or probability value is obtained from the corresponding model of the set. This score or probability comes from the distance between the data and each mean. In our study, both classifiers map the data to a score value, according to a model of regression before the generation of Gaussian models. This means that the models are also scalar rather than multivariate, unlike discriminative classifiers, where the mapping to the class is direct [25]. Consequently, discriminative models are more affected by the statistical nature of the data. This is reflected in the difference of the results between generative and discriminative classifiers.

CONCLUSIONS

In this study, a bootstrapping method is presented to solve two important problems in the P300 speller. The method generates a new training set by re-sampling with replacement from the original set, reaching two important goals at the same time.

First, the number of trials across classes is balanced. It avoids dropping data in the process, as suggested in other approaches [13] [14] [15] [16] [17], which prevents a possible bias in the classification results.

Second, the statistical properties of the training data are made equivalent between the training and the test sets. This is achieved when the number of averaged trials for each instance in training equals the number of averaged samples during testing.

Unbalanced classes and the difference in statistical properties are considerable issues present in the state-of-the-art implementations of the P300 classification task.

Results presented here indicate that the proposed method improves significantly the detection of P300 and non-P300 classes in linear discriminative classifiers, by dealing with the aforementioned issues.

REFERENCES

[1] Farwell LAA, Donchin E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalogr Clin Neurophysiol 1988;70:510-23. doi:10.1016/0013-4694(88)90149-6

[2] Blankertz B, Krauledat M, Dornhege G, Williamson J, Murray-Smith R, Müller K-R. A Note on Brain Actuated Spelling with the Berlin Brain-Computer Interface. Int Conf Univers Access Human-Computer Interact 2007;2007:759-68. doi:10.1007/978-3-540-73281-5_83

[3] Treder MS, Blankertz B. (C)overt attention and visual speller design in an ERP-based brain-computer interface. Behav Brain Funct 2010;6:28. doi:10.1186/1744-9081-6-28

[4] Kaufmann T, Schulz SM, Grünzinger C, Kübler a. Flashing characters with famous faces improves ERP-based brain-computer interface performance. J Neural Eng 2011;8:056016. doi:10.1088/1741-2560/8/5/056016

[5] Speier W, Deshpande A, Cui L, Chandravadia N, Roberts D, Pouratian N. A comparison of stimulus types in online classification of the P300 speller using language models. PLoS One 2017;12:e0175382. doi:10.1371/journal.pone.0175382

[6] Polich J. Updating P300: An integrative theory of P3a and P3b. Clin Neurophysiol 2007;118:2128-48. doi:10.1016/j.clinph.2007.04.019

[7] Speier W, Arnold C, Lu J, Deshpande A, Pouratian N. Integrating Language Information With a Hidden Markov Model to Improve Communication Rate in the P300 Speller. IEEE Trans Neural Syst Rehabil Eng 2014;22:678-84. doi:10.1109/TNSRE.2014.2300091

[8] Chaurasiya RK, Londhe ND, Ghosh S. An efficient P300 speller system for Brain-Computer Interface. 2015 Int. Conf. Signal Process. Comput. Control, IEEE; 2015, p. 57-62. doi:10.1109/ISPCC.2015.7374998

[9] De Vos M, Kroesen M, Emkes R, Debener S. P300 speller BCI with a mobile EEG system: comparison to a traditional amplifier. J Neural Eng 2014;11:036008. doi:10.1088/1741-2560/11/3/036008

[10] Xiaofeng Shi, Guoqiang Xu, Furao Shen, Jinxi Zhao. Solving the data imbalance problem of P300 detection via Random Under-Sampling Bagging SVMs. 2015 Int. Jt. Conf. Neural Networks, vol. 2015-Septe, IEEE; 2015, p. 1-5. doi:10.1109/IJCNN.2015.7280834

[11] Tibon R, Levy DA. Striking a balance: Analyzing unbalanced event-related potential data. Front Psychol 2015;6:1-4. doi:10.3389/fpsyg.2015.00555

[12] Valverde-Albacete FJ, Peláez-Moreno C. 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. PLoS One 2014;9. doi:10.1371/journal.pone.0084217

