




A hierarchical semi-supervised extreme learning machine method for EEG recognition

Qingshan She¹ · Bo Hu¹ · Zhizeng Luo¹ · Thanh Nguyen² · Yingchun Zhang^{1,2} 

Received: 12 June 2017 / Accepted: 22 December 2017 / Published online: 28 July 2018
© International Federation for Medical and Biological Engineering 2018

Abstract

Feature extraction and classification is a vital part in motor imagery-based brain-computer interface (BCI) system. Traditional deep learning (DL) methods usually perform better with more labeled training samples. Unfortunately, the labeled samples are usually scarce for electroencephalography (EEG) data, while unlabeled samples are available in large quantity and easy to collect. In addition, traditional DL algorithms are notoriously time-consuming for the training process. To address these issues, a novel method of hierarchical semi-supervised extreme learning machine (HSS-ELM) is proposed in this paper and applied for motor imagery (MI) task classification. Firstly, the deep architecture of hierarchical ELM (H-ELM) approach is employed for feature learning automatically, and then these new high-level features are classified using the semi-supervised ELM (SS-ELM) algorithm which can exploit the information from both labeled and unlabeled data. Extensive experiments were conducted on some benchmark datasets and EEG datasets to evaluate the effectiveness of the proposed method. Compared with several state-of-the-art methods, including SVM, ELM, SAE, H-ELM, and SS-ELM, our HSS-ELM method can achieve better classification accuracy, a mean kappa value of 0.7945 and 0.5701 across all subjects in the training and evaluation sessions of BCI Competition IV Dataset 2a, respectively. Finally, it comes to the conclusion that the proposed method has achieved superior performance for feature extraction and classification of EEG signals.

Keywords Motor imagery electroencephalography · Extreme learning machines · Semi-supervised learning · Hierarchical · Deep learning

1 Introduction

Brain-computer interface (BCI) is an emerging technology to communicate with external devices using exclusively brain activities [1, 2], which has been widely applied to auxiliary control, medical rehabilitation, astronaut training, smart home, entertainment games, and other fields [3]. In most BCI systems, electroencephalography (EEG) signals are used owing to its noninvasive nature and affordable recording

equipment which facilitate real-time operation [4]. Motor imagery (MI) is one of numerous paradigms which rely on various known modulations of the EEG signal related to an event, specifically the imagination of a motor action without any actual movement of limbs [5]. During MI, an event-related desynchronization (ERD) occurs, resulting in a decrease in sensorimotor rhythms, most notably in the mu band (8–12 Hz), and the ERD is then followed by an event-related synchronization (ERS) which occurs directly after the termination of MI [6]. In recent years, MI EEG pattern recognition research has attracted increasing attention in BCI. However, MI EEG is notoriously difficult to analyze due to the low signal-to-noise ratio, significant subject-to-subject variations, and the requirement for long time training. Therefore, it is of great interest to be able to extract and recognize features of different MI tasks quickly and efficiently [7].

MI-based BCI systems decode brain signals via pre-processing, feature extraction, classification, and generation of control commands. There are two principle challenges. The

✉ Qingshan She
qsshe@hdu.edu.cn

✉ Yingchun Zhang
yingchun.umn@gmail.com

¹ Institute of Intelligent Control and Robotics, Hangzhou Dianzi University, Zhejiang, Hangzhou 310018, China

² Department of Biomedical Engineering, University of Houston, Houston, TX 77204, USA

first is the high variability of the recorded EEG data across both trials and individuals, and thus adaptive and discriminative features are needed. To date, there are many feature extraction algorithms, including common spatial pattern (CSP) algorithm, filter bank CSP (FBCSP) [8], augmented complex CSP (ACCSP) [9], and combined features [10]. Recently, Djemal [10] reported exceptional classification results on three-class EEG datasets by combining the features of the phase and amplitude of the brain signals using fast Fourier transform (FFT) and autoregressive (AR) modeling of the reconstructed phase space as well as the modification of the BCI parameters. The second hurdle is the pattern recognition domain. Various classification algorithms have been proposed to discriminate different MI tasks, including k-nearest neighbor (KNN) [11], support vector machine (SVM) [10], neural networks [12], and naive Bayes [13]. Recently, a new method called extreme learning machine (ELM) has been proposed by Huang et al. [14–17]. It is a simple and efficient learning algorithm for training single layer feed-forward neural networks (SLFNs), which has faster learning speed and better generalization in comparison with the well-known back propagation (BP) neural networks and SVM. As a result, ELM has been applied to pattern recognition tasks in the BCI systems and has shown its superiority over traditional classification methods [18–21].

