EEG-Based Adaptive Driver-Vehicle Interface Using Variational Autoencoder and PI-TSVM

Luzheng Bi*, *Senior Member, IEEE*, Jingwei Zhang, and Jinling Lian, *Student Member, IEEE*

***Abstract*— Event-related potential (ERP)-based driver-vehicle interfaces (DVIs) have been developed to provide a communication channel for people with disabilities to drive a vehicle. However, they require a tedious and time-consuming training procedure to build the decoding model, which can translate EEG signals into commands. In this paper, to address this problem, we propose an adaptive DVI by using a new semi-supervised algorithm. The decoding model of the proposed DVI is first built with a small labeled training set, and then gradually improved by updating the proposed semi-supervised decoding model with new collected unlabeled EEG signals. In our semi-supervised algorithm, independent component analysis (ICA) and Kalman smoother are first used to improve the signal-to-noise ratio (SNR). After that, variational autoencoder is applied to provide a robust feature representation of EEG signals. Finally, a prior information-based transductive support vector machine (PI-TSVM) classifier is developed to translate these features into commands. Experimental results show that the proposed DVI can significantly reduce the training effort. After a short updating, its performance can be close to that of the supervised DVI requiring a lengthy training procedure. This work is vital for advancing the application of these DVIs.**

***Index Terms*—brain-controlled vehicle, EEG, semi-supervised learning, variational autoencoder, transductive support vector machine.**

I. Introduction

Vehicles controlled by brain signals but not limbs (known as brain-controlled vehicles) can provide a way for people with disabilities to improve their mobility [1-3]. A driver-vehicle interface (DVI) based on electroencephalogram (EEG) signals is the core part of a brain-controlled vehicle, which generally includes stimuli interface and EEG decoding module translating EEG signals into commands and sending these translated commands to vehicles. Since the stimuli interface may obstruct the visual field of drivers, it is usually projected onto the windshield of the vehicles via a head-up-display (HUD) system, which enables the drivers to change their attention between the interface and road quickly [3], [4].

According to the function of output commands, EEG-based DVIs can be divided into two categories: task-level DVIs based on EEG signals and servo-level DVIs based on EEG signals. For task-level DVIs based on EEG signals, users first use a DVI to choose a task from a predefined list. Then, the selected task is sent to an automatic control system, which is responsible for performing the selected task [4-6]. For example, the drivers can first use such DVI to select their desired destinations from predefined ones. Then an autonomous driving system takes the responsibility to reach the desired destination. For servo-level DVIs based on EEG signals, drivers can use them to output motion control commands (such as direction and speed control commands) to directly control a vehicle [3], [7], [8].

Both classes of DVIs require building a decoding model of EEG signals, which can translate EEG signals into commands. Since EEG signals are nonstationary and have low signal-to-noise-ratio (SNR) owing to artifacts, such as electromyogram (EMG) and electrooculogram (EOG) signals [9-13], the decoding model usually requires substantial training data to obtain a good performance [14], [15]. Therefore, for using brain-controlled vehicles, brain-controlled-vehicle drivers have to carry out a time-consuming training process to obtain enough training data needed by building the decoding model. This long training process is tiring for the users and reduces the attractiveness of EEG-based DVIs. Thus, it is vital to reduce the training cost while not decrease the system performance for the application of EEG-based DVIs. One possible way to reduce the training cost of EEG-based DVIs is to first initialize the decoding model with a small labeled training set and then update the decoding model with new collected unlabeled EEG signals by using a semi-supervised algorithm.

However, to the best of our knowledge, no studies have explored how to develop semi-supervised algorithms to build an EEG-based DVI, although semi-supervised algorithms have been proposed and applied to event-related potential (ERP)-based spellers [16-19].

Compared to an ERP-based speller, an ERP-based DVI needs to send commands more quickly. That is, it needs to use fewer rounds of flashings to issue a command than an ERP speller, which makes ERP potentials more challenging to recognize. Thus, these semi-supervised algorithms applied to ERP spellers cannot be directly applied to an ERP-based DVI.

In this paper, we propose a novel semi-supervised decoding algorithm to develop an adaptive ERP-based DVI by combining variational autoencoder (VAE) with transductive support vector machine (TSVM). In the proposed semi-supervised algorithm,

This work was supported in part by National Natural Science Foundation of China under Grants 51575048 and 51975052. (*Corresponding author: Luzheng Bi*).

L. Bi, J. Zhang, and J. Lian are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081 China. (e-mail: bhxblz@bit.edu.cn, zhangjw1995@hotmail.com, and lianjinling@bit.edu.cn)

