SLES: A Novel CNN-based Method for Sensor Reduction in P300 Speller

Hongchang Shan, Todor Stefanov

Abstract— **A Brain Computer Interface (BCI) character speller allows human-beings to directly spell characters using eye-gazes, thereby building communication between the human brain and a computer. Current popular BCI character speller systems employ a large number of sensors, which prevents the utilization of such systems in human's daily life. Using sensor selection methods to select appropriate sensor subsets from an initial large sensor set can reduce the number of sensors needed to acquire brain signals without losing the character spelling accuracy, thereby promoting the BCI character spellers into people's daily life. However, current sensor selection methods cannot select an appropriate sensor subset such that they can further reduce the number of sensors needed to acquire brain signals without losing the spelling accuracy. To address this issue, we propose a novel sensor selection method based on a specific Convolutional Neural Network (CNN) we have devised. Our method uses a parametric backward elimination algorithm which uses our devised CNN as a ranking function to evaluate sensors and eliminate less important sensors. We perform experiments on three benchmark datasets and compare the minimal number of sensors selected by our proposed method and other selection methods to acquire brain signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. The results show that the minimal number of sensors selected by our method is lower than the minimal number of sensors selected by other methods in most cases. Compared with the minimal number of sensors selected by other methods, our method can reduce this number with up to 44 sensors.**

I. Introduction

A Brain Computer Interface (BCI) translates brain signals into computer commands, thereby building communication between the human brain and outside devices. In this way, human-beings can use only the brain to express their thoughts without any real movement. As a result, BCIs become an important communication pathway for the people who lose motor ability, such as patients with Amyotrophic Lateral Sclerosis (ALS) [1] or spinal-cord injury. In recent years, BCI has also been popularly developed for healthy people, in application domains such as entertainments [2], mental state monitoring [3] as well as in IoT services [4]. Electroencephalogram (EEG)-based BCI attracts most of the research due to its noninvasive way of measuring/acquiring brain signals [5]. An EEG-based BCI includes an EEG headset for acquisition of EEG signals as well as a hardware/software platform for processing and translating EEG signals into computer commands. An important application of EEG-based BCIs is the P300 speller [6] because the P300 speller performs outstandingly well among all kinds of EEG-based character spellers.

Nevertheless, the P300 speller is still not used in human's daily life and remains in an experimental stage at research labs. Some of the reasons for this situation are : 1) Current popular EEG headsets in BCI systems used for the P300 speller employ a large number of sensors to achieve high spelling accuracy. For example, the BCI systems Brain Products ActiCHamp, g.HIamp and Biosemi employ up to 160, 256, and 256 sensors, respectively. The price of the EEG headset is significantly high when the number of sensors is large because a lot of sensors require a complicated electrode cap and a lot of amplifier channels. For example, a 256-sensor BCI system (BioSemi) costs 87000 dollars while a 14-sensor BCI system (EMOTIV EPOC+) costs 799 dollars; 2) Employing a large number of sensors makes the P300 speller to consume a lot of power, which is unacceptable for a battery-powered mobile BCI system. Such system employs a wireless EEG headset and a resource-constrained hardware platform for data processing. A large number of sensors increase the amount of the data needed to be recorded and processed, thereby increasing the power consumption of the wireless BCI headset and the hardware platform. This does not allow a mobile P300 speller to work for a long time period on a single battery charge; 3) Employing a large number of sensors strengthens the user's discomfort and increases the installation time of the P300 speller.

To address the aforementioned problems caused by the employment of a large number of sensors, sensor selection methods could be used to select an appropriate sensor subset from an initial large set of sensors while keeping acceptable spelling accuracy. So, a good sensor selection method should enable substantial reduction of the sensors needed to acquire brain signals. Therefore, good sensor selection methods are in urgent need for designing comfortable, cheap, and power-efficient P300 spellers and for promoting such P300 spellers into the human's daily life. Sensor selection methods for the P300 speller have been studied in recent years. For example, [7] [8] [9] [10] employ a backward elimination algorithm as a sensor selection strategy. These works propose different ranking functions to evaluate and eliminate sensors such as the P300 signal detection accuracy, the P300 spelling accuracy [10], C_{cs} score [7], Signal to Signal and Noise Ratio (SSNR) [8] [9] [10], Area Under the Receiver Operating Characteristic (AUC) [11]. Alternatively, [12] and [13] directly select the important sensors for a given user by analysing the weights of a trained neural network. Unfortunately, the aforementioned sensor selection methods cannot select an appropriate sensor subset such that they can further reduce the number of sensors used to acquire brain signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. As a consequence, the cost, power consumption, and discomfort of a P300 speller are still unacceptably high when using the aforementioned sensor selection methods to design and configure P300 spellers. In order to further reduce the cost and power consumption of a P300 speller, we propose an effective sensor selection method based on a specific novel Convolutional Neural Network (CNN) we have devised. The novel contributions of this paper are the following:

- Our devised CNN extracts the spatial features related to P300 signals from the input EEG signals. Our sensor selection method uses this CNN to evaluate and rank the sensors in the sensor selection process. This method features an iterative parametric backward elimination algorithm to eliminate and select sensors. The parameter configured in this algorithm controls the training frequency of the CNN and the number of sensors to eliminate in every iteration.
- We perform experiments on three benchmark datasets and compare the minimal number of sensors selected by our proposed method and other selection methods needed to acquire brain signals while keeping the

Hongchang Shan and Todor Stefanov are with Leiden Institute of Advanced Computer Science, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands. E-mail: hongchang.shan@outlook.com, t.p.stefanov@liacs.leidenuniv.nl

978-1-5386-1311-5/19/$31.00 ©2019 IEEE

spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. The results show that, compared with the minimal number of sensors selected by other methods, our method can reduce this number with up to 44 sensors.