However, there are still some shortcomings about ELM's applications in BCI: (1) In real clinical applications, labeled EEG samples are scarce and expensive to obtain, while unlabeled EEG samples are abundant yet not fully utilized [18]; (2) Due to its shallow structure (i.e., single-layer), feature learning may not be effective for EEG signals [22]. To overcome the abovementioned problems, ELM has been extended to both semi-supervised learning and deep learning (DL) domains [22–27]. On the one hand, semi-supervised learning is a technique that can utilize both labeled and unlabeled data to achieve higher prediction accuracy than supervised or unsupervised learning. It can help solve many practical problems, such as text classification, spam filtering, and natural language processing, where obtaining training labels is nontrivial while unlabeled data is available in large quantity and easy to collect [28]. In 2014, Huang et al. have proposed the semi-supervised ELM (SS-ELM) algorithm using manifold regularization technique [26]. SS-ELM inherits the advantages of both classical ELM and semi-supervised learning methods, yielding better cluster results on the University of California Irvine (UCI) datasets than Laplacian support vector machine (LapSVM) [29]. On the other hand, inspired by the great success of DL, multilayer ELM (ML-ELM) has been proposed by virtue of the ELM auto-encoder (ELM-AE) approach which represents features based on singular values [23]. Resembling deep networks, ML-ELM stacks on top of ELM-AE to create a multilayer neural network. This method learns significantly faster than existing deep networks, outperforming deep belief

network (DBN) [30], stacked auto-encoder (SAE) [31], stacked denoising auto-encoder (SDAE) [32], and deep Boltzmann machine (DBM) [33] on the MNIST datasets. In [17], ML-ELM is used to identify binary-class MI tasks, and improve classification performance in contrast to other methods, such as KNN and Bayes. Similarly, a deep representation learning via ELM method (DrELM) has been proposed by designing a new stacked architecture to learn deep representations, and it achieved good prediction results on both synthetic and real-world data sets [27]. As an improved version from ML-ELM, a hierarchical extreme learning machine (H-ELM) method has been proposed recently. H-ELM uses l_1 -norm instead of l_2 -norm to obtain more compact and sparse hidden information, and thus achieves better and faster performance than SAE, SDAE, DBN, ML-ELM, and DBM algorithms [22].

Inspired by the DL and semi-supervised learning theory, we propose a novel hierarchical semi-supervised ELM (HSS-ELM) method for the application of multiclass MI EEG classification. The deep architecture of the H-ELM algorithm is first employed to perform feature learning of both labeled and unlabeled data, and then the resulted high-level features are applied to obtain classification results by the SS-ELM approach. Compared to existing ELM-based algorithms, the new method has several noteworthy aspects:

- (1) Considering that feature learning is inadequate in the SS-ELM algorithm with single hidden layer, it is extended to the deep networks in order to extract more hidden information automatically in EEG data.
- (2) Since the H-ELM approach uses exclusively labeled samples, it is modified with semi-supervised learning to exploit both labeled and unlabeled data to find more useful information, contributing to improved classification accuracy and better generalization capability of the deep feed-forward neural networks.
- (3) Inheriting the advantages of basic ELM algorithm, the proposed method can improve the classification accuracy and have faster training speed compared with traditional DL algorithm SAE.