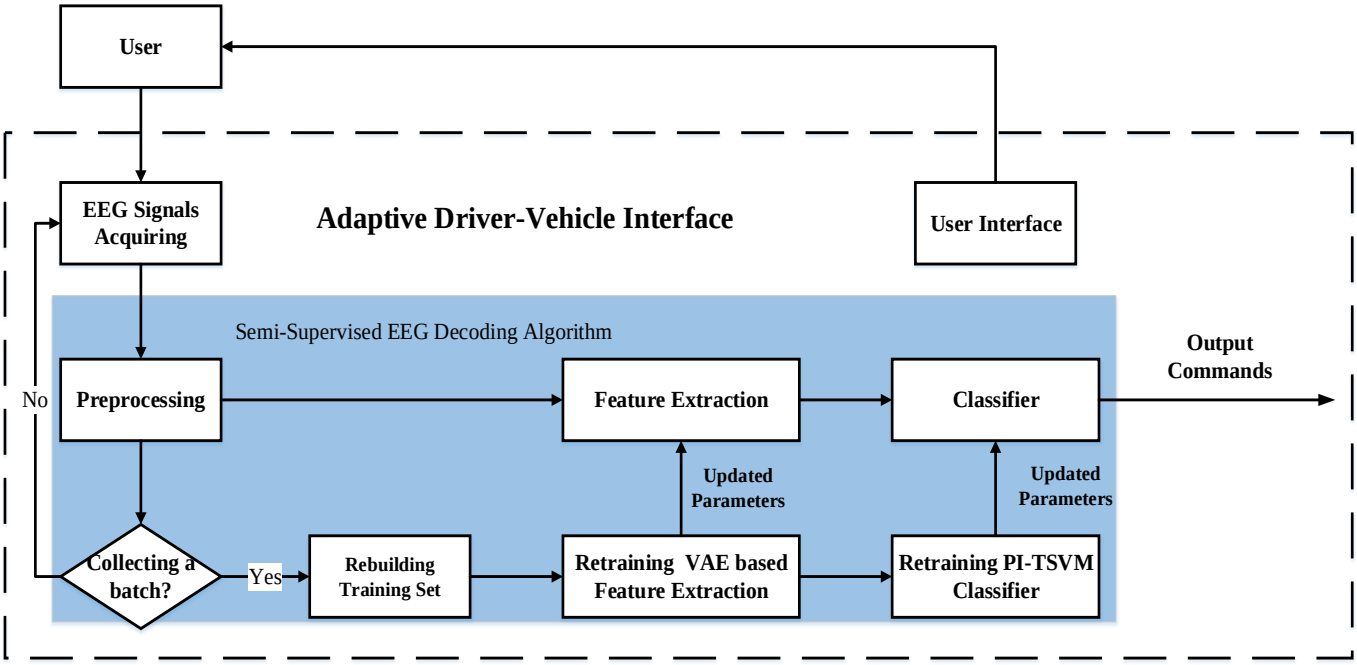

Fig. 1. Architecture of the adaptive driver-vehicle interface based on EEG

EEG signals are first preprocessed with the algorithm that we proposed in [20] to improve the SNR. After that, a deep generative model called variational autoencoder (VAE) [21] is used to provide a low dimensional and robust feature representation of the EEG signals. Finally, we improve the transductive support vector machine (TSVM) [22] by using prior information and employ the improved TSVM for classification.

The remainder of the paper is organized as follows. Section II introduces the proposed adaptive DVI including the semi-supervised decoding algorithm. Section III describes the experiment for the evaluation of the adaptive DVI. Section IV presents the experimental results and discussion. The conclusion of this paper is given in Section V.

II. Adaptive Driver-Vehicle Interface Based on EEG

A. System Architecture

As shown in Fig.1, the users select commands by focusing on the desired letter on the user interface. The DVI system uses the EEG decoding model to translate the EEG signals into commands.

The adaptive DVI first uses a small training set to initialize the EEG decoding model and then gets into online output command mode. In the online procedure, the DVI first uses the online collected unlabeled EEG signals to enrich the training set, and then retrain the EEG decoding model with the enriched training set.

To reduce the computational complexity, we update the training set in batches and retrain the decoding model after each update.

The adaptive process is as follows:

Step 1: Build an initial training dataset and train an initial EEG decoding model with the initial training set.

Step 2: Collect the new unlabeled EEG signals. If the new samples reach a certain amount, the collected EEG signals are added to the training dataset.

Step 3: Retrain the variational autoencoder and PI-TSVM classifier with the rebuilt training dataset to update the semi-supervised EEG decoding model.

Step 4: Use the updated EEG decoding model to replace the old one.

Step 5: Repeat Step 2 to Step 4.

B. User Interface

The interface is based on a new P300 paradigm that we have proposed in [8]. It can send rapid and accurate commands to interact with a brain-controlled vehicle.

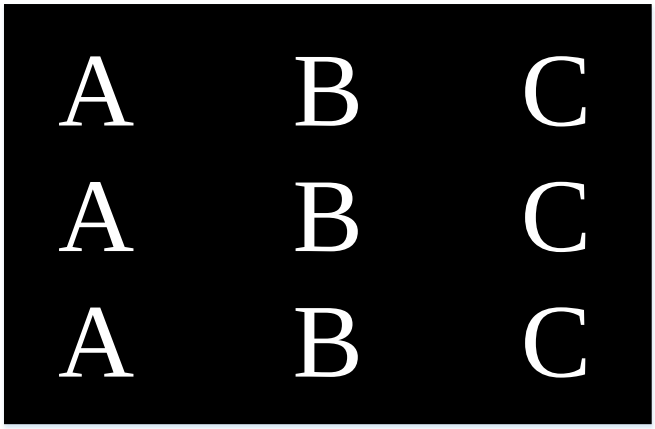

Fig. 2. User interface, where A, B and C correspond to a command, respectively.

As shown in Fig. 2, the interface has nine letters with every three same letters representing a command. When the interface works, each letter flashes in a random order for 120 ms without a break between two letters. Therefore, every round lasts 1.08s (120ms/letter*9 letter). The users generate a special EEG potential (event-related potential) by focusing on the desired

letter [23]. The DVI sends commands by detecting event-related potentials. Obviously, every command is associated with one target sample and two non-target samples, since the desired letters correspond to event-related potentials, while the undesired ones do not. To reduce the occlusion of the visual field by the user interface and to enable drivers to focus on the user interface while paying attention to the road. We displayed the user interface on the windshield of vehicles via a HUD system, as shown in Fig. 3.

Fig. 3. Head-up display system

Usually, the interface flashes several rounds and the EEG signals corresponding to the same letters are superimposed to make the EEG signals easier to recognize. To send commands quickly, we used only two rounds to send a command. The disadvantage of only using two rounds is that the EEG signals are feeble and difficult to recognize, which requires stronger preprocessing and feature extraction algorithms.

C. Data Collection

EEG signals were recorded by 16 Ag/AgCl electrodes from different standard locations (i.e. Cz, Fz, Pz, Oz, CPz, POz, F3, F4, C3, C4, P3, P4, P7, P8, O1, O2) based on a 10-20 BCI system. In each experiment trial, 512 ms EEG signals were segmented after every flashing stimulus. The sampling frequency of the signals was 1000Hz, and the power frequency interference was filtered by a notch filter.

D. Semi-Supervised Decoding Algorithm

In the proposed semi-supervised algorithm, EEG signals are first preprocessed with independent component analysis (ICA) and Kalman smoother to improve the SNR. After that, the VAE is used to provide a lower dimensional and robust feature representation of EEG signals. Finally, a PI-TSVM is utilized for classification.

