

**© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.**

# Subject-Independent P300 BCI using Ensemble Classifier, Dynamic Stopping and Adaptive Learning

Kha Vo<sup>\*†</sup>, Diep N. Nguyen<sup>\*</sup>, Ha Hoang Kha<sup>†</sup>, and Eryk Dutkiewicz<sup>\*</sup>

<sup>\*</sup> University of Technology Sydney

<sup>†</sup> Bach Khoa University

Global Big Data Technologies Centre Faculty of Electrical and Electronics Engineering

**Abstract**—Brain-computer interfaces (BCIs) are used to assist people, especially those with verbal or physical disabilities, communicate with the computer to indicate their selections, control a device or answer questions only by their mere thoughts. Due to the noisy nature of brain signals, the required time for each experimental session must be lengthened to reach satisfactory accuracy. This is the trade-off between the speed and the precision of a BCI system. In this paper, we propose a unified method which is the integration of ensemble classifier, dynamic stopping, and adaptive learning. We are able to both increase the accuracy, as well as to reduce the spelling time of the P300-Speller. Another merit of our study is that it does not require the training phase for any new subject, hence eliminates the extensively time-consuming process for learning purposes. Experimental results show that we achieve the averaged bit rate boost up of 182% on 15 subjects. Our best achieved accuracy is 95.95% by using 7.49 flashing iterations and our best achieved bit rate is 40.87 bits/min with 83.99% accuracy and 3.64 iterations. To the best of our knowledge, these results outperformed most of the related P300-based BCI studies.

**Index Terms**—Brain-computer interface, subject-independent, support vector machines

## I. INTRODUCTION

Brain-computer interfaces (BCIs) are used to translate intentions of users, especially those with severe motor impairments such as amyotrophic lateral sclerosis (ALS) [1], [2], into the system's commands. BCIs primarily use electroencephalograms (EEG), which is the electrical measurements on the human scalp, as the main modality. Event-related potential (ERP) is the dominant feature of EEG [3]. Generally, an ERP is an electrical response produced by the human brain as a reaction to a specific evoke or stimulus. On top of that, P300 [4] is one of the most robust components of ERP which had been used in various BCI studies [5]–[16]. P300 responses manifest as a positive voltage peak in EEG and presents approximately 300 ms after the visual stimulus onset.

To better exploit the predominant aspect of P300, Farwell first introduced the so-call P300-Speller (P3S) [17]. The ultimate goal of P3S BCIs is to help people indicate their choosing among the preset options on the computer screen. The set of available images on the screen can be a matrix of alphabetical letters or symbols from any language, or it can even be a group of graphical images indicating people's every basic needs (food, light, hygiene, etc.). We only care about the stimulus timing rather than the meaning of the images or the symbols on the screen. Therefore the set of user options can be almost anything. Each symbol in that set is flashed sequentially

according to a specific flashing paradigm. In most 36-letter P3S, each one-letter spelling session is usually conducted by a fixed number of iterations. Each iteration consists of 12 letter intensifications (or flashes), corresponding to 6 rows and 6 columns of the P3S. A signal timeline for a P3S is illustrated in Fig. 1.

Since then, various developments and research were carried out to boost up its performance, i.e. accuracy, speed, or bit rate. These developments can be the modifications on stimulus flashing paradigms [5]–[7], language model integration [8]–[11], feature processing and frameworks [12]–[14], or classification algorithms [15], [16]. However, the original method remains the same that the subjects focus their eyesight at one desired option out of the offered set of letters appearing on the screen, thereby to convey their spelling intentions.

As increasing iterations can improve the relatively low signal-to-noise (SNR) ratio of P300 responses over time and thereby increase the decision's accuracy, they however inevitably prolong the spelling session. To deal with this trade-off, various dynamic stopping (DS) methods had been proposed [1], [10], [18]–[22] to abruptly stop the flashing paradigm at any moment, when a set of user-defined conditions is satisfied. DS methods therefore can significantly reduce the spelling time while also preserving the expected accuracy that could be achieved by static stopping methods. We take a brief summary on each approach as follows.

Lenhardt [19] used two thresholds as the real-time criteria to dynamically stop the P3S. The first threshold represents the sum score of all normalized letter candidate scores (or called, the overall matrix brightness) over iterations. The average matrix brightness over 36 letters was proved by them to consistently decrease as the iteration increases. The second threshold is the ratio between two highest scores of two letter candidates.

Mainsah [1], [10], Clements [21], Kindermans [11], [22] all used the similar DS method on different signal processing frameworks. They implement the DS by updating the posterior probabilities of each letter via a Bayesian interference scheme after each iteration, given the classifier score history of the previous iterations. Kindermans [11], [22] provided a comprehensive study on unsupervised and adaptive transfer learning [22] (or subject-independent classification with unlabeled training data).