2 Methods

2.1 Conventional ELM

Extreme learning machine (ELM) has been proposed for training SLFNs and demonstrated to have excellent learning accuracy and speed in various applications [14]. Suppose we have a training set with N samples, $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where

each sample is denoted by $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{ip}]^T \in \mathbb{R}^p$, and its corresponding network target vector is $\mathbf{y}_i = [y_{i1}, y_{i2}, \dots, y_{iq}]^T \in \mathbb{R}^q$, where p and q represent their corresponding dimensions, and T denotes a transpose operation. **ELM aims to learn a decision rule from the training data in classification tasks.**

Assuming that m is the number of hidden neurons, the output function of ELM for SLFNs is given by

$$\begin{aligned} y_i &= \sum_{j=1}^m \beta_j g(\mathbf{a}_j^T \mathbf{x}_i + b_j), i = 1, 2, \dots, N, \quad j \\ &= 1, 2, \dots, m \end{aligned} \quad (1)$$

where $\mathbf{a}_j = [a_{j1}, a_{j2}, \dots, a_{jp}]^T$ represents the weights between the j -th hidden node and input layer, and the corresponding weight vector is denoted by $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m]^T$, b_j is the bias of the j -th hidden node, and the corresponding bias vector is represented by $\mathbf{b} = [b_1, b_2, \dots, b_m]^T$, and $g(\bullet)$ is the activation function which can be any nonlinear piecewise continuous functions, such as the Sigmoid function and Gaussian function. Note that both \mathbf{a} and \mathbf{b} can be randomly generated according to any continuous probability distribution.

For convenience, formulation (1) can be written in matrix form as:

$$\mathbf{Y} = \mathbf{H}\boldsymbol{\beta} \quad (2)$$

where $\mathbf{Y} = [y_1, y_2, \dots, y_N]_{N \times q}^T$ represents the target output of the network, $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_m]_{m \times q}^T$ denotes the output weights between the hidden layer and output layer. \mathbf{H} is the hidden layer output matrix given by

$$\begin{aligned} \mathbf{H} &= h_\theta(\mathbf{X}) \\ &= \begin{bmatrix} g(\mathbf{a}_1^T \mathbf{x}_1 + b_1) & \dots & g(\mathbf{a}_m^T \mathbf{x}_1 + b_m) \\ \dots & \dots & \dots \\ g(\mathbf{a}_1^T \mathbf{x}_N + b_1) & \dots & g(\mathbf{a}_m^T \mathbf{x}_N + b_m) \end{bmatrix}_{N \times m} \end{aligned} \quad (3)$$

where $h_\theta(\bullet)$ is actually a feature mapping function which maps the data from the N -dimensional input space to the m -dimensional hidden layer feature space, and $\theta = \{\mathbf{a}, \mathbf{b}\}$ are the parameters of the mapping function.

The ordinary ELM aims to minimize the objective function, $\arg\min_{\boldsymbol{\beta}} (\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|_2^2)$. In order to improve its stability and generalization performance, a small positive value can be added [15], and thus, the objective function can be rewritten as:

$$\arg\min_{\boldsymbol{\beta}} (\|\boldsymbol{\beta}\|_2^2 + C\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|_2^2) \quad (4)$$

where C is a penalty coefficient on the training errors, and $\|\bullet\|_2$ denotes the l_2 -norm of a matrix or a vector. We can obtain the output weight vector $\boldsymbol{\beta}$ using the Moore-

Penrose principle. If the number of training data N is more than m , the solution of Eq.(4) is:

$$\boldsymbol{\beta} = \left(\frac{1}{C} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{Y} \quad (5)$$

If N is less than m , the solution of Eq. (4) is:

$$\boldsymbol{\beta} = \mathbf{H}^T \left(\frac{1}{C} + \mathbf{H} \mathbf{H}^T \right)^{-1} \mathbf{Y} \quad (6)$$

2.2 SS-ELM

Given a training set $\{\mathbf{X}, \mathbf{Y}\} = \{\{\mathbf{X}_l, \mathbf{Y}_l\}, \mathbf{X}_u\}$, and let $\{\mathbf{X}_l, \mathbf{Y}_l\} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_l, \mathbf{y}_l)\}$ be the first l labeled samples and $\mathbf{X}_u = \{\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+u}\}$ be the last u unlabeled samples in \mathbf{X} , the objective function of SS-ELM is defined as:

$$\arg\min_{\boldsymbol{\beta}} (\|\boldsymbol{\beta}\|_2^2 + C\|\mathbf{G}\boldsymbol{\beta} - \mathbf{Y}\|_2^2 + \lambda \boldsymbol{\beta}^T \mathbf{H}^T \mathbf{L} \mathbf{H} \boldsymbol{\beta}) \quad (7)$$