The rest of the paper is organized as follows. Section II describes the related work. Section III provides background information on the P300 speller and the datasets used in this paper. Section IV presents the proposed sensor selection method. Section V describes the experimental setup and the experimental results on the comparison of the minimal number of sensors selected by our proposed method and other sensor selection methods to acquire brain signals for the P300 speller. Section VI discusses how the number of sensors eliminated in an iteration influences the performance of our proposed method as well as how the CNN network architecture influences the sensor selection process. Section VII ends the paper with conclusions.

II. Related Work

In this section, we describe the related works on sensor selection methods for the P300 speller in BCI.

[7] [10] employ a backward elimination algorithm as a sensor selection strategy. Different ranking functions are proposed to evaluate and eliminate sensors. These ranking functions include the P300 detection accuracy, the average spelling accuracy across different epochs [10], C_{cs} score [7], Signal to Signal and Noise Ratio (SSNR) [10] and Area Under the Receiver Operating Characteristic (AUC) [11]. In order to select a sensor subset, the backward elimination algorithm either eliminates one sensor [10] or a group of sensors [7] in each iteration of the algorithm. Starting with a set of n sensors in an iteration, the backward elimination algorithm removes each sensor in the current sensor set and evaluates the resulting subsets with $(n-1)$ sensors using the aforementioned ranking functions. The sensor or the group of sensors which removal maximizes the ranking score is eliminated. In contrast to these methods, we proposes a novel ranking function (see Section IV-C) based on a specific novel Convolutional Neural Network (CNN) we have devised. Experimental results (see Section V-B) show that our sensor selection method is able to select a sensor subset with smaller number of sensors needed to acquire the brain signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used, compared with the sensor subset selected by the aforementioned sensor selection methods. Therefore, our sensor selection methods can further reduce the cost and power consumption of the P300 speller.

[12] and [13] propose CNN-based classifiers for character spelling in the P300 speller. By analysing the weights of the spatial convolution layer of their trained CNNs, they determine which sensors are more important in the sensor set. This can be a potential sensor selection method for the P300 speller. However, the problem of such potential method is that it loses important information needed for proper sensor selection. The aforementioned CNNs have multiple convolution layers and only the information of the first layer is used for analysis and sensor selection. Unfortunately, the information needed for proper sensor selection is distributed over all convolution layers, thus the aforementioned methods lose information needed for sensor selection from other convolution layers. In contrast to the aforementioned CNNs, we propose a simple and novel CNN which has only one convolution layer and this layer performs the spatial convolution operation. All the information needed for sensor selection is captured by the weights of this single spatial convolution layer. Moreover, our CNN has similar ability to extract very useful P300-related features compared to the aforementioned CNNs (see Section VI-B). We analyse the weights of the single spatial convolution layer in our CNN to select sensors. Thus, our method uses all the information available for proper sensor selection compared to the aforementioned methods. As a result, our method can select more appropriate sensor subsets and further reduce the minimal number of sensors needed to acquire brain signals without losing spelling accuracy. For more detailed discussion see Section VI-C.

III. Background

In this section, we provide some background information for the P300 speller method and the benchmark datasets used in this paper.

A. P300 Speller

The P300 speller is one of the most investigated applications in BCI [14] [15] [6]. A target character is spelled using the property of the P300 signal. As shown in Figure 1, a P300 signal, recorded in EEG, occurs as a positive deflection in voltage with a latency of about 300ms after a rare stimulus is presented to a subject (person). The following method is used to evoke a P300 signal in a subject's brain and then the evoked P300 signal is used to spell characters. The subject is presented with a 6 by 6 character matrix (see Figure 2) and he focuses his attention on a target character he wants to spell. All rows and columns in this matrix are intensified successively and randomly but separately. Two out of twelve intensifications contain the target character, i.e., one target row and one target column. As a result, the target row/column intensification becomes a rare stimulus to the subject. A P300 signal is then evoked by the rare stimulus. By detecting the P300 signal, we can infer which row or column the subject is focused on. By combing the row and column positions, we can infer the target character position. Assume that one epoch includes 12 intensifications, in which there exist one target row intensification and one target column intensification. In practice, people use several consecutive epochs for the P300 speller to infer one target character, because it is hard to use only one epoch to correctly spell one target character [16].

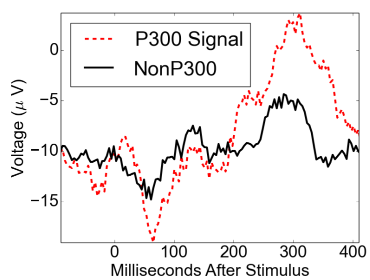

Fig. 1. P300 signal.

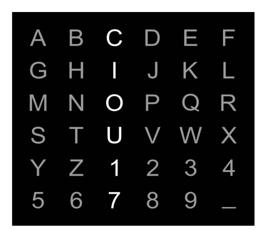

Fig. 2. P300 speller character matrix.

B. Datasets

This paper uses three benchmark datasets, namely, BCI Competition II - Data set IIb [17] as well as BCI Competition III - Data set II Subject A and Subject B [18]. In this paper, we use II to denote BCI Competition II - Data set IIb, III-A to denote BCI Competition III - Data set II Subject A, and III-B to denote BCI Competition III - Data set II Subject B. Here, we give a short description of the three datasets.