Since most existing works implemented the Bayesian probability update process, we propose a DS method by using two criteria calculated from the ensemble support vector machine



Let  $\mathbf{x}_R(s, i)$  and  $\mathbf{x}_C(s, j)$  (where  $i, j \in \{1, \dots, 6\}$ ) denote the preprocessed post-stimulus vector associated to the flash of row  $i$  and column  $j$ , respectively, during the  $s$ -th iteration. Equivalent the SVM classifier as described in Eq.(1), the score associated to one post-stimulus vector  $\mathbf{x}$  (where  $\mathbf{x}$  can stand for either  $\mathbf{x}_R(s, i)$  or  $\mathbf{x}_C(s, j)$ ), produced by classifier index  $p$ , is computed as

$$f_p(\mathbf{x}) = \sum_{i \in \mathcal{L}_p} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b, \quad (3)$$

where  $\mathbf{x}_i, y_i$  are the vectors and labels of the learning set  $\mathcal{L}_p$ , and  $\alpha_i$  are the evaluated SVM parameters corresponding to  $\mathcal{L}_p$ . In this paper, we implement the adaptive learning method which uses newly-classified trials to reinforce the existing SVM solution. Due to this reason, expressing the classifier's score as in the form of Eq.(3) is more appropriate for later analysis than the compact form of Eq.(1). Section IV will take more focus on the adaptive SVM scheme that is used in this section.

The accumulated score of each row or column, performed by  $P$  classifiers, after  $s$  iterations, is defined as

$$\begin{aligned} F_R(s, i) &= \sum_{t=1}^s \sum_{p=1}^P f_p(\mathbf{x}_R(t, i)), \\ F_C(s, j) &= \sum_{t=1}^s \sum_{p=1}^P f_p(\mathbf{x}_C(t, j)), \end{aligned} \quad (4)$$

where  $i, j \in [1, \dots, 6]$ .

### III. DYNAMIC STOPPING METHOD

#### A. Proposed Conditions for Dynamic Stopping

During each spelling session, the desired letter can be correctly elicited by using only a few first iterations when the classifiers' score is sufficiently confident about its decision, hence dramatically reduce the spelling time. The confidence level checking process is implemented continually after each iteration, and is almost based on the Bayesian updating scheme [1], [10], [11], [20]–[22]. However, since those schemes do not work with our adaptive SVM, we propose a method which based on the SVM scores. Our preliminary result had been reported in [27]. In this paper, we make some modifications to the criteria in [27] to better suits the real-time adaptive SVM. As compared to [27], the DS criteria in this paper are less computational. This is the desirable aspect for a real-time system. Moreover, we also propose an analytical grid search algorithm to obtain the optimal DS parameters. With this algorithm one can specify the DS thresholds which can yield pre-defined accuracies on purpose.

The candidate letter outputted by the classification scheme is determine by a pair of row  $\hat{i}_s$  and column  $\hat{j}_s$ , which intersect that letter on the P3S.

$$\{\hat{i}_s, \hat{j}_s\} = \operatorname{argmax}_{i, j \in [1, \dots, 6]} \{F_R(s, i) + F_C(s, j)\}.$$

Let  $\hat{\theta}_1$  and  $\hat{\theta}_2$  denote two thresholds to be compared with for two conditions of the dynamic stopping method. We define those two conditions as

- Condition 1:

$$\theta_1 = \frac{F_R(s, \hat{i}_s) - F_R(s, \hat{i}_s) + F_C(s, \hat{j}_s) - F_C(s, \hat{j}_s)}{1/6 \times \sum_{i, j=1}^6 (F_R(s, i) + F_C(s, j))} \geq \hat{\theta}_1 \quad (5)$$

where

$$\{\tilde{i}_s, \tilde{j}_s\} = \operatorname{argmax}_{i, j \in [1, \dots, 6]} \{F_R(s, i) + F_C(s, j)\}. \quad (6)$$

and

$$\{\hat{i}_s, \hat{j}_s\} = \operatorname{argmax}_{\substack{i, j \in [1, \dots, 6] \\ i \neq \tilde{i}_s; j \neq \tilde{j}_s}} \{F_R(s, i) + F_C(s, j)\}. \quad (7)$$

- Condition 2:

$$\theta_2 = \max_{i, j} \left( \sum_{p=1}^P v_p(s, i, j) \right) / P \geq \hat{\theta}_2, \quad (8)$$

where  $v_p(s, i, j) = 1$  if  $i = \operatorname{argmax}_{i'} \sum_{t=1}^s f_p(\mathbf{x}_R(t, i'))$ ,  $j = \operatorname{argmax}_{j'} \sum_{t=1}^s f_p(\mathbf{x}_C(t, j'))$ ; and  $v_p(s, i, j) = 0$  otherwise.