1) Preprocessing

As shown in Fig. 4, an algorithm based on ICA and Kalman smoother are applied to improve the SNR of EEG signals in our previous work [20]. First, the sixteen-channel EEG signals are treated as an instantaneous linear mixture of sixteen underlying sources and are decomposed into sixteen independent

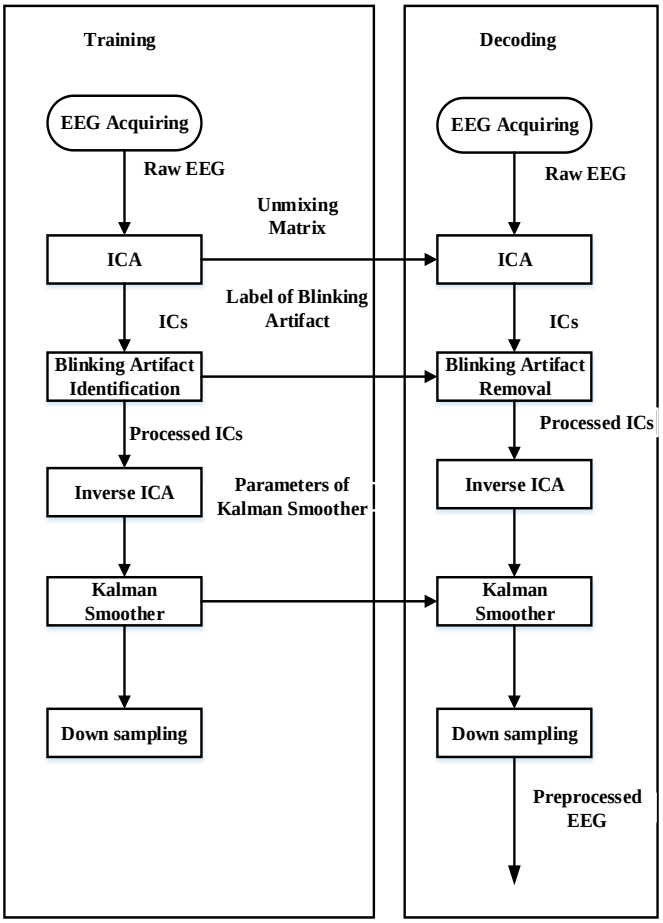

Fig. 4. Signal flowchart of the proposed preprocessing algorithm.

components (ICs) by the ICA algorithm [24], [25] to remove the blinking artifacts, which can be written as

$$s(t) = Wx(t), \tag{1}$$

where W is the unmixing matrix calculated by using Infomax algorithm [26]. $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]$, $x_n(t)$ denotes the EEG data collected by the nth electrode, n =16, and t is the sampling time point. $S(t) = [S_1(t), S_2(t), \ldots, S_n(t)]$, $S_n(t)$ denotes the nth independent components (ICs).

In the initial phase of the training procedure, three-second EEG data are used to calculate the unmixing matrix. The independent component associated with the blinking artifact is identified manually [27], and the label of the identified IC is recorded. In the decoding procedure, first, the calculated unmixing matrix is used to decompose the raw EEG signals into ICs. Then, the IC related to blinking artifact is set to be zero. By this means, the eye blinking artifact is first removed. After that, an inverse operation of the equation (1) is used to transform EEG signals back into time domain:

$$x(t) = W^{-1}s(t) \tag{2}$$

Then, the eye blinking removed EEG signal x(t) is processed by Kalman smoother to attenuate the residual noise [20]. In this process, EEG signals are written as a state-space model [28]. By giving the recursive mean estimate of the state, Kalman smoother can obtain the denoised EEG signals. All the parameters of Kalman are set previously, which means that the parameters are same for every subject. More details can be

found in our previous work [20]. After that, the signals are downsampled to 125 Hz and 512 ms EEG signals are segmented after every flashing stimulus as an epoch. The epochs of same letters are superimposed and averaged to further improve the SNR. Finally, multichannel signals have been vectorized by concatenating the multichannel signals into a vector.

2) Feature Extraction

The VAE, as a good generative model, provides the opportunity to map the time-domain features to a latent space. In the latent space, the samples can be represented as low dimensional and robust features, which may lead to better classification performance. Thus, after the EEG signals are preprocessed, the VAE is applied to provide a low dimensional and robust feature representation of the EEG signals.

VAE assumes that EEG signal x is generated from a lower dimensional latent vector z. It uses $q_\phi(z|x)$ as a recognition model to obtain the latent representation z and uses $p_\theta(x|z)$ as the generative model to reconstruct EEG signal x by utilizing z, where the φ and θ are the parameters of $q_\phi(z|x)$ and $p_\theta(x|z)$ [29].

To estimate the parameters of the model, we need to maximize the following log-likelihood function

$$\log p_\theta(X) = \sum_{i=1}^{N_S} \log p_\theta(x^i) \tag{3}$$

where N_S is the number of EEG samples.

By variational inference, the individual log-likelihood function of VAE can be rewritten as

$$\log p_\theta(x) = KL(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x)) + \Gamma(\theta, \phi; x), \tag{4}$$

where the first term is the Kullback-Leibler (KL) divergence that measures the difference between $q_\phi(z|x)$ and $p_\theta(z|x)$. The second term is the variational lower bound.

It is intractable to optimize (4), so we optimize its variational lower bound as the second best [29]. The variational lower bound in (4) can be further written as

$$\Gamma(\theta, \phi; x) = -KL(q_\phi(z \mid x) \,\|\, p_\theta(z)) + E_{q_\phi(z|x)}[\log(p_\theta(x \mid z)] \tag{5}$$

For the recognition model $q_\phi(z|x)$, we can sample the distribution of $z \sim q_\phi(z|x)$ by using a differentiable function $g_\phi(z|x)$. By applying this sampling technique, we can give an estimation of the expectation term in (5).

$$E_{z\sim q(z|x)}[\log(p_\theta(x \mid z)] \simeq \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x \mid z^l), \tag{6}$$

where L is the sampling number and

$$z^l = g_\phi(\varepsilon^l, x), \quad \varepsilon^l \sim p(\varepsilon) \tag{7}$$

We simply set L=1, the optimization objective (5) can be rewritten as

$$\Gamma(\theta, \phi; x) = -KL(q_\phi(\hat{z} \mid x) \,\|\, p_\theta(z)) + \log(p_\theta(x \mid \hat{z}) \tag{8}$$