[13] Xu N, Gao X, Hong B, Miao X, Gao S, Yang F. BCI competition 2003 - Data set IIb: Enhancing P300 wave detection using ICA-based subspace projections for BCI applications. IEEE Trans Biomed Eng 2004;51:1067-72. doi:10.1109/TBME.2004.826699

[14] Kaper M, Meinicke P, Grossekathoefer U, Lingner T, Ritter H. BCI Competition 2003—Data Set IIb : Support Vector Machines for the P300 Speller Paradigm. Ieee Trans Biomed Eng 2004;51:1073-6. doi:10.1109/TBME.2004.826698

[15] Rakotomamonjy A, Guigue V, Mallet G, Alvarado V. Ensemble of SVMs for improving brain computer interface P300 speller performances. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2005;3696 LNCS:45-50. doi:10.1007/11550822_8

[16] Meinicke P, Kaper M, Hoppe F, Heumann M, Ritter H. Improving Transfer Rates in Brain Computer Interfacing: A Case Study. Adv. Neural Inf. Process. Syst., 2003, p. 1107-14.

[17] Hoffmann U, Garcia G, Vesin J-MJM, Diserens K, Ebrahimi T, Diserenst K, et al. A Boosting Approach to P300 Detection with Application to Brain-Computer Interfaces. Conf. Proceedings. 2nd Int. IEEE EMBS Conf. Neural Eng. 2005., vol. 2005, IEEE; 2005, p. 97-100. doi:10.1109/CNE.2005.1419562

[18] Farquhar J, Hill NJ. Interactions between pre-processing and classification methods for event-related-potential classification: Best-practice guidelines for brain-computer interfacing. Neuroinformatics 2013;11:175-92. doi:10.1007/s12021-012-9171-0

[19] Bostanov V. BCI competition 2003 - Data sets Ib and IIb: Feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. IEEE Trans Biomed Eng 2004;51:1057-61. doi:10.1109/TBME.2004.826702

[20] Rakotomamonjy A, Guigue V. BCI competition III: Dataset II-ensemble of SVMs for BCI P300 speller. IEEE Trans Biomed Eng 2008;55:1147-54. doi:10.1109/TBME.2008.915728

[21] Yang Liu, Zongtan Zhou, Dewen Hu, Guohua Dong. T-weighted Approach for Neural Information Processing in P300 based Brain-Computer Interface. 2005 Int. Conf. Neural Networks Brain, vol. 3, IEEE; 2005, p. 1535-9. doi:10.1109/ICNNB.2005.1614924

[22] Schalk G, McFarland DJ, Hinterberger T, Birbaumer N, Wolpaw JR. BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. IEEE Trans Biomed Eng 2004;51:1034-43. doi:10.1109/TBME.2004.827072

[23] Krusienski DJ, Sellers EW, McFarland DJ, Vaughan TM, Wolpaw JR. Toward enhanced P300 speller performance. J Neurosci Methods 2008;167:15-21. doi:10.1016/j.jneumeth.2007.07.017

[24] Krusienski DJ, Sellers EW, Cabestaing F, Bayoudh S, McFarland DJ, Vaughan TM, et al. A comparison of classification techniques for the P300 Speller. J Neural Eng 2006;3:299-305. doi:10.1088/1741-2560/3/4/007

[25] Bishop CM. Pattern Recognition and Machine Learning. Springer-Verlag New York; 2013.

[26] Delgado Saa JF, Pesters A de, McFarland D, Çetin M. Word-level language modeling for P300 spellers based on discriminative graphical models. J Neural Eng 2015;12:026007. doi:10.1088/1741-2560/12/2/026007

[27] Orhan U, Erdogmus D, Roark B, Purwar S, Hild KE, Oken B, et al. Fusion with language models improves spelling accuracy for ERP-based brain computer interface spellers. 2011 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., vol. 100, IEEE; 2011, p. 5774-7. doi:10.1109/IEMBS.2011.6091429

[28] Martens SMM, Mooij JM, Hill NJ, Farquhar J, Schölkopf B. A Graphical Model Framework for Decoding in the Visual ERP-Based BCI Speller. Neural Comput 2011;23:160-82. doi:10.1162/NECO_a_00066

[29] Schlögl A, Lee F, Bischof H, Pfurtscheller G. Characterization of four-class motor imagery EEG data for the BCI-competition 2005. J Neural Eng 2005;2:L14-22. doi:10.1088/1741-2560/2/4/L02

[30] Schögl A, Kronegg J, Huggins JE, Mason SG. Evaluation Criteria for BCI Research. Towar Brain-Computer Interfacing 2007:327-42.