#### B. Estimation of $\hat{\theta}_1$ and $\hat{\theta}_2$

We conduct an iterative grid search on the learning set to estimate the optimal values for dynamic stopping parameters  $\hat{\theta}_1$  and  $\hat{\theta}_2$ . For each pair of  $\theta_1$  and  $\theta_2$ , an associated accuracy  $A(\theta_1, \theta_2)$  is generated. We can specify a desire accuracy reduction  $a$  as compared to the maximum accuracy, and then conduct the search to achieve the corresponding dynamic stopping parameters. Let  $S(a)$  denote the set of parameters which result in the maximum accuracy reduced by the margin of  $a\%$  as

$$S(a) = \{(\theta_1, \theta_2) | A(\theta_1, \theta_2) \geq \max_{\theta'_1, \theta'_2} A(\theta'_1, \theta'_2) - a\}. \quad (9)$$

The optimal values for  $\hat{\theta}_1$  and  $\hat{\theta}_2$  associated to the accuracy reduction  $a$  can be obtained as

$$\hat{\theta}_1, \hat{\theta}_2 = \operatorname{argmin}_{(\theta_1, \theta_2) \in S(a)} \{A(\theta_1, \theta_2)\}, \quad (10)$$

The grid search result is shown in Fig. 2. The tuning range for  $\theta_2$  is  $[0, \dots, 1]$  as  $\max_{i, j} \theta_2 = \sum_p \max_{i, j} v_p(s, i, j) / P = (\sum_p 1) / P = 1$ . Similarly, the tuning range for  $\theta_1$  can be obtained by calculating  $\lim_{s \rightarrow \infty} \theta_1$ . However since the tuning range for  $\theta_1$  is fixed, we can omit the specification for the tuning range of  $\theta_2$ , as we can iteratively increase  $\theta_2$  with step  $\eta_2$  from 0 until the maximum accuracy is obtained. The step  $\eta_1$  and  $\eta_2$  for increasing  $\theta_1$  and  $\theta_2$  can be freely chosen. In this paper, we select  $\eta_1 = 0.2$  and  $\eta_2 = 0.25$  through empirical observation. The algorithm of the iterative grid search method is presented in Algorithm 2.

### IV. ADAPTIVE LEARNING UPDATE

#### A. SVM Updating Methods for Streaming Data

This section briefly review the primal SVM method which has been mentioned in Section II, but in the dual form. The SVM dual form problem [24] is desirable for the adaptive learning technique which will be described in Section IV-B.

Let  $\mathcal{L} = \{(\mathbf{x}_{\mathcal{L}, l}, y_{\mathcal{L}, l})\}_{l=1}^{N_1}$  denote the initially given learning set of  $N_1$  training samples, and  $\mathcal{T} = \{(\mathbf{x}_{\mathcal{T}, t}, y_{\mathcal{T}, t})\}_{t=1}^{N_2}$  denote

$\theta_1$						
0.00	42	42	58	76	90	93
0.25	47	48	61	76	90	93
0.50	65	65	70	79	90	93
0.75	79	79	81	85	90	93
1.00	83	83	83	88	92	93
1.25	89	89	88	91	92	93
1.50	92	92	92	93	93	93
1.75	93	93	93	93	93	93
2.00	93	93	93	93	93	93
$\theta_2$	0.0	0.2	0.4	0.6	0.8	1.0

(a) Accuracy

$\theta_1$						
0.00	1.0	1.1	1.9	4.0	6.7	15.0
0.25	1.2	1.3	2.1	4.0	6.7	15.0
0.50	2.3	2.4	3.0	4.5	6.8	15.0
0.75	4.8	4.8	5.1	5.9	7.7	15.0
1.00	7.0	7.0	7.2	7.9	9.0	15.0
1.25	10.0	10.0	10.1	10.4	10.8	15.0
1.50	11.9	11.9	11.9	12.1	12.3	15.0
1.75	13.3	13.3	13.3	13.5	13.5	15.0
2.00	14.1	14.1	14.1	14.2	14.2	15.0
$\theta_2$	0.0	0.2	0.4	0.6	0.8	1.0

(b) Iterations needed

Figure 2: The iterative grid search result for the estimation of  $\hat{\theta}_1$  and  $\hat{\theta}_2$ .

**Algorithm 1** Real-time dynamic stopping algorithm for one letter spelling session

---

```

1: Input:  $\hat{\theta}_1, \hat{\theta}_2$ 
2: Output:  $s, \tilde{i}_s, \tilde{j}_s$ 
3: Initialization:  $s \leftarrow 0, s_{max}$ 
4: while DYNAMIC_FLAG  $\neq 1$  do
5:   if  $s \leq s_{max}$  then
6:      $s \leftarrow s + 1$ 
7:     Compute  $F_R(s, i), F_C(s, j)$  (as in Eq.(4))
8:     Compute  $\theta_1, \theta_2$  (as in Eq.(5) and Eq.(8))
9:     if  $(\theta_1 \geq \hat{\theta}_1)$  and  $(\theta_2 \geq \hat{\theta}_2)$  then
10:       DYNAMIC_FLAG  $\leftarrow 1$ 
11:     end if
12:   else DYNAMIC_FLAG  $\leftarrow 1$ 
13:   end if
14: end while
15: Stop flashing, compute  $\tilde{i}_s, \tilde{j}_s$  (as in Eq.(6))
16: return  $s, \tilde{i}_s, \tilde{j}_s$ 