To start the optimization process, we also need to set the distribution of differentiable $q_\phi(z|x)$, $p_\theta(x|z)$, and $p_\theta(z)$

$$\begin{aligned} p_\theta(z) &= N(z; 0, I) \\ p_\theta(x \mid z) &= N(x; \mu(z), \sigma(z) I) \\ q_\phi(z \mid x) &= N(x; \mu(x), \sigma(x) I) \end{aligned} \tag{9}$$

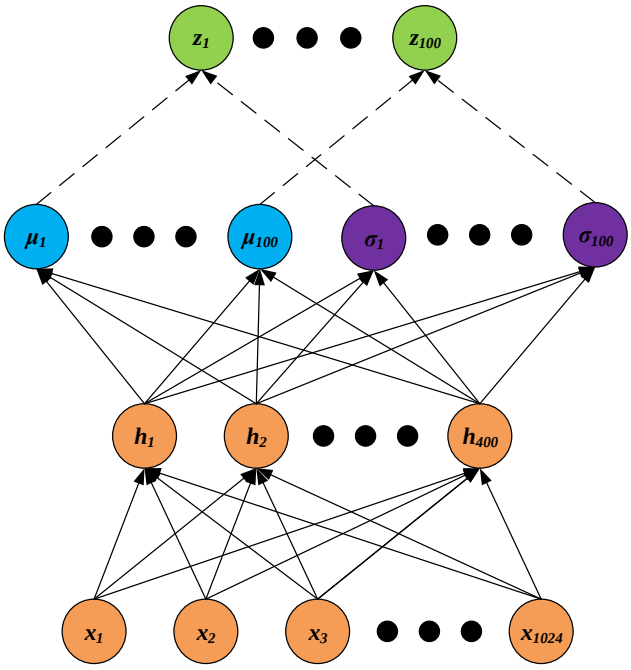

Fig. 5. Architecture of VAE network. Note that the solid lines denote connection and the dash lines denote sampling. The input layer x contains 1024 features, and the latent layer z contains 100 features.

Further, the differentiable function $g_\phi(z|x)$ is defined as

$$\begin{aligned} p(\varepsilon) &= N(\varepsilon; 0, I) \\ g_\phi(\varepsilon, x) &= \mu + \sigma \odot \varepsilon \end{aligned}, \tag{10}$$

where $\odot$ denotes the element-wise product.

By substituting (9) and (10) into (8), the first term (KL divergence) of equation (8) can be written as

$$KL(q_\phi(z \mid x) \,\|\, p(z)) = -\frac{1}{2}\sum_{d=1}^{D_l}(1 + \log \sigma_{z_d}^2 - \mu_{z_d}^2 - \sigma_{z_d}^2) \tag{11}$$

where D_l is the dimension of the latent space.

The second term (log-likelihood) can be written as

$$\log p_\theta(x \mid z) = -\frac{1}{2}\sum_{d=1}^{D_v}(\log(2\pi\sigma_{x_d}^2) + \frac{(x_d - \mu_{x_d})^2}{\sigma_{x_d}^2}) \tag{12}$$

where D_v is the dimension of the visible space.

After removing the constant term, the optimization objective can be simplified as:

$$\{\theta, \phi\} = \underset{\theta,\,\phi}{\operatorname{argmin}}(-\frac{1}{2}\sum_{d=1}^{D_l}(1 + \log \sigma_{z_d}^2 - \mu_{z_d}^2 - \sigma_{z_d}^2) + \frac{1}{2}\sum_{d=1}^{D_v}(x_d - \mu_{x_d})^2) \tag{13}$$

Once we obtain the optimization objective, we can start to determine the specific neural network structure of the VAE. Here, we choose the multi-layer perceptron (MLP) to approximate the $p_\theta(x|z)$ and $q_\phi(z|x)$ owing to its strong approximate ability for solving extremely complex problems. The MLP is a kind of neural network, which consists of several layers of nodes. Once the $p_\theta(x|z)$ and $q_\phi(z|x)$ are approximated by neural networks, they compose the architecture of the VAE. The network architecture of the VAE is shown in Fig. 5. We need to determine the parameters of the VAE network by optimizing (13). Considering (13) is a non-

convex function, we use the stochastic gradient descent (SGD) to optimize the objective.

3) Classification

Once we obtain the lower-dimension and robust features, we propose a prior information-based transductive support vector machine (PI-TSVM) to classify the EEG signals. Its optimization objective is the same as the typical transductive support vector machine (TSVM) [22], [30], but the prior information is used to solve its optimization and to output the prediction labels.

Same as typical TSVM, the optimization objective of PI-TSVM can be described as follows: given a set of labeled EEG samples $D_l = \{(z_1, y_1), (z_2, y_2), \dots, (z_l, y_l)\}$, where the corresponding class labels $y_i \in \{-1,1\}$, and a set of unlabeled EEG samples $D_u = \{z_{l+1}, z_{l+2}, \dots, z_{l+u}\}$, TSVM first performs various possible assignments to unlabeled samples, and then finds a best assignment that maximizes the margin of two-class samples based on both labeled and unlabeled samples, which could be written as

$$\begin{aligned} &\min_{\omega,b,\hat{y},\xi} \ \frac{1}{2}\|\omega\|^2 + C_l\sum_{i=1}^{l}\xi_i + C_u\sum_{i=l+1}^{l+u}\xi_i \\ &s.t.\ \ y_i(\omega^T\varphi(z_i)+b) \ge 1-\xi_i, \ \ i=1,2,...,l \\ &\quad \hat{y}_i(\omega^T\varphi(z_i)+b) \ge 1-\xi_i, \ \ i=l+1,l+2,...,l+u \\ &\quad \xi_i \ge 0, i=1,2,...,n \end{aligned} \tag{14}$$

where (ω, b) determines the hyperplane of TSVM, ξ_i is the slack variable, C_u and C_l are the penalty factors, y_i is the label of labeled samples, and $\hat{y}_i$ is the assigned label of unlabeled samples.

The prediction value p of each testing sample is calculated by

$$p_i = W^T z + b \tag{15}$$

Training a TSVM is to solve the optimization objective (14) with training data. The prediction process first utilizes (15) to calculate the prediction value and then determines the prediction label according to a certain rule. The decision rule of the typical TSVM is as follows:

$$y_i = \begin{cases} 1, & p_i > 0 \\ -1, & p_i < 0 \end{cases} \tag{16}$$

PI-TSVM utilizes prior information to improve the optimization process and decision rule. According to the experimental paradigm, the ratio of positive samples to negative samples is one to two. This prior information is used to improve the optimization and prediction process.

