2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

# Ensemble Learning Approach for Subject-Independent P300 Speller

Ayana Mussabayeva, Prashant Kumar Jamwal and Muhammad Tahir Akhtar

***Abstract*— P300 speller is a brain-computer interface (BCI) speller system, used for enabling human with different paralyzing disorders, such as amyotrophic lateral sclerosis (ALS), to communicate with the outer world by processing electroencephalography (EEG) signals. Different people have different latency and amplitude of the P300 event-related potential (ERP) component, which is used as the main feature for detecting the target character. In order to achieve robust results for different subjects using generic training (GT), the ensemble learning classifiers are proposed based on linear discriminant analysis (LDA), support vector machine (SVM), $k$-nearest neighbors ($k$NN), and convolutional neural network (CNN). The proposed models are trained using data from healthy subjects and tested on both healthy subjects and ALS patients. The results show that the fusion of LDA, $k$NN and SVM provides the most accurate results, achieving the accuracy of 99% for healthy subjects and about 85% for ALS patients.**

## I. INTRODUCTION

Brain-computer interface (BCI) is a closed-loop system controlled by human brain signals. Most of BCI systems use electroencephalography (EEG) for brain activity detection, as it is relatively cheap, portable and non-invasive neuroimaging method [1]. EEG-based BCI systems can use three different paradigms, which are: event-related potential (ERP), steady-state visual evoked potential (SSVEP) and motor imagery (MI). ERP paradigm is classically used in speller systems, such as P300 speller, introduced in 1988 [2].

ERP paradigm states that a positive deflection can be detected at around 300 ms after the target stimuli. This positive deflection is called P300 component [3]. This concept is also known as the oddball paradigm [4]. The graphical user interface of an ERP-based speller system usually looks like a table of symbols, in which rows and columns of symbols are flashing randomly, while the subject is staring at a particular chosen character. The classical English-oriented graphical user interface (GUI) of P300 speller is represented as a $6 \times 6$ matrix of symbols.

The main problem of the ERP-based speller systems is that a P300 component is not necessarily detected at 300 ms after the visual stimuli, as its latency varies from one human to another, so it is a subject-dependent paradigm. Classical P300 speller requires training phase for each user, which takes a lot of time. The objective of this work is to design a robust, subject-independent classifier for P300 speller, which will not only be accurate enough for any user, but also can process data faster. For achieving this aim, the proposed ensemble models are trained using generic training (GT) approach [5].

Section II overviews the data preprocessing steps and training approach, followed by the description of the chosen classifiers and their further fusion in ensemble models. Section III presents the results obtained for healthy subjects and ALS patients. Finally, Section IV contains the conclusions.

*This work was partly supported by the Faculty Development Competitive Research Grants Program of Nazarbayev University under the grant number 110119FD4525 and partly supported under the grant number 091019CRP2116.

Authors are with the Department of Electrical and Computer Engineering, School of Engineering and Digital Sciences, Nazarbayev University, Kabanbay Batyr Ave. 53, Nur-Sultan 010000 Kazakhstan (Emails:{ayana.mussabayeva, prashant.jamwal, muhammad.akhtar}@nu.edu.kz, akhtar@ieee.org)

## II. METHODOLOGY

### A. Data Preprocessing

The models are trained using data of eight healthy subjects from Akimpech P300 dataset [6]. Testing data consists of another five healthy subjects from the above mentioned dataset and five subjects, suffering amyotrophic lateral sclerosis (ALS) from BCI Horizon 2020 ALS patients P300 dataset [7].

The EEG values vectors $X$ are marked with $y$ label. For EEG vectors containing target P300 peak $y = 1$, while for non-target flashings $y = -1$. In order to have the same number of electrodes, two electrodes (C3, C4) were removed from Akimpech data, resulting 8-channel (Fz, Cz, Pz, P3, P4, P07, P08, Oz) data from both datasets.

EEG signal has been band-passed using 0.1-30 Hz frequency range, as the frequencies higher than 30 Hz or $\gamma$-band of EEG signal is not necessary to be considered in the oddball paradigm.

To reduce the redundancy of the data, only the regions starting from -100 ms before the flashing, ending with the 700 ms after the flashing are considered. This slight change can improve the running time for ensemble models, which require more computational resources than standalone classifiers. Moreover, the dataset was balanced by removing redundant non-target EEG vectors. As a result, the dataset comprised of 60% of the non-target class and 40% of the target class data.