```

---

the testing set of  $N_2$  samples, and  $N = N_1 + N_2$ . The optimal solution for  $\mathcal{L}$ , denoted by  $\{\alpha, b\}$ , is obtained by solving the following dual quadratic program

$$\min_{0 \leq \alpha_i \leq C} \frac{1}{2} \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{L}} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i \in \mathcal{L}} \alpha_i + b \sum_{i \in \mathcal{L}} y_i \alpha_i. \quad (11)$$

In this paper we employ the SVM solver provided by [28]. To obtain the appropriate regularization parameter  $C$  in the problem above, we conduct the leave-one-partition-out cross-validation on the learning set (where the possible values for  $C$  are 0.01, 0.05, 0.1, 0.25, 0.50, 0.75, and 1.00). In other words,

---

**Algorithm 2** Iterative grid search for  $\hat{\theta}_1, \hat{\theta}_2$

---

```

1: Input:  $a$ 
2: Output:  $\hat{\theta}_1, \hat{\theta}_2$ 
3: Initialization:  $\eta_1, \eta_2$ 
4:  $\theta_1 \leftarrow 0, \theta_2 \leftarrow 0, A_{max} \leftarrow 0, \text{STOP\_FLAG} \leftarrow 0, S \leftarrow \emptyset$ 
5: while STOP_FLAG  $\neq 1$  do
6:   for  $\theta_2 = 0 : 1$  (step  $\eta_2$ ) do
7:     Compute and save  $A(\theta_1, \theta_2)$ 
8:     if  $A(\theta_1, \theta_2) > A_{max}$  then  $A_{max} \leftarrow A(\theta_1, \theta_2)$ 
9:     end if
10:  end for
11:  if  $A(\theta_1, 0) = A_{max}$  and  $A(\theta_1, 0) = A(\theta_1 - \eta_1, 0)$  then
12:    STOP_FLAG  $\leftarrow 1; \theta_{1,max} \leftarrow \theta_1$ 
13:  end if
14:   $\theta_1 \leftarrow \theta_1 + \eta_1$ 
15: end while
16: for each  $\theta_1, \theta_2$  do
17:   if  $A(\theta_1, \theta_2) \geq A_{max} - a$  then add  $(\theta_1, \theta_2)$  to  $S$ 
18:   end if
19: end for
20: Search for  $\hat{\theta}_1, \hat{\theta}_2$  in  $S$  (as in Eq.(10))
21: return  $\hat{\theta}_1, \hat{\theta}_2$ 

```

---

each training partition has its own regularization parameter  $C$  which yields the highest accuracy validated on the other partitions.

In an adaptive learning method, one needs to use the existing hyperplane solution of the learning set  $\mathcal{L}$  to find the optimal solution for the combined set  $\mathcal{C} = \mathcal{L} \cup \mathcal{T}$ , denoted by  $\{\hat{\alpha}, \hat{b}\}$ . The label  $y_t$  of  $\mathcal{T}$  is first obtained by estimating  $\mathbf{x}_t$  from  $\{\alpha, b\}$  based on the SVM framework as

$$y_t = f(\mathbf{x}_t) = \sum_{i \in \mathcal{L}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_t) + b, \forall j \in \{N+1, \dots, N+n\}, \quad (12)$$

where  $[\alpha_1, \dots, \alpha_N]^T = \alpha$  and  $K(\cdot, \cdot)$  is the kernel. In this paper we adopt the homogeneous polynomial kernel  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$ .

### B. Incremental SVM

Incremental SVM (ISVM) learning [28] takes approaching classified samples as the new training data to adjust the available classifier parameters without re-conducting the optimization problem on the whole training set, which is expensively computational. ISVM is an on-line method which takes one new training sample at a time and increment the existing hyperplane solution, as long as the Kuhn-Tucker conditions are satisfied on all data samples (including the old and new points). Considering ISVM as a black box which takes  $\alpha_{\mathcal{L}}, b_{\mathcal{L}}$  as the existing solution input (obtained from  $\mathcal{L}$ ), and  $\mathcal{T}$  as the incremental set, we can briefly defined this method as

$$(\alpha_{\mathcal{T}}, b_{\mathcal{T}}) = \text{ISVM}(\mathcal{L}, \alpha_{\mathcal{L}}, b_{\mathcal{L}}, \mathcal{T}) \quad (13)$$

In our implementation of ISVM on the proposed P300 framework, the feature vectors which were classified during

the  $s$ -th iteration are used to increment the SVM solution for the  $(s + 1)$ -th iteration. It should be noticed that the labels of the newly-classified vectors are determined by the row/column candidates  $\tilde{i}_s, \tilde{j}_s$ , rather than the vector-by-vector classifications achieved from the  $s$ -th iteration. The on-line incremental learning method is presented in detail in Algorithm 3.