where $\|\mathbf{G}\boldsymbol{\beta} - \mathbf{Y}\|_2^2$ contributes the empirical risk minimization, $\|\boldsymbol{\beta}\|_2^2$ denotes the regularization term, $\boldsymbol{\beta}^T \mathbf{H}^T \mathbf{L} \mathbf{H} \boldsymbol{\beta}$ is the manifold regularization framework, and λ is a trade-off parameter, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is known as the graph Laplacian, \mathbf{D} is a diagonal matrix with its diagonal elements $\mathbf{D}_{ii} = \sum_{j=1}^{l+u} w_{i,j}$, and $\mathbf{H}\boldsymbol{\beta}$ is the output matrix of the network. \mathbf{W} denotes the similarity matrix. Calculate the hidden layer output matrix \mathbf{H} for \mathbf{X} with $\mathbf{H} = h_\theta(\mathbf{X})$ and the hidden layer output matrix \mathbf{G} for \mathbf{X}_l with $\mathbf{G} = h_\theta(\mathbf{X}_l)$.

According to [26], when the number of data is larger than or equal to the number of hidden neurons ($l+u > m$), we obtain the solution to the SS-ELM:

$$\boldsymbol{\beta} = (\mathbf{I}_m + C\mathbf{G}^T \mathbf{G} + \lambda \mathbf{H}^T \mathbf{L} \mathbf{H})^{-1} C\mathbf{G}^T \mathbf{Y} \quad (8)$$

where \mathbf{I}_m is an identity matrix of dimension m . When the number of labeled data is fewer than the number of hidden neurons ($l+u < m$), which is common in semi-supervised learning, an alternative solution is given in the following:

$$\boldsymbol{\beta} = \mathbf{G}^T (\mathbf{I}_{l+u} + C\mathbf{G} \mathbf{G}^T + \lambda \mathbf{L} \mathbf{H} \mathbf{H}^T)^{-1} C\mathbf{Y} \quad (9)$$

where \mathbf{I}_{l+u} is an identity matrix of dimension $l+u$.

2.3 H-ELM

Recently, deep learning and representation have attracted many research interests with its remarkable success in many applications [34]. Thus, the original ELM algorithm is modified by adopting multilayer or deep learning architectures.

Due to the fact that deep architectures can potentially capture relevant higher level abstractions and characterize the data representations more accurately, H-ELM [22] have been proposed to extend ELM methods to incorporate deep learning framework showing better feature learning ability compared to SLFNs.

Utilizing unsupervised feature extraction and supervised feature classification, the H-ELM method demonstrated much faster training speed and better learning efficiency compared to other DL methods, such as SAE, SDA, DAN, DAM, and ML-ELM. The framework of H-ELM is shown in Fig. 1. H-ELM's training architecture consists of unsupervised hierarchical feature representation and supervised classification [22]. First, the input raw data is transformed into an ELM random feature space, which can help to exploit hidden information among training samples, and then a multi-layer unsupervised learning is performed to obtain the high-level sparse features using the sparse ELM-AE approach [22]. Secondly, original ELM is applied to identify the obtained high-level features.

2.4 HSS-ELM

Combining the advantages of both H-ELM and SS-ELM, we proposed a new semi-supervised extreme learning machine method with deep architecture, namely, HSS-ELM. The detailed schematic of the HSS-ELM algorithm is shown in Fig. 2. Firstly, for the input data, high-level features are extracted from both label and unlabeled data by the deep architecture in H-ELM. Then, these new features are used for the classification step by applying the SS-ELM algorithm.

In semi-supervised setting, we have few labeled data and relatively more unlabeled data. Given a training set $\{\mathbf{X}, \mathbf{Y}\} = \{\{\mathbf{X}_l, \mathbf{Y}_l\}, \mathbf{X}_u\}$, where \mathbf{X}_l represents the first l labeled samples in \mathbf{X} , and \mathbf{X}_u represents the last u unlabeled samples, the details of our method are described as follows:

Step (1): Utilize the H-ELM algorithm to extract features from \mathbf{X} and \mathbf{X}_l , obtaining the corresponding high-

level representations \mathbf{H}_K and \mathbf{G}_K , respectively, where K denotes the number of hidden layers.