Dataset II, III-A and III-B are provided by the Wadsworth Center, NYS Department of Health. They are recorded from three different subjects with the BCI2000 platform, using the P300 speller method described in Section III-A. EEG brain signals are collected from 64 sensors. The brain signals are sampled at a frequency of 240Hz. One intensification lasts for 100ms, followed by a 75ms blank period for the matrix. The experiment uses 15 epochs for each character. After each sequence of 15 epochs, the matrix is blank for 2.5s, to inform the subject that this character is completed and to focus on the next character. In all three datasets, there are two separate sub-datasets. We use the first sub-dataset for sensor selection and we call this sub-dataset the preliminary dataset. We use the second sub-dataset for evaluation of the performance of the P300 speller after sensor selection and we call this sub-dataset the evaluation dataset. In Dataset II, the preliminary dataset has 42 characters and the evaluation dataset has 31 characters. In Dataset III-A and III-B, each preliminary dataset has 85 characters and each evaluation dataset has 100 characters.

IV. Our Sensor Selection Method

In this section, we present our novel iterative sensor selection method for the P300 speller. We call it Spatial Learning based Elimination Selection (SLES).

A. Spatial Learning based Elimination Selection

Our SLES method is described in Algorithm 1. The symbols used in Algorithm 1 and their corresponding descriptions are listed in Table I. The input of SLES is the initial sensor set S and the parameter N_s. The output of SLES is a set of selected sensor subsets SUB. For each iteration in Algorithm 1, SLES trains $SCNN_{(S)}$ (described in Section IV-B) with the input signals recorded with the sensors in sensor set S (see Line 2 in Algorithm 1). After training $SCNN_{(S)}$, the ranking scores $score_j$ for all sensors s_j in sensor set S are calculated (Line 3-4) using $SCNN_{(S)}$ and Equation (2) explained in Section IV-C. The sensor with the minimal score is found and removed from sensor set S (Lines 6-7). This reduced sensor set S is the selected sensor subset in this iteration (Line 8). The input parameter N_s controls the training frequency of $SCNN_{(S)}$ (Line 1) and the number of sensors to eliminate after training $SCNN_{(S)}$ (Line 5).

TABLE I
THE SYMBOLS USED IN ALGORITHM 1.

Symbol	Description
S	Sensor set.
s_j	The jth sensor in S.
C	Number of sensors in the initial sensor set.
SUB	A set of selected sensor subsets.
sub_m	A selected sensor subset with m sensors.
$SCNN_{(S)}$	The novel parametric CNN given in Section IV-B
N_s	Number of sensors to eliminate in an iteration.
$score_j$	The ranking score for s_j.
s_{remove}	The sensor to remove.

Algorithm 1: Proposed SLES algorithm.

Input: A set of $S = \{s_1, s_2, ..., s_j, ... s_C\}$, N_s;
Output: A set of $SUB = \{sub_1, sub_2, ... , sub_m, ... sub_{C-1}\}$;

1 **for** $1 \leq k \leq C/N_s$ **do**
2   Train a $SCNN_{(S)}$ with the input signals recorded using S;
3   **for** $s_j \in S$ **do**
4   Calculate $score_j$ using $SCNN_{(S)}$ and Equation (2);
5   **for** $1 \leq m \leq N_s$ **do**
6         $s_{remove} = \underset{s_j \in S}{argmin} \{score_j\}$;
7         $S \leftarrow S - s_{remove}$;
8         $sub_{(C-N_s*(k-1)-m)} \leftarrow S$;

B. Spatial Convolutional Neural Network

In this section, we introduce our novel parametric Spatial Convolutional Neural Network ($SCNN_{(S)}$), which weights are used in SLES to calculate the ranking scores of sensors.

1) Input Tensor: The input to $SCNN_{(S)}$ is the tensor ($N \times |S|$) shown in Figure 3. S is the sensor set used in Algorithm 1. x_{ji} denotes the ith temporal signal sample in the time domain and this signal sample is recorded with sensor s_j in sensor set S in the space domain. $SCNN_{(S)}$ is parameterized by S because the input tensor to $SCNN_{(S)}$ is constructed by the EEG signals samples acquired using the sensors in sensor set S and S is changed in each main iteration of Algorithm 1. N denotes the number of temporal signal samples. Here $N= T_s \times F_s$, where T_s denotes the time period between 0 and T_s posterior to the beginning of each row/column intensification (see Section III-A) and F_s denotes the signal sampling frequency. In the input tensor, the temporal signal samples are bandpass filtered between 0.1Hz and 20Hz to remove high frequency noise. Then, the temporal signal samples are normalized to have zero mean and unit variance based on each individual pattern and for each sensor. Each individual pattern represents N signal samples in the time period between 0 and T_s posterior to the beginning of each intensification.

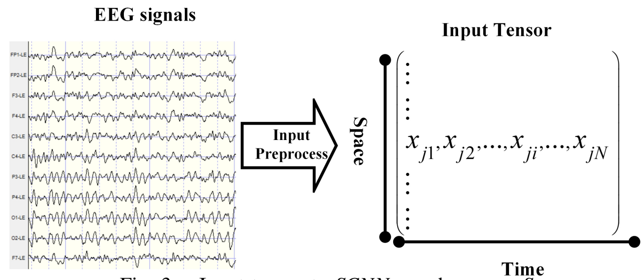

Fig. 3. Input tensor to $SCNN_{(S)}$, where $s_j \in S$.