**Algorithm 3** On-line incremental learning update for continual P300 spelling sessions

---

```

1: Initialization:  $k \leftarrow 0, \alpha \leftarrow \alpha_{\mathcal{L}}, b \leftarrow b_{\mathcal{L}}$ 
2: while USER_STOP  $\neq 1$  do
3:   Implement flashing paradigm of the  $k$ -th session
4:   Implement Dynamic Stopping (as in Algorithm 1)
5:   Obtain  $s, \tilde{i}_s, \tilde{j}_s$  using solution  $(\alpha, b)$ 
6:   Input: USER_STOP (1 for stop)
7:   Assign new labels for  $y_{\mathcal{T},t}$  according to  $\tilde{i}_s, \tilde{j}_s$ 
8:   Add new  $N_2 = s \times 12$  samples to  $\mathcal{T} = \{(\mathbf{x}_{\mathcal{T},t}, y_{\mathcal{T},t})\}_{t=1}^{N_2}$ 
9:   Increment solution  $(\alpha, b) \leftarrow \text{ISVM}(\mathcal{L}, \alpha, b, \mathcal{T})$ 
10: end while

```

---

## V. EXPERIMENTS AND DISCUSSION

This section describes the experimental settings, including a brief review on the dataset, classification schemes, preprocessing techniques, evaluation metrics, experimental results, and comparisons between our approach with the others.

### A. Data sets

The Akimpech dataset [29] consists of P300-based EEG signals of 30 subjects. Each subject has 15 letters sessions for training and 12-24 sessions for testing. Each session is composed of 15 iterations. The inter-stimulus interval is 125 ms. We use the data of the first 9 subjects (ACS, APM, ASG, ASR, CLL, DCM, DLP, DMA, ELC) for the training purposes and 15 subjects (FSZ, GCE, ICE, IZH, JLD, JLP, JMR, JSC, JST, LAC, LAG, LGP, PGA, WFG, XCL) for evaluation. The data from each training subject is used as one constituent classifier for the ensemble scheme. The data was recorded at the sampling rate of 256 Hz with 10 channels (Fz, C3, Cz, C4, P3, Pz, P4, PO7, PO8, Oz).

TABLE I: CLASSIFICATION SCHEMES' SETTINGS

Scheme	Learning Subjects	Samples/Partition	Partitions	Ensemble	Adaptive
DS3	3	1620	1	No	No
DS9	9	4860	1	No	No
EN3	3	540	3	Yes	No
EN9	9	540	9	Yes	No
ADA	3	540	3	Yes	Yes

### B. Classification Schemes

The single classifier DS3 is learned from the mixed learning samples of 3 subjects ACS, APM, and ASG. Similarly, DS9 is learned from the mixed samples of all 9 training subjects as described in Section V-A. Both DS3 and DS9 do not involve the adaptive learning. EN3 and EN9 are the ensemble

schemes which use the same learning data as in DS3 and DS9, respectively, although both EN3 and EN9 also do not involve the adaptive learning. ADA is the fully-integrated scheme involving ensemble classifier, dynamic stopping, and adaptive learning which use the same learning data as DS3 and EN3. The details of the mentioned classification schemes are presented in Table I.

### C. Signal Preprocessing

Prior to any learning or testing stage, we perform some preprocessing techniques on the raw EEG dataset. Each post-stimulus trial has the length of 66.67 ms, equivalent to 160 discrete samples per channel. The extracted feature vectors are decimated and filtered by the 4-th order band-pass filter with the cut-off frequencies of 0.1 and 20 Hz. After decimation each vector has the size of 14 samples/channel  $\times$  10 channels = 140 samples. Then each decimated and filtered vector is normalized into the interval of  $[-1, 1]$ .

### D. Evaluation Metrics

We use three metrics for evaluating the BCI performances of our proposed schemes, namely, the accuracy ( $A$ ), number of iterations (or iterations) and theoretical bit rate (BR). The accuracy is simply computed as the number correctly classified letters over the total  $T$  spelled letters per subject

$$A = (\text{correct characters})/T.$$

The bit rate per trial  $B$  is computed from the accuracy  $A$  and the total of possible spelled letters in the P3S  $L = 36$  as

$$B = \log_2 L + A \log_2 A + (1 - A) \log_2 \frac{1 - A}{L - 1} \text{ (bits/trial)}. \quad (14)$$

The theoretical bit rate (BR) is taken into account for measuring P3S communication speed as:

$$\text{BR} = B / (s \times \text{ISI} \times 12/60) \text{ (bits/min)}$$

where  $s$  is the number of flashing sequences, ISI is the inter-stimulus interval.

### E. Experimental Results

We implement the experiments on 5 proposed schemes (Table I) using the 9 selected pairs of dynamic stopping parameters  $\hat{\theta}_1 \in \{0.3, 0.5, 0.7\}$  and  $\hat{\theta}_2 \in \{0.5, 1.0, 1.5\}$ . The averaged result of all subjects on each parameters pair is shown in Table II. Notice that  $\hat{\theta}_1 = 1.5$  and  $\hat{\theta}_2 = 0.7$  are also the parameters obtained from the iterative grid search method as presented in Algorithm 2, with input  $a = 0$ . The influence of ensemble method is demonstrated in the accuracy improvement between DS3 and EN3 (accuracy rises from 65.31% to 74.03%), or between DS9 and EN9 (accuracy rises from 70.01% to 84.89%). However there are also rises in flashing iterations from DS3 to EN3 (5.96 to 8.56), and from DS9 to EN9 (5.69 to 9.35), which we really do not expect to happen. This problem causes a slight reduction in the resulted bit rates from DS3 to EN3 (16.50 to 14.08), as well as from DS9 to EN9 (19.32 to 16.18). The most valuable