### B. Generic Training

There are two different training approaches used for P300 speller, which are subject-specific training (SST) and generic training (GT).

SST assumes training for each user separately and it is used in most classical P300 speller systems. The training dataset in this case is collected from a single subject. After the training phase, the system can be tested using the same human's brain signals.

GT approach merges the data from different subjects into a single training dataset. The generically trained model can further be used for new subjects without training again. GT approach represented better results in the previous experiments [5] and is used further for the proposed ensemble voters training.

### C. Classifiers

Linear-discriminant analysis (LDA) is one of the most useful classifiers in BCI research, as it is computationally efficient and provides robust results. Despite the fact that this trivial algorithm was proposed in 1980-s [8], it is still one of the most useful methods applied for classification of various data, including multi-channel EEG time-series. For instance, when using EEG and electrooculography (EOG) combined together for detecting user's response, LDA achieves more than 97% of accuracy [9].

Support vector machine (SVM) classifier uses kernel functions to transform the data from one dimension to another and then constructs an optimal hyperplane to separate data classes. The optimal hyperplane is found by solving quadratic optimization problem. For solving this problem usually derivative tests are applied, such as Karush-Kuhn-Tucker (KKT) Conditions[10]. Classical Fisher LDA was outperformed by SVM classifier in discriminating early vascular dementia patients by EEG data [11].

In this paper, the SVM classifier uses hyperbolic tangent ($\tanh$) as a kernel function for ensemble classifier, computed as

$$\tanh(X_i) = \frac{\exp(X_i) - \exp(-X_i)}{\exp(X_i) + \exp(-X_i)}, \tag{1}$$

where $X_i$ denotes the EEG feature vector.

One of the most simple and efficient classifiers is $k$-nearest neighbors classifier ($k$NN), which identifies the distance between the classified data point and its $k$ neighbors. The distance metric used for $k$NN classifier can be cosine distance, Euclidean distance, Manhattan distance and etc. By trying different hyperparameters, it turned out that the best result for this case is provided by Manhattan distance, computed as

$$d(X_j, X_i) = \sum_{m=0}^{m=M-1} |x_{jm} - x_{im}|, \tag{2}$$

where the classified EEG vector of length $M$ is compared to its $k$ neighbors. Here $X_i$ denotes the $i^{th}$ neighbor's vector and $x_{im}$ denotes the $m^{th}$ data point of this vector. The best number of $k$ was evaluated using grid search (GS). The classifier reached a promising result of F-measure = 98.6% for $k = 3$.

CNN became quite popular in BCI research over the last decade. The reason for that is its ability to process multi-channel EEG data without some additional dimensionality reduction preprocessing. CNN can provide accurate results for P300 component identification. For example, CNN model with residual block achieved 96.77% accuracy for one subject and 93.3% for another [12]. CNN is frequently used in ensemble models, combining several classifiers [13]. The ensemble voting classifier comprised of two CNNs achieved 96.5% of accuracy, the same accuracy was obtained by ensemble SVM [14].