2) Spatial Convolutional Neural Network: In our proposed $SCNN_{(S)}$, there are two layers in total. Table II shows the details of the $SCNN_{(S)}$ architecture. The first column shows the name of the layers. The second column shows the operation performed in the corresponding layer. The third column shows the kernel size in the convolution layer. The fourth column shows how many feature maps/neurons are employed in the convolution/fully-connected layer.

TABLE II
$SCNN_{(S)}$ ARCHITECTURE.

Layer	Operation	Kernel Size	Feature Maps/Neurons
1	Convolution	(1,$\|S\|$)	10
	Dropout	—	—
2	Fully-Connected	—	2

In Layer 1, $SCNN_{(S)}$ performs a spatial convolution operation to extract the spatial features related to P300 signals from the input tensor. The detailed calculation in this convolution operation is shown in Equation (1), where f_{ki} denotes the ith datum in the kth feature map. w_{kj} denotes the jth weight of the filter and this filter outputs abstract data for the kth feature map. The activation function we employ in this layer is the Rectified Linear Unit (ReLU). In this layer, we employ Dropout in order to prevent the network from overfitting. In this layer, we do not use bias in the convolution operation, thus all the learned features are captured by the weights w_{kj}. This layer outputs 10 feature maps in total. These generated feature maps are the input to Layer 2.

$$f_{ki} = \sum_{s_j \in S} x_{ji} w_{kj} \quad (1) \qquad score_j = \sum_{k=1}^{10} |w_{kj}| \quad (2)$$

In Layer 2, $SCNN_{(S)}$ performs the fully-connected operation. This layer has two neurons in total. The fully-connected operation makes correlation between the feature maps generated in Layer 1 and the two neurons. The two neurons represent two classes. One class represents “P300”, denoting the presence of a P300 signal. The other class represents “non-P300”, denoting the absence of a P300 signal. The activation function employed in this layer is the Softmax function.

3) Training: The training of $SCNN_{(S)}$ is carried out by minimizing the binary cross-entropy loss function. It uses Stochastic Gradient Descent as an optimizer with momentum and weight decay. The learning rate is set to 0.01. The momentum is set to 0.9. The batch size is set to 128.

The weight decay is set to 0.0005. The weights of all neurons in the single convolution layer are regularized by L2 Regularizer.

C. Ranking Function

Our proposed novel ranking function used in SLES is described in Equation (2), where $score_j$ is the ranking score for sensor s_j used in Algorithm 1. w_{kj} are the weights described in Equation (1). These weights are obtained from the trained $SCNN_{(S)}$, described in Section IV-B.2 and used in Algorithm 1. Note that we take the absolute value of the weights in Equation (2) because weights with a large negative value also indicate that the corresponding sensors are important in sensor set S.

V. Experimental Evaluation

In this section, we present the experiments, we have performed, in order to compare the minimal number of sensors selected by our method and other methods to acquire EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. We first introduce our experimental setup and then we present and analyse the obtained experimental results.

A. Experimental Setup

We use two different implementations of the P300 speller to perform the experiments: one uses the CNN-based classifier OCLNN [16] and the other uses the SVM-based classifier ESVM [7]. We want to confirm the robustness of our SLES method by showing that our method is effective for different P300 speller implementations.

We compare our SLES method with 12 other sensor selection methods. These methods are summarized in Table III. In this table, the first row gives the name of the different methods. The second row describes the sensor elimination algorithms used in the methods, where BE-1 denotes a backward elimination algorithm which eliminates one sensor at a time; BE-4 denotes a backward elimination algorithm which eliminates 4 sensors at a time; "–" denotes that the corresponding method does not use a backward elimination algorithm. The last row indicates the ranking functions used in the methods, where P300 denotes the P300 detection accuracy; Char denotes the average character spelling accuracy across all epochs; AUC denotes Area Under the Receiver Operating Characteristic [11]; C_{cs} denotes the ranking score proposed in [7]; SSNR denotes Signal to Signal and Noise Ratio [10]; CCNN and BN3 denote that the corresponding method selects sensors by analysing the weights obtained from the trained networks CCNN [12] and BN3 [13], respectively.

TABLE III
METHODS COMPARED WITH SLES.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Algo.	BE-1	BE-1	BE-1	BE-1	BE-1	BE-4	BE-4	BE-4	BE-4	BE-4	–	–
Function	P300	Char	AUC	C_{cs}	SSNR	P300	Char	AUC	C_{cs}	SSNR	CCNN	BN3

We compare the minimal number of sensors selected by the different methods to acquire EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. We use the preliminary dataset of Dataset II, III-A and III-B described in Section III-B to perform sensor selection using the different sensor selection methods to select sensor subsets for the corresponding subject. Then, we use the evaluation dataset of Dataset II, III-A and III-B to evaluate the spelling accuracy of the aforementioned P300 speller implementations with the selected sensor subsets. The spelling accuracy is calculated using Equation (3). In this equation, acc_k^m denotes the spelling accuracy when using the first k epochs for each character and using the EEG signals acquired with the selected sensor subset containing m number of sensors. N_k^m denotes the number of truly predicted characters when using the first k epochs for each character and using the EEG signals acquired with the selected sensor subset containing m number of sensors, and M denotes the number of all characters in the evaluation dataset. After the evaluation of the spelling accuracy, the minimal number of sensors needed to acquire EEG signals for epoch k is calculated as m_{min}, where m_{min} is the minimal $m \in [1,63]$ which makes $acc_k^m >= acc_k^{64}$.

$$acc_k^m = \frac{N_k^m}{M} \tag{3}$$

The setup for our SLES algorithm (Algorithm 1) is the following. The input to SLES is $S = \{s_1,\ s_2,\ ...,\ s_j,\ ...\ s_C\}$ and N_s. We set C=64 because the datasets used in the experiments are recorded with 64 sensors. We set N_s=4. For detailed discussion why N_s=4 see Section VI-A. SLES uses $SCNN_{(S)}$ as the ranking function. $SCNN_{(S)}$ uses the input tensor $(N \times |S|)$. $N = T_s \times F_s = 240$, where $F_s = 240$ Hz and we set $T_s = 1000$ms because we take each individual pattern to be the signal samples between 0 and 1000 ms posterior to the beginning of each intensification.