The optimization process is as follows:

Step 1: Solve (14) with labeled data set D_l, since there are no unlabeled data, the model degenerates into a support vector machine:

$$\begin{aligned} &\min_{\omega,b,\xi} \ \frac{1}{2}\|\omega\|^2 + C_l\sum_{i=1}^{l}\xi_i \\ &s.t.\ \ y_i(\omega^T\varphi(z_i)+b) \ge 1-\xi_i, \ \ i=1,2,...,l \\ &\quad \xi_i \ge 0, i=1,2,...,l \end{aligned} \tag{17}$$

Step 2: Calculate all prediction values $\hat{P} = \{\hat{p}_{i+1}, \hat{p}_{i+2}, \dots, \hat{p}_{i+u}\}$ of all unlabeled samples based on (15).

Step 3: Sort the prediction values in a descending order and mark the first third of the sorted prediction values as positive labels and the rest as negative labels.

Step 4: Initialize parameters C_u and C_l

Step 5: Solve (14) to obtain (ω, b) and ξ_i.

Step 6: Exchange the labels of a pair of samples according to the following rules:

$$\begin{aligned} &\exists\{i, j \mid (\hat{y}_i\hat{y}_j < 0) \wedge (\xi_i > 0) \wedge (\xi_j > 0) \wedge (\xi_i + \xi_j > 2)\} \\ &\hat{y}_i = -\hat{y}_i, \hat{y}_j = -\hat{y}_j \end{aligned} \tag{18}$$

Step7: Resolve the (14) based on the updated $\hat{y}$, repeat Step 6 to Step7 until no samples meet the criteria of (18).

Step8: Set $C_u = 2C_u$, and repeat Step 5 to Step 7 until $C_u = C_l$.

In the testing phase, prediction values of testing samples are calculated by (15). In addition, according to the experimental paradigm, every command is associated with one target sample and two non-target samples. Consequently, for the three samples associated with the same command, the sample that is associated with the maximal prediction value is determined as the target sample T, i.e.,

$$T = \arg\max_{j\in\{1,2,3\}}(p_i) \tag{19}$$

III. EXPERIMENT

A. Participants

Seven participants (aged 21-28) participated in the experiments to verify the effectiveness of the proposed DVI. All of them were right-handed and had normal vision. Two out of them had experience with BCI, whereas the remaining five did not have any BCI experience before this experiment. Every participant was confirmed to be conscious and have no brain diseases. The study adhered to the principles of the 2013 Declaration of Helsinki. All subjects signed the informed consent forms.

B. Experimental Procedure

During the experiment, the participants sat on a simulated driving platform and the user interface (a 3*3 matrix) was projected on the windshield of the driving platform. The participants were told to focus on his attention to the desired letter.

In the experimental procedure, for every command, participants were asked to count the number of the flashings of the target letter. Over the course of the experiment, every participant sent 324 commands totally.

C. Pseudo-Online Test

In this paper, we used a pseudo-online experiment to test the performance of the proposed adaptive DVI. The pseudo-online test was used to simulate the online adaption procedure. The reason for using a pseudo-online test is that it is convenient for performance analysis and comparison. The pseudo-online test process is similar to the online test, except that the data need to be collected in advance and the test dataset is an independent fixed dataset to make the comparison more reasonable. The main difference between the pseudo-online test and real online test is that for the real online test, the users can know the results

of the system (i.e., feedback of the system), which may affect the state of the users, in real-time.

For each participant, we performed a six-fold cross-validation to verify the performance of DVI. In each fold, the data were divided into two parts, one was used to simulate the online updating process, and the other was an independent test set for validating the performance of the online update DVI. In the online update process, the first 30 commands were all labeled and served as an initial training dataset. The remaining 240 commands were used as online collected unlabeled data and were progressively added to the training set in batches of 30. After updating the training set, the EEG decoding model was updated to simulate the online updating process.

To validate the proposed semi-supervised method (proposed DVI), we compared the performance between the proposed method, an adaptive DVI based on TSVM (TSVM-based DVI) [16] , and a standard supervised DVI, which used a 0.8 to 15 Hz bandpass filter for preprocessing and support vector machine for classification. In addition, to verify the effectiveness of the PI-TSVM, we implemented a PI-TSVM-based DVI and compared it with the TSVM-based DVI. All these kinds of DVIs compared in this paper and their differences in method are summarized in TABLE I.

TABLE I
SUMMARY OF FOUR KINDS OF DVI COMPARED IN THIS PAPER

Name	Feature Extraction	Classification
Proposed DVI	VAE feature	PI-TSVM
PI-TSVM DVI	Standard feature	PI-TSVM
TSVM DVI	Standard feature	TSVM
Standard DVI	Standard feature	SVM

IV. RESULTS AND DISCUSSION

A. Performance Evaluation

To determine a suitable benchmark, we calculated the system accuracy of the supervised DVI given 30, 60, and 90 commands were used for training. The results are shown in TABLE II. We