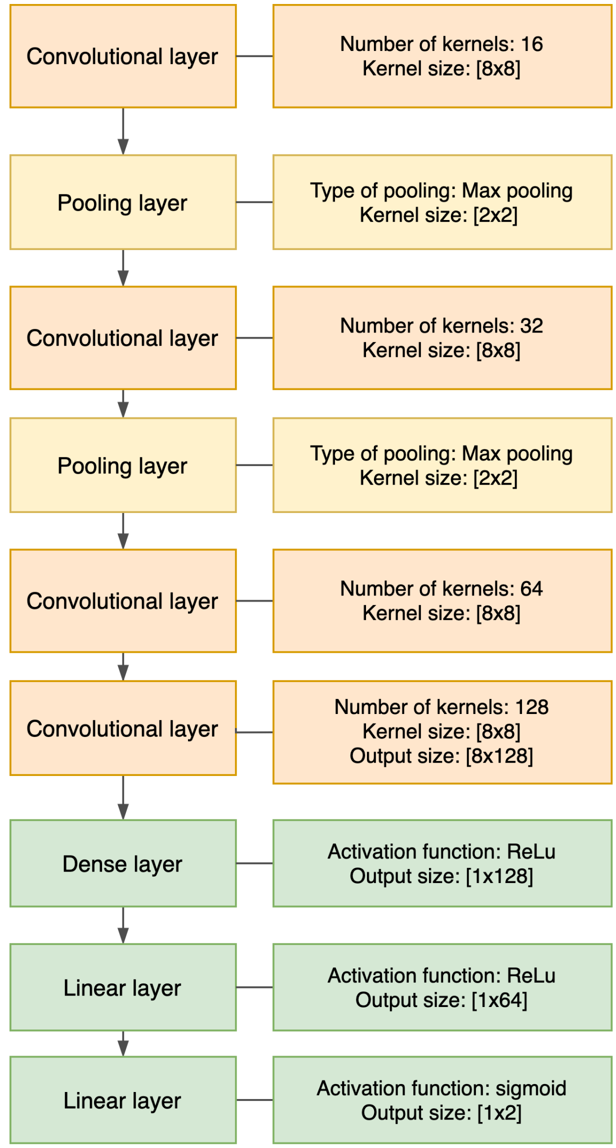


Fig. 1: Architecture of the CNN used: features are extracted using convolutional layer and pooling layers, followed by linear layers

The CNN architecture used for multi-channel EEG classification is presented in Fig. 1. For the activation function of convolutional layers it was decided to use rectified linear unit (ReLU) function, computed as

$$\mathrm{ReLU}(X_i) = \max(0, X_i). \tag{3}$$

The output of the last layer is a vector of length of two, representing the probability of the input EEG data containing target P300 component $P(X_i, y = 1)$ or non-target component $P(X_i, y = -1)$.

Multi-channel EEG signal is passed directly to CNN classifier, while it is required to average the signal over the channels before passing it to LDA, SVM and $k$NN classifiers.

### D. Ensemble Classifiers

Ensemble learning is a technique of combining several models for achieving more stable results. Recently, ensemble learning became a popular choice for brain signal's features classification. The popular choice is to ensemble several CNN classifiers [14] or a number of SVM classifiers with different hyperparameters [15].

Ensemble learning represent stable results for P300 component classification, however it requires much more computational time to be trained. That is why the proposed methodology is designed for GT approach, when the model is trained on a merged dataset from different subjects and does not require training for a new user.

The classification results of the ensemble averaged voting models are computed as

$$P_{avg}(X|y=1) = \frac{\sum_{i=1}^{N} P_i(X|y=1)}{N}, \qquad (4)$$

where $P_i(X|y=1)$ is the $i^{th}$ classifier's prediction of EEG vector $X$ containing target P300 component. $N$ is the number of classifiers in the ensemble voter.

It is assumed that weighted voting can be more efficient than simple ensemble averaging. However it is not always true, for instance, the weighted model based on CNN, SVM and stepwise LDA did not improve the accuracy of the classification [16].

In the proposed weighted voter W-LDA-SVM-$k$NN each result of the inner classifiers is multiplied by the weight $w_i$, resulting

$$P_w(X|y=1) = \frac{\sum_{i=1}^{N} w_i P_i(X|y=1)}{\sum_{i=1}^{N} w_i}, \qquad (5)$$

where weights $w_i$ can be found using random search (RS). In this work classical fixed step size RS [17] is used, however some more optimized methods such as adaptive step size RS [18] may also be applied.

## III. RESULTS

### A. Performance Evaluation

In order to evaluate the performance of each classifier, the number of true positive ($TP$), true negative ($TN$), false positive ($FP$) and false negative ($FN$) predictions are calculated. The accuracy is computed as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \qquad (6)$$

However, as the dataset is not perfectly balanced, there might be the case when the classifier identifies only non-target EEG signals, but fails to classify the target class. In order to check, whether the target class is correctly recognized and the number of $FN$ is low, recall is calculated as

$$\text{Recall} = \frac{TP}{TP + FN}. \qquad (7)$$

Precision value indicates an EEG signal labelled as positive (target response) is positive indeed and is computed as

$$\text{Precision} = \frac{TP}{TP + FP}. \qquad (8)$$

The most efficient metric for unbalanced datasets is F-measure, which is calculated as

$$\text{F-measure} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}. \qquad (9)$$

### B. Testing on Healthy Subjects

The classification models are trained on the 8-channel data of eight healthy subjects and further tested on five healthy subjects. Two baseline classifiers are trained and tested on the same data. The classical gradient boosting classifier [19] and the extreme gradient boosting or XGBoost [20] are chosen as baseline classifiers, as they provide high performance and efficiency for P300 classification [21].