B. Experimental Results

Table IV and Table V show the minimal number of sensors selected by the different sensor selection methods to acquire EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set of all 64 sensors is used. The first column in the tables lists the different selection methods we compare. Each row provides the minimal number of sensors selected by a method to acquire EEG signals for different epoch numbers $k \in [1,15]$. A number in bold indicates that the minimal number of sensors selected by the corresponding method is the lowest among all methods. Overall, the minimal number of sensors selected by our SLES method is lower than the minimal number of sensors selected by all other methods in most cases. SLES is able to reduce the minimal number of sensors selected by other methods with up to 44 sensors.

For the P300 speller with the CNN-based classifier (see Table IV), in 42 out of 45 cases, the minimal number of sensors selected by our SLES is lower than the minimal number of sensors selected by all other method. Our SLES is able to reduce the minimal number of sensors selected by other methods with up to 44 sensors. The largest reduction occurs when comparing the minimal number of sensors selected by SLES with the minimal number of sensors selected by C_8 on epoch number $k = 7$ for Dataset III-A.

For the P300 speller with the SVM-based classifier (see Table V), in 41 out of 45 cases, the minimal number of sensors selected by our SLES is lower than the minimal number of sensors selected by all other methods. Our SLES is able to reduce the minimal number of sensors selected by other methods with up to 40 sensors. The largest reduction occurs when comparing the minimal number of sensors selected by SLES with the minimal number of sensors selected by C_{12} on epoch number $k = 2$ for Dataset III-B.

Finally, our SLES method is robust because: 1) SLES is effective in reducing the number of sensors when the P300 speller is implemented with different classifiers. From Table IV and Table V, we can see that no matter the P300 speller is implemented with CNN-based classifier or SVM-based classifier, the minimal number of sensors selected by SLES is lower than the minimal number of sensors selected by all other methods in most cases; 2) SLES is effective when our method is used for different subjects, i.e., no matter that SLES is used with Dataset III-A, Dataset III-B or Dataset II, the minimal number of sensors selected by SLES is lower than the minimal number of sensors selected by all other methods in most cases.

VI. Discussions

In this section, we discuss the configuration of input parameter N_s in SLES (see Algorithm 1). Also, we discuss

TABLE IV

MINIMAL NUMBER OF SENSORS SELECTED BY DIFFERENT METHODS FOR DATASET III-A, III-B AND II. THE P300 SPELLER IS IMPLEMENTED USING THE CNN-BASED CLASSIFIER OCLNN [16]. SL DENOTES OUR PROPOSED METHOD SLES.

Dataset III-A

	Epochs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SL	**17**	**50**	**50**	**25**	**40**	**35**	**20**	**50**	**60**	**20**	**50**	**35**	**21**	**16**	35
C_1	27	64	58	59	55	64	43	58	64	29	57	49	29	29	53
C_2	29	64	64	60	59	59	36	60	**60**	31	56	56	31	28	60
C_3	28	64	64	60	64	60	55	64	64	36	64	55	39	39	46
C_4	53	64	63	64	64	64	60	64	61	61	64	56	55	37	61
C_5	30	64	56	64	55	64	43	63	64	31	64	57	36	33	55
C_6	30	63	64	64	64	64	57	64	64	61	64	53	53	53	59
C_7	44	57	64	64	64	64	62	64	64	55	64	53	59	52	51
C_8	49	59	60	63	64	64	64	63	64	60	64	56	56	54	56
C_9	22	64	56	55	58	64	56	56	64	34	52	48	27	36	**34**
C_{10}	34	63	64	64	64	64	59	64	64	61	64	58	61	53	59
C_{11}	25	64	56	55	58	64	56	56	64	34	54	48	27	36	39
C_{12}	28	64	64	56	64	64	58	56	64	41	64	53	39	36	41

Dataset III-B

	Epochs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SL	**25**	**17**	**31**	**27**	**26**	**21**	**27**	**32**	**37**	**32**	**31**	**29**	**34**	**29**	**23**
C_1	32	23	**31**	28	31	24	32	40	43	44	38	32	42	35	26
C_2	28	24	52	38	36	25	35	45	41	41	**31**	30	42	38	31
C_3	26	29	42	50	35	23	32	54	49	48	38	**29**	39	38	33
C_4	31	25	61	40	45	24	38	57	60	57	59	55	49	47	43
C_5	44	33	36	34	31	24	34	41	43	48	38	42	42	37	26
C_6	50	56	55	49	48	58	48	50	50	50	48	44	51	49	45
C_7	52	48	49	49	50	49	49	59	48	48	48	46	45	47	46
C_8	49	49	54	52	49	54	48	59	49	49	49	44	46	44	44
C_9	44	56	49	49	40	25	35	39	39	49	51	37	**34**	35	42
C_{10}	49	52	55	44	50	51	46	50	52	51	49	50	51	49	50
C_{11}	44	56	49	49	40	25	35	49	39	49	51	37	38	34	42
C_{12}	54	61	59	55	52	27	30	42	45	63	54	48	47	34	42