TABLE II: CLASSIFICATION RESULTS USING DIFFERENT VALUES OF DYNAMIC STOPPING CRITERIA  $\hat{\theta}_1, \hat{\theta}_2$ . THE ITERATIVE GRID SEARCH (ALGORITHM 2) WITH INPUT  $a = 0$  YIELDS  $\hat{\theta}_1 = 1.50, \hat{\theta}_2 = 0.70$ ; WITH  $a = 5$  YIELDS  $\hat{\theta}_1 = 1.00, \hat{\theta}_2 = 0.50$ ; AND WITH INPUT  $a = 10$  YIELDS  $\hat{\theta}_1 = 1.00, \hat{\theta}_2 = 0.30$ .

DS Criteria		Accuracy (%)					Iterations					Bit rate (bits/min)				
$\hat{\theta}_1$	$\hat{\theta}_2$	DS3	DS9	EN3	EN9	ADA	DS3	DS9	EN3	EN9	ADA	DS3	DS9	EN3	EN9	ADA
0.50	0.30	44.13	52.65	56.83	69.59	62.06	1.86	1.87	4.10	3.70	2.27	28.20	37.28	19.23	29.46	39.98
0.50	0.50	44.37	52.65	64.70	85.35	63.30	1.87	1.87	4.81	7.96	2.49	28.39	37.28	20.13	19.18	37.64
0.50	0.70	44.13	52.65	80.78	89.65	83.99	1.86	1.87	9.72	11.88	3.64	28.20	37.28	14.32	14.01	40.87
1.00	0.30	70.15	72.53	71.11	77.80	83.19	5.98	5.28	6.75	6.01	4.25	18.46	22.07	16.72	21.76	34.42
1.00	0.50	69.87	72.53	72.83	86.87	83.66	5.97	5.28	7.24	8.74	4.32	18.38	22.07	16.21	18.01	34.16
1.00	0.70	70.15	72.53	81.59	89.65	86.53	5.98	5.28	10.46	11.99	4.68	18.46	22.07	13.53	13.87	33.38
1.50	0.30	81.83	84.85	78.50	87.19	92.85	9.99	9.93	10.57	10.16	7.34	14.23	15.22	12.55	15.59	24.15
1.50	0.50	81.87	84.85	79.83	88.80	94.44	9.97	9.93	10.91	11.05	7.03	14.27	15.22	12.50	0.30	26.03
1.50	0.70	81.32	84.85	80.10	89.14	95.95	10.19	9.93	12.46	12.66	7.49	13.80	15.22	11.01	13.01	25.20
<b>Average</b>		<b>65.31</b>	<b>70.01</b>	<b>74.03</b>	<b>84.89</b>	<b>82.89</b>	<b>5.96</b>	<b>5.69</b>	<b>8.56</b>	<b>9.35</b>	<b>4.83</b>	<b>16.50</b>	<b>19.32</b>	<b>14.08</b>	<b>16.18</b>	<b>30.06</b>

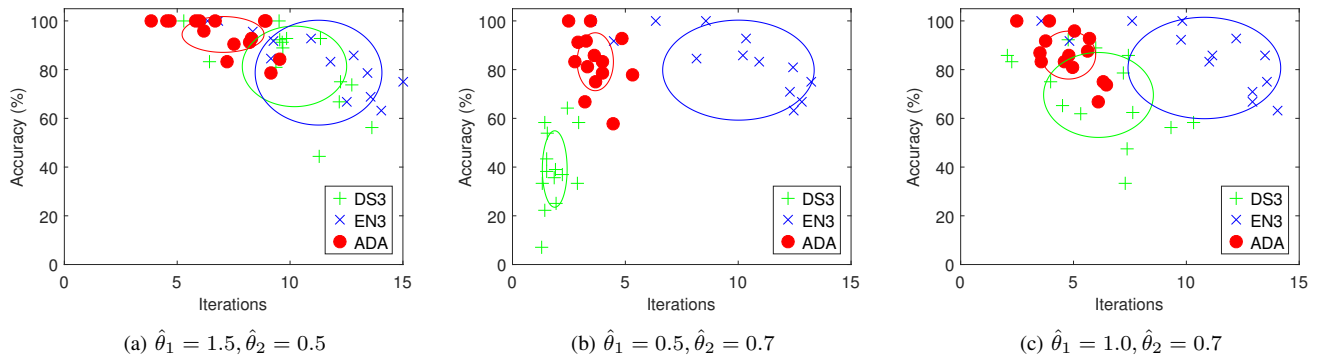


Figure 3: The subject-wise results for all test subjects using different input values of dynamic stopping parameters  $\hat{\theta}_1$  and  $\hat{\theta}_2$ . Each individual point represents the performance of one subject. The ellipses represents the average area of each scheme, with the centers of the ellipses are the mean and the radii are the standard deviations of the accuracy and iterations on the whole scheme.

is the significant improvements in both accuracy and iteration of ADA. Averagely, ADA boost up nearly 20% in accuracy while also reduce spelling time from 5.96 to 4.83 iterations. Combining these two improvements, we retrieve a huge leap from 16.50 to 30.06 in bit rate, which is nearly equivalent to a two-fold increase.