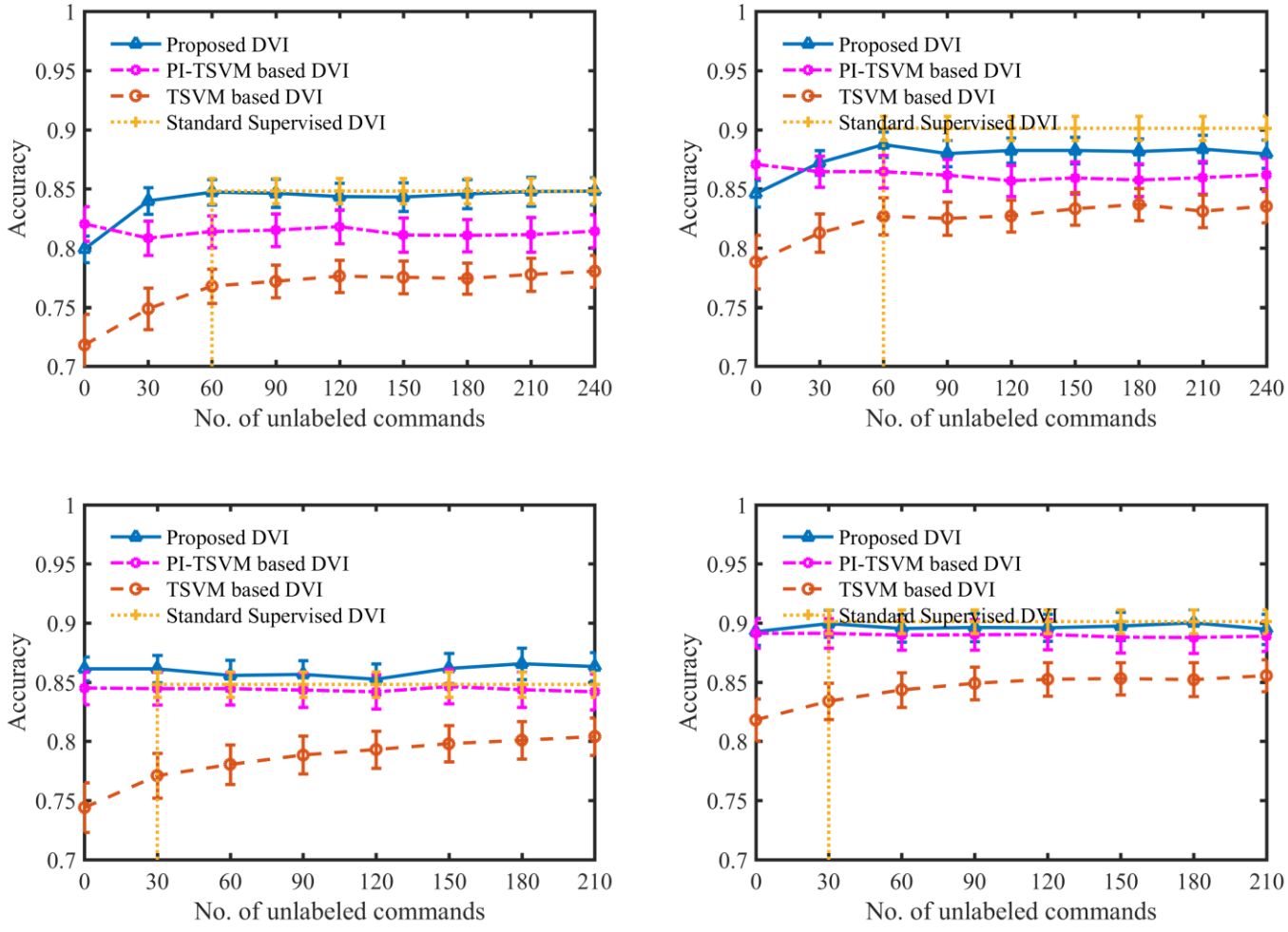

Fig. 6. System accuracy across seven participants obtained after each update, where the solid line denotes the accuracy obtained by proposed DVI, the dashed line denotes the accuracy obtained by TSVM-based DVI, the chain line denotes the accuracy obtained by PI-TSVM-based DVI, the dotted line denotes the accuracy obtained by standard supervised DVI. (a) Initial training set = 30, Number of rounds = 2. (b) Initial training set = 30, Number of rounds = 3. (c) Initial training set = 60, Number of rounds = 2. (d) Initial training set = 60, Number of rounds = 3.

can see that the system trained by 90 commands had the highest accuracy. Since the supervised DVI was used as the benchmark, we used the system trained by 90 commands in our work.

TABLE II
SYSTEM ACCURACY OF STANDARD SUPERVISED DVI GIVEN DIFFERENT AMOUNT OF TRAINING DATA

	30 commands	60 commands	90 commands
2 Rounds	0.7867	0.8176	0.8481
3 Rounds	0.8283	0.8642	0.9012

To send commands quickly, participants only used two rounds of flashes to send a command. For each participant, the first 30 commands were used as a labeled initial training set for the proposed DVI. The rest of the commands were treated as unlabeled data and were progressively added in batches of 30.

Fig. 6 shows the average accuracy curves across seven participants obtained after each update given different amount of initial training data and different rounds. The solid line is the average accuracy curve of the proposed adaptive DVI. The dashed line represents the average accuracy curve of the adaptive DVI based on TSVM [16]. The chain line denotes the average accuracy curve of the adaptive DVI based on PI-TSVM. The dotted line stands for the average accuracy curve of the standard supervised DVI. The standard supervised DVI labeled the first 90 commands and used them as the training set. That is to say, when the adaptive DVI starts to decode, the standard supervised DVI does not start working yet. Therefore, in the beginning, the accuracy of the standard supervised DVI is zero since no decoding takes place.

It can be seen that, in most situations, updating the DVIs with unlabeled data increases the accuracy. The reason for this could be related to the improved generalization performance of DVIs, since it uses new collected EEG signals to enrich the training set [31]. However, as more and more unlabeled samples are added to the training set, the abundance of the training set can hardly be further improved. Consequently, while the number of unlabeled data gets larger and larger, the improvement of accuracy becomes smaller and smaller.

Compared with the TSVM-based DVI [16], the proposed DVI makes a better use of unlabeled data. The experiment results show that the proposed DVI increases the performance faster and provides a more competitive performance.

The performance of the PI-TSVM-based DVI is between the TSVM-based DVI and the proposed DVI. The accuracy of the PI-TSVM is much higher than that of the TSVM, which shows the effectiveness of the improved TSVM (i.e., PI-TSVM). The proposed DVI increases the accuracy faster and provides a higher accuracy compared with the PI-TSVM-based DVI. The better performance of the proposed algorithm is because the VAE provides a more robust feature representation of the EEG signals.

For the proposed DVI, when the number of unlabeled commands reaches 60, the adaptive proposed DVI obtains a comparable accuracy (84.72%) to that of the standard DVI (84.81%). The initial training set (30 commands) of the adaptive DVI is much smaller than that of the standard DVI (90 commands). Therefore, the adaptive DVI can effectively reduce the training time while almost not decreasing the accuracy. This is convenient for users, since using less labeled data to train the system means that users need to spend less time on EEG data collection before they use this system.

It should be noted that, for the PI-TSVM-based DVI, it seems that updating with unlabeled samples does not improve the performance. The reason for this might be because the system performance depends on not only the classification algorithm but also on the features used. Especially for semi-supervised learning, the accuracy might even decrease when the features of unlabeled samples are not suitable for classification. This phenomenon has been reported and discussed in [32-34]. As the PI-TSVM DVI does not use the features extracted by the VAE, it is possible that its accuracy cannot increase by using the unlabeled data.