Dataset II

	Epochs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SL	**11**	**11**	**46**	**15**	**22**	11	**9**	**3**	**3**	**3**	**3**	4	**3**	**3**	**3**
C_1	32	30	47	39	38	22	16	6	6	6	6	6	6	7	6
C_2	17	20	55	19	34	15	12	6	5	3	4	**3**	6	7	7
C_3	18	18	47	18	30	10	12	6	5	**3**	5	6	6	6	6
C_4	24	27	49	17	28	**8**	10	7	6	7	4	**3**	**3**	5	6
C_5	40	33	48	41	38	35	20	9	8	9	9	8	9	9	8
C_6	48	33	49	47	32	30	28	28	20	15	10	20	10	15	15
C_7	44	32	49	48	31	27	27	27	22	18	10	10	8	10	10
C_8	44	36	50	38	33	32	25	25	10	10	10	10	10	17	12
C_9	45	35	44	34	34	34	25	25	10	17	18	17	15	15	15
C_{10}	48	35	49	44	40	30	33	29	21	19	20	20	20	18	17
C_{11}	25	25	54	24	24	14	15	15	10	10	12	17	15	15	15
C_{12}	29	27	59	25	31	22	15	18	13	13	15	18	19	21	18

TABLE V

COMPARISON OF THE MINIMAL NUMBER OF SENSORS SELECTED BY DIFFERENT METHODS FOR DATASET III-A, III-B AND II. THE P300 SPELLER IS IMPLEMENTED USING THE SVM-BASED CLASSIFIER ESVM [7]. SL DENOTES OUR PROPOSED METHOD SLES.

Dataset III-A

	Epochs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SL	**19**	**44**	**54**	**38**	**40**	**42**	33	56	**58**	**39**	**44**	**27**	**35**	**21**	**36**
C_1	31	56	58	49	45	57	44	**54**	64	43	55	51	46	32	56
C_2	32	53	57	48	44	59	47	55	64	43	50	53	47	32	55
C_3	29	53	**54**	48	47	56	35	55	63	45	52	57	42	36	49
C_4	56	50	53	44	42	55	**31**	**54**	60	56	49	51	43	37	57
C_5	31	55	58	50	45	49	43	60	64	46	58	48	47	33	57
C_6	35	60	61	57	54	56	52	64	64	53	62	59	57	50	62
C_7	43	59	61	55	49	57	52	64	64	50	62	58	55	50	62
C_8	47	54	64	57	48	54	55	62	64	59	57	52	56	49	63
C_9	33	55	60	51	48	62	59	63	64	56	53	54	39	46	49
C_{10}	33	62	62	59	53	57	57	64	64	54	61	60	59	52	55
C_{11}	29	61	59	54	55	59	56	58	64	46	53	45	41	40	37
C_{12}	35	64	63	56	59	61	60	59	64	51	58	47	44	46	43

Dataset III-B

	Epochs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SL	**18**	**19**	**34**	**29**	**17**	31	**39**	**41**	**22**	**36**	**24**	33	**33**	**26**	**27**
C_1	28	27	43	39	32	35	45	50	36	51	36	**30**	45	37	29
C_2	27	25	42	38	29	31	**39**	54	36	49	40	**30**	44	30	**27**
C_3	24	31	45	35	26	30	42	49	34	44	38	32	47	**26**	**27**
C_4	23	26	59	36	27	**29**	44	52	35	42	37	36	39	37	30
C_5	35	29	44	40	29	33	48	50	39	49	38	33	43	40	31
C_6	49	58	56	51	44	40	57	61	48	59	46	42	57	51	47
C_7	41	52	53	48	46	48	53	63	50	54	49	44	52	47	44
C_8	40	50	53	49	47	50	49	60	47	50	41	44	51	50	46
C_9	42	51	48	47	41	37	52	61	43	49	45	39	46	44	45
C_{10}	42	54	51	52	38	43	57	61	44	55	50	46	53	50	49
C_{11}	43	55	49	47	34	33	45	53	40	52	51	39	49	29	46
C_{12}	52	59	56	61	46	41	51	64	49	64	56	41	60	30	46

Dataset II

	Epochs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SL	**13**	**10**	**36**	**20**	**31**	**16**	**8**	**4**	**3**	**5**	**3**	**5**	**3**	**6**	**3**
C_1	30	32	42	38	43	28	15	8	6	9	7	10	6	8	6
C_2	25	28	40	29	40	19	16	9	6	11	**3**	11	7	**6**	8
C_3	21	23	38	30	34	25	13	8	7	10	6	14	7	7	7
C_4	26	24	**36**	33	38	18	14	10	8	12	5	13	4	9	9
C_5	43	29	37	40	41	36	24	11	9	13	8	12	10	16	11
C_6	50	38	52	50	50	33	26	30	18	19	11	24	12	14	16
C_7	47	40	55	48	49	29	29	29	16	21	13	23	11	17	13
C_8	50	37	54	42	46	31	30	29	16	23	13	20	16	16	19
C_9	51	35	49	39	54	39	29	31	19	25	14	19	13	19	17
C_{10}	49	44	50	40	50	30	31	26	20	21	19	19	19	20	16
C_{11}	28	30	56	30	36	19	17	16	9	20	16	19	16	17	12
C_{12}	31	33	61	36	38	23	18	20	11	24	20	21	21	26	14

the impact of different CNN architectures on extracting P300-related features and on selecting sensors.