The weights of the W-LDA-SVM-$k$NN model were found using RS. Searching for the weights took 40.89 s for the data from eight subjects. The obtained weights are $w_1 = 0.16$ for LDA classifier's output, $w_2 = 0.79$ for SVM output and $w_3 = 0.21$ for $k$NN model. The running time elapsed is 3.86 s, excluding the weights search (see Table I).

It is seen that the proposed classifiers provide good results, excepting for the model which uses CNN. LDA-SVM-$k$NN-CNN ensemble voter turned out to be computationally ineffective due to the complex structure of the neural network. The fastest model proposed is LDA-$k$NN fusion, which takes only 0.71 s to train for eight subjects. This can be explained by the fact that LDA is an efficient choice for EEG classification with low computational complexity and $k$NN is an instance based algorithm which just computes the distance for only $k = 3$ neighbors. The weighted ensemble model does not show any performance improvement compared to simple averaged LDA-SVM-$k$NN model. However, both models provide the best result of F-measure achieving 99.93%.

The proposed classifiers provide better results than the baseline classifiers in terms of computational complexity. This is explained by the fact that the gradient boosting nests decision trees one after another to achieve the necessary performance, requiring more time for computation.

### C. Testing on ALS Patients

The Table II represents the simulation results obtained while testing on five ALS patients data. The overall performance has decreased compared to the results obtained when using data of healthy subjects only. Still, the proposed methods do not fail to work with ALS patients, which means that the classifiers are subject-independent.

The baseline classifiers are performing better, reaching more than 85% of F-measure. The weighted voter classifier W-LDA-SVM-$k$NN outperforms gradient boosting and achieves the best performance metrics among the proposed classifiers in this case. Hence, it can be assumed that SVM classifier, which has the most value in the weighted voter,

TABLE I: Testing results using healthy subjects data

| Model | Accuracy (%) | Recall(%) | F-measure (%) | Time elapsed(s) |
|---|---|---|---|---|
| Gradient Boosting | 98.27 | 80.45 | 81.99 | 16.25 |
| XGBoost | 99.90 | 97.00 | 97.90 | 4.89 |
| LDA-kNN | 99.91 | 99.91 | 99.00 | 0.71 |
| LDA-SVM-kNN | 99.93 | 99.20 | 99.12 | 3.81 |
| LDA-SVM-kNN-CNN | 88.20 | 80.56 | 81.03 | 2687.55 |
| W-LDA-SVM-kNN | 99.93 | 99.20 | 99.12 | general time: 3.86<br>weights evaluation: 40.89 |

TABLE II: Testing results using ALS patients data

| Model | Accuracy (%) | Recall(%) | F-measure (%) |
|---|---|---|---|
| Gradient Boosting | 85.01 | 83.56 | 85.98 |
| XGBoost | 89.95 | 87.42 | 88.21 |
| LDA-kNN | 84.79 | 81.00 | 82.95 |
| LDA-SVM-kNN | 84.18 | 84.99 | 83.99 |
| LDA-SVM-kNN-CNN | 76.04 | 73.89 | 74.45 |
| W-LDA-SVM-kNN | 85.35 | 83.97 | 86.00 |

performs slightly better on ALS 8-channel data than LDA and $k$NN. The simple ensemble averaging models LDA-SVM-$k$NN and LDA-$k$NN achieve more than 84% of accuracy, which is a meaningful result, despite the fact that these models are slightly outperformed by the boosting algorithms.

## IV. CONCLUSIONS

The proposed ensemble voting models based on LDA, SVM, $k$NN and CNN classifiers have been trained using generic training in order to achieve maximum subject-independency of the system. The LDA-$k$NN voter provided the best computational complexity, which makes it the most optimal option for large datasets. Weighted ensemble voter with SVM classifier provided the best performance for ALS patients data, achieving more than 85% of accuracy. There is a trade-off between the accuracy and the computational complexity. LDA-$k$NN voter is a better option for large datasets, while W-LDA-SVM-$k$NN provides better results, requiring much more time for training. Trained on the healthy subjects, ensemble voters turned out to be efficient for detecting P300 component from ALS patients data. For the further experiments the data collected from different electrodes is planned to be used to see if the number of channels can be decreased without affecting the performance. Moreover, the data from other types of neuropathy patients can be used in the future to evaluate the subject-independency of the system.