Step (2): Discriminate the high-level features \mathbf{G}_K and \mathbf{H}_K using the SS-ELM classification method.

- 1) The graph Laplacian \mathbf{L} is calculated to construct the manifold regularization framework.
- 2) The high-level features \mathbf{H}_K and \mathbf{G}_K are used to modify Eq. (7) in SS-ELM, the new formulation of HSS-ELM is written as:

$$\underset{\beta}{\operatorname{argmin}} \left(\|\beta\|_2^2 + C\|\mathbf{G}_K\beta - \mathbf{Y}\|_2^2 + \lambda\beta^T\mathbf{H}_K^T\mathbf{L}\mathbf{H}_K\beta \right) \quad (10)$$

and the output weights β can be obtained by Moore-Penrose principle. When the number of input data is larger than or equal to the number of hidden layer's nodes ($l+u \geq m$), the solution of Eq. (10) is given:

$$\beta = (\mathbf{I}_m + C\mathbf{G}_K^T\mathbf{G}_K + \lambda\mathbf{H}_K^T\mathbf{L}\mathbf{H}_K)^{-1}C\mathbf{G}_K^T\mathbf{Y} \quad (11)$$

and when $l+u < m$, the solution of Eq. (10) is:

$$\beta = \mathbf{G}_K^T(\mathbf{I}_{l+u} + C\mathbf{G}_K\mathbf{G}_K^T + \lambda\mathbf{L}\mathbf{H}_K\mathbf{H}_K^T)^{-1}C\mathbf{Y} \quad (12)$$

Step (3): For test data $\{\mathbf{X}_{test}, \mathbf{Y}_{test}\} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, the high-level representation \mathbf{T}_K is calculated through the deep architecture in H-ELM, and then the labels $\mathbf{Y}_{predict}$ can be predicted in the following:

$$\mathbf{Y}_{predict} = \mathbf{T}_K\beta \quad (13)$$

While partially derived from H-ELM and SS-ELM, it is important to highlight the key differences of the proposed HSS-ELM method compared to its predecessors. Firstly,

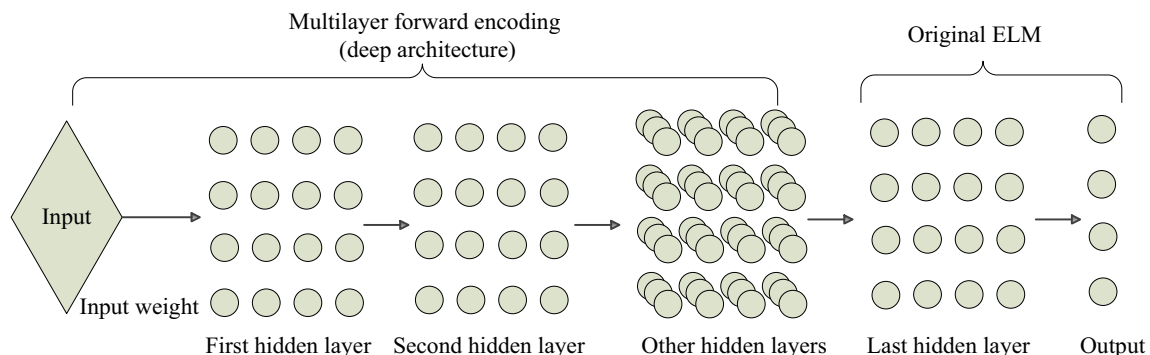
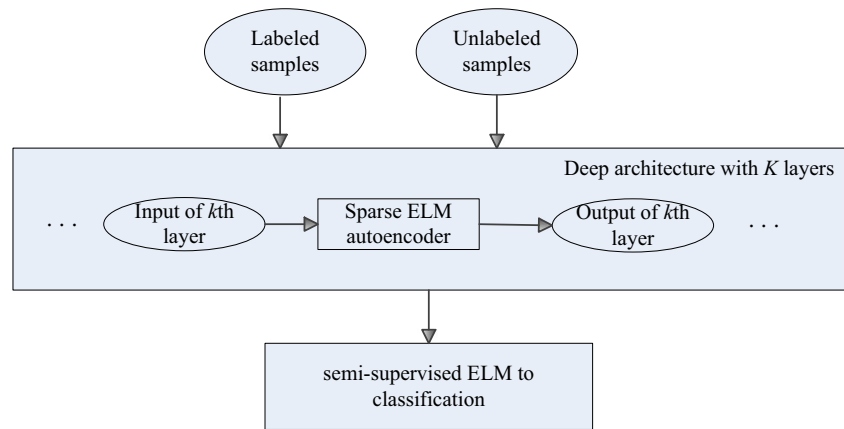