A. Configuration of N_s in SLES

In order to configure the input parameter N_s in SLES, we use the preliminary dataset of Dataset III-A to perform experiments and tune N_s. The P300 speller implemention used for this experiment is the CNN-based classifier OCLNN [16]. We use 60% of this preliminary dataset to train $SCNN_{(S)}$ (see Section IV-B) while running SLES with different N_s configurations, i.e., N_s=1, 2, 4, 8, 16, 32 and 64. With each N_s configuration, SLES selects a set of sensor subsets for Dataset III-A. We use the left 40% of this preliminary dataset to evaluate the spelling accuracy of aforementioned P300 speller implementation with the selected sensor subsets. The P300 spelling accuracy is calculated using Equation (3). Then, we calculate the minimal number of sensors m_{min} for the different N_s configurations as described in Section V-A.

TABLE VI

MINIMAL NUMBER OF SENSORS SELECTED BY SLES WITH DIFFERENT N_s CONFIGURATIONS.

	Epochs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
N_s=1	27	56	58	57	50	60	44	53	63	30	55	33	24	19	46
N_s=2	18	**45**	52	34	43	46	28	**48**	58	26	47	30	20	16	**36**
N_s=4	**15**	46	**49**	**28**	**40**	**36**	**18**	**48**	59	**21**	**46**	**27**	**19**	**13**	38
N_s=8	19	54	**49**	31	41	48	26	**48**	**57**	25	**46**	34	20	**13**	37
N_s=16	22	54	55	42	45	56	39	53	63	29	53	35	25	20	37
N_s=32	25	59	57	47	49	58	48	54	63	32	54	37	28	23	41
N_s=64	27	60	60	48	50	60	55	55	64	36	59	42	31	24	44

The experimental results are shown in Table VI. The first column in the table lists the different configurations of N_s in SLES. Each row provides the minimal number of sensors selected by SLES for different epoch numbers $k \in [1, 15]$. A number in bold indicates the minimal number of sensors selected by SLES with the corresponding N_s configuration is the lowest compared with the minimal number of sensors selected by SLES with other N_s configurations. From Table VI, we can see that, in most cases, the minimal number of sensors selected by SLES with N_s=4 is the lowest compared to the minimal number of sensors selected by SLES with other N_s configurations. Therefore, we set N_s=4 when using SLES.

B. Exploring the Impact of CNN Architecture on Extracting P300-related Features

We perform experiments to explore the impact of different CNN architectures on extracting useful P300-related features. We implement three P300 spellers. The first P300 speller implementation uses our $SCNN_{(S)}$ (see Section IV-B.2) as the classifier. The second P300 speller implementation uses CCNN [12] as the classifier. The third P300 speller implementation uses BN3 [13] as the classifier. Then, we evaluate the spelling accuracy of the aforementioned P300 spellers using Dataset III-A with all the 64 sensors.

The experimental results are shown in Figure 4. From this figure, we can see that $SCNN_{(S)}$ and BN3 achieve similar spelling accuracy across different epoch numbers $k \in [1, 15]$. CCNN achieves lower spelling accuracy on smaller epoch numbers and it achieves similar spelling accuracy with $SCNN_{(S)}$ and BN3 on larger epoch numbers. We can conclude from this experiment that our $SCNN_{(S)}$ extracts similar useful P300-related features as BN3 and CCNN.

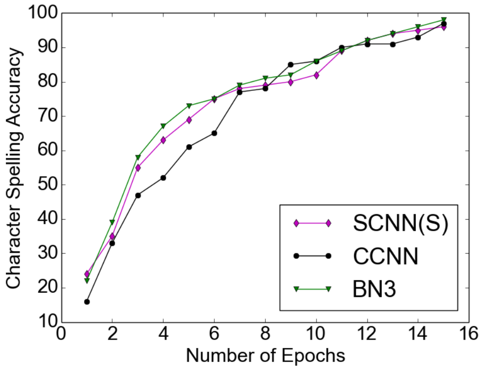

Fig. 4. Spelling accuracy of SCNN, CCNN and BN3.

C. Exploring the Impact of CNN Architecture on Sensor Selection

We perform experiments to explore the impact of different CNN architectures on the sensor selection process. We introduced this issue in Section II. The P300 speller implemention used for this experiment is the CNN-based classifier OCLNN. We use the preliminary dataset of Dataset III-A to train our $SCNN_{(S)}$ (see Section IV-B), CCNN [12] and BN3 [13]. We select sensor subsets by directly analysing the weights of the spatial convolution layer of our $SCNN_{(S)}$, CCNN and BN3. We use the evaluation dataset of Dataset III-A to evaluate the P300 spelling accuracy of the aforementioned P300 speller implementation with the selected sensor subsets. Then, we calculate the minimal number of sensors m_{min} selected by analysing the weights of our $SCNN_{(S)}$, CCNN and BN3. For the detailed calculation of m_{min} see Section V-A.

TABLE VII
MINIMAL NUMBER OF SENSORS SELECTED BY ANALYSING DIFFERENT CNNS.