The subject-wise results of 4 selected parameter pairs are shown in Fig. 3. We exclude the display for DS9 and EN9 since they use more (three-fold) learning samples as compared to DS3, EN3, and ADA. According to Fig. 3, although EN3 can improve the accuracy (as compared to DS3), it also causes the instability in performance, as presented in the broad areas of EN3. On the other hand, ADA always achieves the highest accuracies as well as the lowest iterations in all subjects, as compared to both DS3 and EN3. The ADA areas are narrower, demonstrating that ADA is much more stable than EN3.

#### F. Comparison with Related Studies

Kindermans *et al.* [22] proposed a combined method of dynamic stopping, transfer learning and language model which can also be adapted to novel subjects. Their approach is totally different to ours in terms of classification algorithm, adaptive learning and dynamic stopping method. However they also implemented their work on the Akimpech dataset, hence we are able to perform direct comparison (in terms of accuracy and iteration) between our study and theirs. Using the dynamic stopping and transfer learning scheme, they obtained the accuracy of 93.33% with 6.3 iterations. In this

paper, our best case is 95.95% accurate with 7.49 iterations ( $\hat{\theta}_1 = 1.5, \hat{\theta}_2 = 0.7$ ). For comparison, our approach attains better accuracy with a slightly higher number of iterations. Lenhardt [19] conducted their subject-dependent experiments using their own data set on the 36-letter P3S. When combining the DS method into their approach, they achieved the bit rate boost of +40.6% at the approximate accuracy of  $80\% \pm 5\%$ . In our work, we obtained the two-fold bit rate boost of +87.69% but with much higher accuracy (95.95%), although we implement the subject-independent classification rather than subject-dependent scheme.

The other works which were mentioned in the literature [1], [10], [11], [21], [30] were not implemented on the Akimpech dataset, as well as not on the subject-independent basis. However, they all have the same dynamic stopping method of Bayesian approach, which was first proposed by [30]. Therefore for a more direct comparison between our proposed method with theirs, we have re-implemented their methods on the Akimpech dataset (using scheme DS3) with subject-independent manner that was used in our paper. The comparison is presented in Table III. According to Table III, our proposed method significantly outperformed the mentioned works in terms of averaged accuracy on 15 validation subjects, although we also require the approximate number of iterations.

## VI. CONCLUSION

We have proposed a novel framework to integrate ensemble classifier, dynamic stopping, and adaptive learning into the

TABLE III: COMPARISON BETWEEN OUR METHOD WITH THE RELATED STUDIES ON THE SUBJECT-INDEPENDENT MANNER. THE DS PARAMETER OF THE BAYESIAN APPROACH IS THE CHARACTER PROBABILITY THRESHOLD [30], WHILE OUR DS PARAMETER IS  $a$  IN ALGORITHM 2.

Method	DS parameter	Accuracy (%)	Iterations
Bayesian [1], [10], [11], [21], [30]	0.90	63.18	4.19
	0.95	64.07	4.84
	0.99	69.93	6.32
Our method	0	95.95	7.49
	5	83.66	4.32
	10	83.19	4.25

subject-independent P300-based BCI, which requires no training stage for new subjects. The experimental results conducted on the multi-subject dataset show that our proposed method can both increase the accuracy and reduce the spelling time. Generally, we can obtain the bit rate boost of nearly two-fold while attaining over 90% accuracy. The trade-off for this boost is the additional computational burden for the system. Due to the limit scope of this paper, we do not bring into account the analysis of computational cost. Further analysis to be investigated are the effects of tuning the dynamic stopping criteria on the adaptive learning performance, and the effects of ensemble classifier dividing scheme.