B. Effects of Initial Training Set and Round Numbers

In this section, we examined the effects of the size of the initial training set and the number of rounds used to send a command on the system performance. The size of initial training set (NI) and the number of rounds (NR) were set to be (30, 2), (60, 2), (30, 3), and (60, 3).

As shown in Fig. 6 (a) and Fig. 6 (b), we can see that when the initial training set is small, the new unlabeled data provide a remarkable promotion. However, according to Fig. 6 (c) and Fig. 6 (d), if the initial training set is large enough, the improvement of updating with unlabeled data is not significant. Although this improvement is not significant, it improves the system performance to a higher level than that of the standard supervised algorithm.

Considering that our purpose is to reduce training effort, the initial training set can be set to a small size, since the performance increases quickly. As a consequence, the initial training set could be set to be 30. When the number of rounds NR=3, the increase of performance becomes fast. This is because as the number of rounds increases, the EEG response becomes significant.

Note that the more rounds a DVI uses, the longer it takes to send a command. Compared to a three-round DVI, a two-round DVI has shorter command sending time but has lower accuracy on average. For a specific user, which DVI should be applied depends on his/her accuracies of two-round and three-round DVIs and what the DVI is used to perform. For example, if a DVI is used to the task-level control of the vehicles and the accuracy of the three-round DVI is higher for the user, the user should use the three-round DVI since the command sending time is not a key factor for the task-level control.

C. Computational Complexity

Computational complexity is a very important evaluation index, which determines whether online application can be realized. Specifically, the training time of decoding model should be shorter than the time interval between two updates. For the proposed adaptive DVI, the EEG samples are preprocessed in the decoding phase before added to the training set. Hence, the computational complexity depends on the training time of the variational autoencoder and PI-TSVM.

The computational time was tested by a computer equipped with Intel Core (TM) i5-8400 2.8 GHz CPU, 8 GB RAM, and Windows 10. As shown in Fig. 7, as more new collected

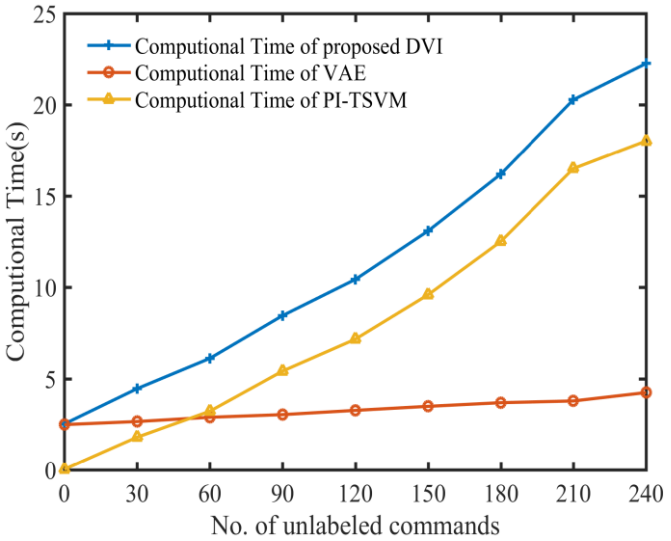

Fig. 7. The computational time of each update of Subject 7, when the initial training is 30 commands and the number of rounds is 2.

unlabeled samples are added to the training set, the computational time increases quickly. At the last update, the number of unlabeled samples is 240, increasing the computational time to 22.26 seconds. For the proposed interface, the interval between two updates is 64.8 seconds. Therefore, when the number of unlabeled samples is 240, the computational complexity meets the requirements of the online application.

However, as the unlabeled samples increase, the computational time will eventually exceed 64.8 seconds. We have to reduce the computational time to meet the requirement of the online application. Note that there is little improvement in the system accuracy when the number of unlabeled samples is around 240. We can keep the number of unlabeled samples at 240 by discarding some old unlabeled samples while collecting new ones.

Furthermore, the increase of computational complexity is mainly caused by the increase of training time of PI-TSVM. Reducing the training time of PI-TSVM can effectively reduce the computational complexity. Specifically, we can take the most accurate decoding given in the decoding process as the labels of the unlabeled samples and add these samples into the labeled training data set. In this way, we can reduce the computational time of PI-TSVM by reducing the number of unlabeled samples.

V. Conclusion

In this paper, we developed an adaptive DVI based on the proposed semi-supervised approach by combining VAE with PI-TSVM. The proposed DVI starts working with only a small initial training set and can be updated with the new collected unlabeled EEG signals. Unlike ERP spellers, DVIs need to send commands quickly to communicate with brain-controlled vehicles. This makes EEG signals feeble and challenging to classify. To solve this problem, we proposed a more powerful semi-supervised algorithm.

Compared with a typical semi-supervised algorithm, the proposed semi-supervised algorithm makes better use of unlabeled data. The experimental results show that the proposed algorithm can increase the accuracy to a competitive level by using the new collected unlabeled EEG signals, whereas the typical algorithm cannot. Besides, if the new acquired EEG signals are sufficient, the proposed algorithm can further improve the performance to a very high level.

The analysis results in different situations show that the proposed adaptive DVI has the advantages of reducing the training cost and improving performance through updating with the new collected unlabeled samples.

This work is of considerable significance to the application of brain-controlled vehicles because it significantly reduces the training effort of DVIs. However, it should be noted that the current proposed adaptive ERP-based DVI is more applicable to the task control rather than the servo control of the vehicles. We hope that the further development of the proposed DVI can also be applied to the servo control of the vehicles.

Our future work will focus on testing the proposed system in real-time by using more subjects, improving its accuracy, and integrating the proposed technique with shared control technique to develop a brain-controlled vehicle.

Acknowledgment

The authors would like to thank all the subjects for volunteering to participate in our experiments.

References

[1] T. Carlson, and J. D. R. Millan, “Brain-Controlled Wheelchairs: A Robotic Architecture,” *international conference on robotics and automation*, vol. 20, no. 1, pp. 65-73, 2013.