	Epochs														
CNN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$SCNN_{(S)}$	**23**	**60**	**55**	**49**	**51**	**60**	**53**	**55**	**64**	**31**	**53**	**46**	**27**	**32**	**39**
CCNN	25	64	56	55	58	64	56	56	**64**	34	54	48	**27**	36	**39**
BN3	28	64	64	56	64	64	58	56	**64**	41	64	53	39	36	41

The experimental results are shown in Table VII. The first column in the table lists the different CNNs. Each row provides the minimal number of sensors selected by analysing the weights of different CNNs for different epoch numbers $k \in [1, 15]$. A number in bold indicates that the minimal number of sensors selected by analysing the weights of the corresponding CNN is the lowest, compared to the minimal number of sensors selected by analysing the weights of other CNNs. Table VII shows that the minimal number of sensors selected by analysing the weights of our $SCNN_{(S)}$ is the lowest, compared to the minimal number of sensors selected by analysing the weights of CCNN and BN3. The reason is the following. CCNN and BN3 have multiple convolution layers and only the weights of the first layer is used for analysis and sensor selection. Unfortunately, the information needed for proper sensor selection is distributed over the weights of all convolution layers. Therefore, CCNN and BN3 do not use all the information available for proper sensor selection. In contrast, our $SCNN_{(S)}$ has only one convolution layer and this layer performs the spatial convolution operation. All the information needed for sensor selection is captured by the weights of this single spatial convolution layer. Moreover, our $SCNN_{(S)}$ achieves similar spelling accuracy in comparison to CCNN and BN3 (see Section VI-B), meaning that our CNN has similar ability to extract useful P300-related features. We analyse the weights of the single spatial convolution layer in our $SCNN_{(S)}$ to select sensors. Thus, our $SCNN_{(S)}$ uses all the information available for proper sensor selection compared to CCNN and BN3. As a result, $SCNN_{(S)}$ can select more appropriate sensor subsets and further reduce the minimal number of sensors needed to acquire brain signals without losing spelling accuracy.

VII. CONCLUSIONS

In this paper, we propose a novel sensor selection method, called SLES, for reducing the number of sensors needed to acquire EEG signals for a P300 speller without losing spelling accuracy. SLES uses an iterative parametric backward elimination algorithm to eliminate and select sensors and it uses our novel $SCNN_{(S)}$ as a ranking function to evaluate the importance of a sensor. Our SLES is also robust across different P300 speller implementations and different subjects. Experimental results show that the minimal number of sensors selected by our SLES method is lower than the minimal number of sensors selected by other methods in most cases. Therefore, our SLES can further reduce the cost and power consumption of the P300 speller, thereby promoting P300 spellers into people's daily life.

REFERENCES

[1] 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.

[2] S. W. Gilroy, J. Porteous, *et al.*, "A brain-computer interface to a plan-based narrative." in *International Joint Conference on Artificial Intelligence*, 2013, pp. 1997–2005.

[3] C.-T. Lin, S.-F. Tsai, and L.-W. Ko, "EEG-based learning system for online motion sickness level estimation in a dynamic vehicle environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1689–1700, 2013.

[4] C.-T. Lin, B.-S. Lin, *et al.*, "Brain computer interface-based smart living environmental auto-adjustment control system in UPnP home networking," *IEEE Systems Journal*, vol. 8, no. 2, pp. 363–370, 2014.

[5] D. Zhang, L. Yao, X. Zhang, S. Wang, W. Chen, and R. Boots, "Cascade and parallel convolutional recurrent neural networks on EEG-Based intention recognition for brain computer interface," in *AAAI*, 2018.

[6] R. Fazel-Rezai, B. Z. Allison, C. Guger, E. W. Sellers, S. C. Kleih, and A. Kübler, "P300 brain computer interface: current challenges and emerging trends," *Frontiers in neuroengineering*, vol. 5, p. 14, 2012.

[7] A. Rakotomamonjy and V. Guigue, "Bci competition iii: dataset ii-ensemble of svms for bci p300 speller," *IEEE transactions on biomedical engineering*, vol. 55, no. 3, pp. 1147–1154, 2008.

[8] B. Rivet, A. Souloumiac, G. Gibert, V. Attina, and O. Bertrand, "Sensor selection for P300 speller brain computer interface," in *ESANN*, 2009.

[9] H. Cecotti, B. Rivet, M. Congedo, C. Jutten, O. Bertrand, E. Maby, and J. Mattout, "Suboptimal sensor subset evaluation in a P300 brain-computer interface," in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 924–928.

[10] H. Cecotti et al., "A robust sensor-selection method for P300 brain–computer interfaces," *Journal of neural engineering*, vol. 8, no. 1, p. 016001, 2011.

[11] K. Colwell, D. Ryan, C. Throckmorton, E. Sellers, and L. Collins, "Channel selection methods for the P300 speller," *Journal of neuroscience methods*, vol. 232, pp. 6–15, 2014.

[12] H. Cecotti and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 433–445, 2011.

[13] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on batch normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

[14] W. Speier, C. Arnold, N. Chandravadia, D. Roberts, S. Pendekanti, and N. Pouratian, "Improving p300 spelling rate using language models and predictive spelling," *Brain-Computer Interfaces*, vol. 5, no. 1, pp. 13–22, 2018.

[15] V. Guy, M.-H. Soriani, M. Bruno, T. Papadopoulo, C. Desnuelle, and M. Clerc, "Brain computer interface with the p300 speller: Usability for disabled people with amyotrophic lateral sclerosis." *Annals of physical and rehabilitation medicine*, vol. 61, no. 1, pp. 5–11, 2018.

[16] H. Shan, Y. Liu, and T. Stefanov, "A simple convolutional neural network for accurate P300 detection and character spelling in brain computer interface." in *27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 1604–1610.

[17] B. Blankertz and K.-R. Müller et al., "BCI competition II webpage," *http://www.bbci.de/competition/ii/*, 2003.

[18] B. Blankertz et al., "BCI competition III webpage," *http://www.bbci.de/competition/iii/*, 2008.