[31] Delgado Saa JF, Çetin M. Hidden conditional random fields for classification of imaginary motor tasks from EEG data. Eur. Signal Process. Conf., vol. 19, 2011, p. 1377-81.

[32] Society AE. American Electroencephalographic Society Guidelines for Standard Electrode Position Nomenclature. J-Clin Neurophysiol 1991;8:200-2. doi:10.1097/00004691-199104000-00007

[33] Batterink L, Karns CM, Neville H. Dissociable mechanisms supporting awareness: The P300 and gamma in a linguistic attentional blink task. Cereb Cortex 2012;22:2733-44. doi:10.1093/cercor/bhr346

[34] Thompson SK. Sampling. 3rd ed. Wiley; 2012.

[35] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Springer-Verlag New York; 2009. doi:10.1007/b94608

[36] Mitchell TM. Machine Learning. WCB / McGraw - Hill; 1997.

[37] Hocking RR. A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression. Biometrics 1976;32:1. doi:10.2307/2529336

[38] Draper NR, Smith H. Applied Regression Analysis. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2014. doi:10.1002/9781118625590

[39] Hoffmann U, Vesin JM, Ebrahimi T, Diserens K. An efficient P300-based brain-computer interface for disabled subjects. J Neurosci Methods 2008. doi:10.1016/j.jneumeth.2007.03.005

[40] Manyakov N V., Chumerin N, Combaz A, Van Hulle MM. Comparison of Classification Methods for P300 Brain-Computer Interface on Disabled Subjects. Comput Intell Neurosci 2011;2011:1-12. doi:10.1155/2011/519868

[41] MacKay DJC. Bayesian Interpolation. Neural Comput 1992. doi:10.1162/neco.1992.4.3.415.

[42] Thulasidas M, Guan C, Wu J. Robust classification of EEG signal for brain-computer interface. IEEE Trans Neural Syst Rehabil Eng 2006;14:24-9. doi:10.1109/TNSRE.2005.862695.

[43] Salvaris M, Sepulveda F. Visual modifications on the P300 speller BCI paradigm. J Neural Eng 2009;6:046011. doi:10.1088/1741-2560/6/4/046011

[44] Chang C-C, Lin C-J. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2011;2:27:1-27:27. doi:10.1145/1961189.1961199

[45] Bashashati A, Fatourechi M, Ward RK, Birch GE. A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. J Neural Eng 2007;4:R32-57. doi:10.1088/1741-2560/4/2/R03

[46] Tomioka R, Aihara K, Müller K-R. Logistic regression for single trial EEG classification. Analysis 2007;19:1377-84.

[47] Prasad PD, Halahalli HN, John JP, Majumdar KK. Single-Trial EEG Classification Using Logistic Regression Based on Ensemble Synchronization. IEEE J Biomed Heal Informatics 2014;18:1074-80. doi:10.1109/JBHI.2013.2289741

[48] Schmidt M. UGM: A Matlab toolbox for probabilistic undirected graphical models 2007.

[49] JCGM. JCGM 200 : 2008 International vocabulary of metrology — Basic and general concepts and associated terms (VIM) Vocabulaire international de métrologie — Concepts fondamentaux et généraux et termes associés (VIM). Int Organ Stand Geneva ISBN 2008;3:104. doi:10.1016/0263-2241(85)90006-5

[50] Zhu X. Knowledge Discovery and Data Mining: Challenges and Realities: Challenges and Realities. Information Science Reference; 2007.

[51] Warrens MJ. Inequalities between multi-rater kappas. Adv Data Anal Classif 2010;4:271-86. doi:10.1007/s11634-010-0073-4

[52] Burn CC, Weir AAS. Using prevalence indices to aid interpretation and comparison of agreement ratings between two or more observers. Vet J 2011;188:166-70. doi:10.1016/j.tvjl.2010.04.021

[53] McHugh ML. Lessons in biostatistics Interrater reliability : the kappa statistic. Biochem Medica 2012;22:276-82. doi:10.11613/BM.2012.031