#### REFERENCES

- [1] B. O. Mainsah, L. M. Collins, K. A. Colwell, E. W. Sellers, D. B. Ryan, K. Caves, and C. S. Throckmorton, "Increasing BCI communication rates with dynamic stopping towards more practical use: an ALS study," *Journal of Neural Engineering*, vol. 12, no. 1, p. 016013, 2015.
- [2] E. W. Sellers and E. Donchin, "A P300-based brain-computer interface: Initial tests by ALS patients," *Clinical Neurophysiology*, vol. 117, no. 3, pp. 538–548, 2006.
- [3] S. Sutton, M. Braren, J. Zubin, and E. R. John, "Evoked-Potential Correlates of Stimulus Uncertainty," *Science*, vol. 150, no. 3700, p. 1187, 1965.
- [4] T. W. Picton, "The P300 wave of the human event-related potential," *J. Clin. Neurophysiol.*, vol. 9, pp. 456–479, 1992.
- [5] R. C. Panicker, S. Puthusserypady, and Y. Sun, "An Asynchronous P300 BCI With SSVEP-Based Control State Detection," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1781–1788, 2011.
- [6] K. Tobias and K. Andrea, "Beyond maximum speed—a novel two-stimulus paradigm for brain–computer interfaces based on event-related potentials (P300-BCI)," *Journal of Neural Engineering*, vol. 11, no. 5, p. 056004, 2014.
- [7] E. Yin, T. Zeyl, R. Saab, T. Chau, D. Hu, and Z. Zhou, "A Hybrid Brain-Computer Interface Based on the Fusion of P300 and SSVEP Scores," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 4, pp. 693–701, 2015.
- [8] W. Speier, C. Arnold, J. Lu, A. Deshpande, and N. Pouratian, "Integrating Language Information With a Hidden Markov Model to Improve Communication Rate in the P300 Speller," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 3, pp. 678–684, 2014.
- [9] W. Speier, C. Arnold, and N. Pouratian, "Integrating language models into classifiers for BCI communication: a review," *Journal of Neural Engineering*, vol. 13, no. 3, p. 031002, 2016.
- [10] B. O. Mainsah, K. A. Colwell, L. M. Collins, and C. S. Throckmorton, "Utilizing a Language Model to Improve Online Dynamic Data Collection in P300 Spellers," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 837–846, 2014.
- [11] P. J. Kindermans, H. Verschore, and B. Schrauwen, "A Unified Probabilistic Approach to Improve Spelling in an Event-Related Potential-Based Brain-Computer Interface," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2696–2705, 2013.
- [12] T. Zeyl, E. Yin, M. Keightley, and T. Chau, "Adding Real-Time Bayesian Ranks to Error-Related Potential Scores Improves Error Detection and Auto-Correction in a P300 Speller," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 1, pp. 46–56, 2016.
- [13] B. Wittevrongel and M. M. Van Hulle, "Faster P300 Classifier Training Using Spatiotemporal Beamforming," *International Journal of Neural Systems*, vol. 26, no. 03, p. 1650014, 2016.
- [14] G. Townsend and V. Platsko, "Pushing the P300-based brain–computer interface beyond 100 bpm: extending performance guided constraints into the temporal domain," *Journal of Neural Engineering*, vol. 13, no. 2, p. 026024, 2016.
- [15] H. Cecotti, M. P. Eckstein, and B. Giesbrecht, "Single-Trial Classification of Event-Related Potentials in Rapid Serial Visual Presentation Tasks Using Supervised Spatial Filtering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2030–2042, 2014.
- [16] G. Dornhege, B. Blankertz, G. Curio, and K. R. Muller, "Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 993–1002, 2004.
- [17] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.
- [18] L. Tao, G. Leslie, G. Shangkai, and H. Bo, "An online brain–computer interface using non-flashing visual evoked potentials," *Journal of Neural Engineering*, vol. 7, no. 3, p. 036003, 2010.
- [19] A. Lenhardt, M. Kaper, and H. J. Ritter, "An Adaptive P300-Based Online Brain-Computer Interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 2, pp. 121–130, 2008.
- [20] B. O. Mainsah, L. M. Collins, and C. S. Throckmorton, "Using the detectability index to predict P300 speller performance," *Journal of Neural Engineering*, vol. 13, no. 6, p. 066007, 2016.
- [21] J. M. Clements, E. W. Sellers, D. B. Ryan, K. Caves, L. M. Collins, and C. S. Throckmorton, "Applying dynamic data collection to improve dry electrode system performance for a P300-based brain–computer interface," *Journal of Neural Engineering*, vol. 13, no. 6, p. 066018, 2016.
- [22] K. Pieter-Jan, T. Michael, M. Klaus-Robert, and S. Benjamin, "Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller," *Journal of Neural Engineering*, vol. 11, no. 3, p. 035005, 2014.
- [23] P.-J. Kindermans, D. Verstraeten, and B. Schrauwen, "A Bayesian Model for Exploiting Application Constraints to Enable Unsupervised Training of a P300-based BCI," *PLOS ONE*, vol. 7, no. 4, p. e33758, 2012.
- [24] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT '92), July 27-29, 1992, Pittsburgh, PA, USA* (D. Haussler, ed.), pp. 144–152, ACM Press, New York, NY, USA, 1992.
- [25] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Proceedings of the First International Workshop on Multiple Classifier Systems*, pp. 1–15, Springer-Verlag, 2000.
- [26] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [27] K. Vo, D. Nguyen, H. H. Kha, and E. Dutkiewicz, "Dynamic Stopping Using eSVM Scores Analysis for Event-Related Potential Brain-Computer Interfaces," in *11th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2017.
- [28] C. P. Diehl and G. Cauwenberghs, "SVM incremental learning, adaptation and optimization," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 4, pp. 2685–2690 vol.4, 2003.
- [29] O. Y.-S. and, L. Bougrain, C. Saavedra, E. Bojorges, and G. Gentiletti, "P300-speller public-domain database," 2011.
- [30] C. S. Throckmorton, K. A. Colwell, D. B. Ryan, E. W. Sellers, and L. M. Collins, "Bayesian Approach to Dynamically Controlling Data Collection in P300 Spellers," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 3, pp. 508–517, 2013.