[2] N. Birbaumer, and U. Chaudhary, “Learning from brain control: clinical application of brain–computer interfaces,” *E-neuroforum*, vol. 6, no. 4, pp. 87-95, 2015.

[3] L. Bi, X. A. Fan, J. Ke, T. Teng, H. Ding, and Y. Liu, “Using a Head-up Display-Based Steady-State Visually Evoked Potential Brain–Computer Interface to Control a Simulated Vehicle,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 959-966, 2014.

[4] L. Bi, X. A. Fan, N. Luo, K. Jie, Y. Li, and Y. Liu, “A Head-Up Display-Based P300 Brain–Computer Interface for Destination Selection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1996-2001, 2013.

[5] D. Göhring, D. Latotzky, W. Miao, and R. Rojas, “Semi-autonomous Car Control Using Brain Computer Interfaces,” 2013.

[6] X. A. Fan, L. Bi, T. Teng, H. Ding, and Y. Liu, “A Brain–Computer Interface-Based Vehicle Destination Selection System Using P300 and SSVEP Signals,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 274-283, 2015.

[7] D. Hood, D. Joseph, A. Rakotonirainy, and S. Sridharan, “Use of brain computer interface to drive : preliminary results,” *Science & Engineering Faculty*, 2012.

[8] J. Lian, L. Bi, and W. Fei, “A Novel Event-Related Potential-Based Brain-Computer Interface for Continuously Controlling Dynamic Systems,” *IEEE Access*, pp. 1-1, 2019.

[9] R. Fazelrezai, B. Z. Allison, C. Guger, E. W. Sellers, S. C. Kleih, and A. Kubler, “P300 brain computer interface: current challenges and emerging trends,” *Frontiers in Neuroengineering*, vol. 5, pp. 14-14, 2012.

[10] L. Bi, X. A. Fan, and Y. Liu, “EEG-Based Brain-Controlled Mobile Robots: A Survey,” *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 2, pp. 161-176, 2013.

[11] K. Yu, K. Shen, S. Shao, W. C. Ng, K. Kwok, and X. P. Li, “A spatio-temporal filtering approach to denoising of single-trial ERP in rapid image triage,” *Journal of Neuroscience Methods*, vol. 204, no. 2, pp. 288-295, 2012.

[12] S. Saeedi, R. Chavarriaga, R. Leeb, and J. D. R. Millan, “Adaptive Assistance for Brain-Computer Interfaces by Online Prediction of

Command Reliability," *IEEE Computational Intelligence Magazine,* vol. 11, no. 1, pp. 32-39, 2016.

[13] G. Blanket, and E. A. Fertuck, "Journal Watch Review of Trauma, dream, and psychic change in psychoanalyses: A dialog between psychoanalysis and the neurosciences," *Journal of the American Psychoanalytic Association,* vol. 63, no. 6, pp. 1241-1244, 2015.

[14] W. Tu, and S. Sun, "Semi-supervised feature extraction for EEG classification," *Pattern Analysis and Applications,* vol. 16, no. 2, pp. 213-222, 2013.

[15] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek, "Eeg data space adaptation to reduce intersession nonstationarity in brain-computer interface," *Neural Computation,* vol. 25, no. 8, pp. 2146-2171, 2013.

[16] X. Liao, D. Yao, and C. Li, "Transductive SVM for reducing the training effort in BCI," *Journal of Neural Engineering,* vol. 4, no. 3, pp. 246-254, 2007.

[17] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system," *Pattern Recognition Letters,* vol. 29, no. 9, pp. 1285-1294, 2008.

[18] J. Wang, Z. Gu, Z. L. Yu, and Y. Li, "An online semi-supervised P300 speller based on extreme learning machine," *Neurocomputing,* vol. 269, pp. 148-151, 2017.

[19] R. C. Panicker, S. Puthusserypady, and Y. Sun, "Adaptation in P300 Brain–Computer Interfaces: A Two-Classifier Cotraining Approach," *IEEE Transactions on Biomedical Engineering,* vol. 57, no. 12, pp. 2927-2935, 2010.

[20] J. Zhang, L. Bi, J. Lian, and C. Guan, "A Single-Trial Event-Related Potential Estimation Based on Independent Component Analysis and Kalman Smoother," in IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 2018, pp. 7-12.

[21] D. P. Kingma, and M. Welling, "Auto-Encoding Variational Bayes," *international conference on learning representations,* 2014.

[22] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines." pp. 200-209.

[23] 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.

[24] A. Delorme, T. J. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis," *NeuroImage,* vol. 34, no. 4, pp. 1443-1449, 2007.

[25] S. Halder, M. Bensch, J. Mellinger, M. Bogdan, A. Kubler, N. Birbaumer, and W. Rosenstiel, "Online artifact removal for brain-computer interfaces using support vector machines and blind source separation," *Computational Intelligence and Neuroscience,* vol. 2007, pp. 8, 2007.

[26] A. J. Bell, and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation,* vol. 7, no. 6, pp. 1129-1159, 1995.

[27] C. A. Joyce, I. Gorodnitsky, and M. Kutas, "Automatic removal of eye movement and blink artifacts from EEG data using blind component separation," *Psychophysiology,* vol. 41, no. 2, pp. 313-325, 2004.

[28] S. Georgiadis, P. O. Rantaaho, M. P. Tarvainen, and P. A. Karjalainen, "Single-trial dynamical estimation of event-related potentials: a Kalman filter-based approach," *IEEE Transactions on Biomedical Engineering,* vol. 52, no. 8, pp. 1397-1406, 2005.

[29] C. Doersch, "Tutorial on Variational Autoencoders," *arXiv: Machine Learning,* 2016.

[30] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms,* 2002.

[31] S. Ding, Z. Zhu, and X. Zhang, "An overview on semi-supervised support vector machine," *Neural Computing and Applications,* vol. 28, no. 5, pp. 969-978, 2017.

[32] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: theory, algorithms, and their application to human-computer interaction," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, no. 12, pp. 1553-1567, 2004.

[33] O. Chapelle, V. Sindhwani, and S. S. Keerthi, "Optimization Techniques for Semi-Supervised Support Vector Machines," *Journal of Machine Learning Research,* vol. 9, pp. 203-233, 2008.

[34] Y. Li, and Z. Zhou, "Towards Making Unlabeled Data Never Hurt," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 37, no. 1, pp. 175-188, 2015.