## REFERENCES


[1] L. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.

[2] L. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroenceph. Clin. Neurophysiol.*, vol. 70, pp. 510–523, 1998.

[3] T. Picton, "The P300 wave of the human event-related potential," *J. Clin. Neurophysiol.*, vol. 9, pp. 456–79, Nov. 1992.

[4] N. K. Squires, K. C. Squires, and S. A. Hillyard, "Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man," *Electroencephalog. Clin. Neurophysiol.*, vol. 38, no. 3, pp. 387–401, 1975.

[5] A. Mussabayeva, P. K. Jamwal, and M. T. Akhtar, "Comparison of generic and subject-specific training for features classification in P300 speller," in *Proc. 2020 Asia-Pacific Sign. and Inf. Process. Assoc. Annual Sum. Conf. (APSIPA ASC)*, 2020, pp. 222–227.

[6] O. Yanez-Suarez, L. Bougrain, C. Saavedra, E. Bojorges, and G. Gentiletti, *P300-Speller Public-Domain Database*. 2011. [Online]. Available: `http://akimpech.izt.uam.mx/p300db`.

[7] *P300 speller with Patients with ALS Database*. BCI Horizon 2020, 2020. [Online]. Available: `http://bnci-horizon-2020.eu/database/data-sets`.

[8] R. D. Fisher and P. Langley, "Methods of conceptual clustering and their relation to numerical taxonomy," *Artif. Intell. Stat.*, vol. 18, pp. 77–116, 1986.

[9] M. H. Lee, J. Williamson, D. O. Won, S. Fazli, and S. W. Lee, "A high performance spelling system based on EEG-EOG signals with visual feedback," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 7, pp. 1443–1459, 2018.

[10] C. P. Diehl and G. Cauwenberghs, "SVM incremental learning, adaptation and optimization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 4, 2003, pp. 2685–2690.

[11] C. Wang, J. Xu, S. Zhao, and W. Lou, "Identification of early vascular dementia patients with EEG signal," *IEEE Access*, vol. 7, pp. 68 618–68 627, 2019.

[12] Z. Lu, Q. Li, N. Gao, T. Wang, J. Yang, and O. Bai, "A convolutional neural network based on batch normalization and residual block for P300 signal detection of P300-speller system," in *Proc. IEEE Int. Conf. Mechatron. Autom. (ICMA)*, 2019, pp. 2303–2308.

[13] S. Kundu and S. Ari, "P300 based character recognition using convolutional neural network and support vector machine," *Biomed. Signal Process. Control*, vol. 55, 2020.

[14] S. Kundu and S. Ari, "Fusion of convolutional neural networks for P300 based character recognition," in *Proc. 2019 Int. Conf. Inf. Technol. (ICIT)*, 2019, pp. 155–159.

[15] K. S. Barsim, W. Zheng, and B. Yang, "Ensemble learning to EEG-based brain computer interfaces with applications on p300-spellers," in *Proc. 2018 IEEE Int. Conf. Syst. Man, Cybern. (SMC)*, 2018, pp. 631–638.

[16] T. Takeichi, T. Yoshikawa, and T. Furuhashi, "Detecting P300 potentials using weighted ensemble learning," in *Proc. 2018 Jt. 10th Int. Conf. Soft Comput. Intell. Syst. 19th Int. Symp. Adv. Intell. Syst. (SCIS-ISIS)*, 2018, pp. 950–954.

[17] L. A. Rastrigin, "The convergence of the random search method in the extremal control of many-parameter system," *Autom. Remote Control*, vol. 24, no. 10, pp. 1337–1342, 1963.

[18] M. A. Schumer and K. Steiglitz, "Adaptive step size random search," *IEEE Trans. Autom. Control*, pp. 270–276, Jul. 1968.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, 2011.

[20] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowl. Discov. and Data Min. (KDD)*, 2016, pp. 785–794.

[21] U. Hoffmann, G. Garcia, J. Vesin, K. Diserens, and T. Ebrahimi, "A boosting approach to P300 detection with application to brain-computer interfaces," in *Proc. 2nd Int. IEEE EMBS Conf. Neural Eng.*, 2005, pp. 97–100.