Fig. 1 The overall framework of the H-ELM learning algorithm

Fig. 2 The schematic of the proposed HSS-ELM algorithm



following the feature extraction routine, H-ELM performs classification using the original ELM method, while HSS-ELM implements SS-ELM for classification, allowing both unlabeled data and labeled data to be utilized. Secondly, the SS-ELM routine in HSS-ELM is modified to incorporate deep network architecture by formulating a new objective function, updating Eq. (10) from Eq. (7). More specifically, \mathbf{H}_K and \mathbf{G}_K are high-level representations of \mathbf{X} and \mathbf{X}_l , respectively. These modified representations can effectively capture the intrinsic features of the input data, thus improving the classification accuracy.

3 Experimental results

In this section, several experiments on benchmark datasets and EEG datasets were performed to show the effectiveness of the proposed HSS-ELM method. The new algorithm was constructed based on both the H-ELM and SS-ELM codes (http://www.ntu.edu.sg/home/egbhuang/elm_codes.html), and compared with the original ELM and other state-of-the-art approaches. All the methods were implemented in MATLAB 2014a environment on a PC with a 3.4GHz processor and 8.0 GB RAM.

3.1 Experiment on benchmark datasets

3.1.1 Description of benchmark datasets

In order to evaluate its performance, the proposed HSS-ELM method was first applied to four popular benchmark datasets: Waveform [35], USPST and COIL20 [26], as well as Yale [36]. These datasets are widely used for evaluating the performance of machine learning algorithms. The details of these multiclass datasets are shown in Table 1. The Waveform dataset consists of 5000 samples from three classes of noisy waveforms, and each sample contains 21 attributes. The USPST dataset, a popular subset of the well-known handwritten digit recognition dataset USPS, contains 2007 digit images

of the size 16×16 pixels with 256 Gy levels. The Columbia object image library (COIL20) is a multiclass image classification dataset consists of 1440 Gy-scale image sample of 20 different objects, in which each sample is a 32×32 gray scale image of one object taken from a specific view. The Yale dataset consists of 165 frontal face images from 15 subjects under varying illumination conditions, with image size of 100×100 pixel.

3.1.2 Experimental setup

The experimental setup is described in further details in [26]. In brief, the analysis for each of the four datasets followed a three-step procedure: (i) the data pool was randomly divided into four subsets with equal proportion: labeled training set, unlabeled training set, validation set, and test set; (ii) each attribute in the image datasets USPST, COIL20, and Yale was normalized to $[0, 1]$ by dividing 255. For Waveform, the training set is first normalized, and then other data is normalized according to the mean and standard deviation obtained from the training set using the z-score approach; (iii) the classification process was repeated ten times, and the averaged accuracy and training time were recorded for further analysis.

In the experiment, three hyperparameters C , λ , and m were defined, and the best model parameters were determined from $C \in \{10^{-9}, 10^{-8}, \dots, 10^8, 10^9\}$, $\lambda \in \{10^{-9}, 10^{-8}, \dots, 10^8, 10^9\}$, and $m \in \{100, 200, \dots, 1400, 1500\}$. To ensure a valid comparison between the different methods, the same range sets of the user-specified parameters were used in cross-validation. Since SS-ELM, H-ELM, and HSS-ELM each has different

Table 1 Description of the benchmark datasets

Datasets	Dimension	Numbers	Number of categories
Waveform	21	5000	3
USPST	256	2007	10
COIL20	1024	1440	20
Yale	10,000	165	15

network structure, the details of hidden nodes in each network are dependent on the selected m as follow: $m \times m \times 5000$ for the three hidden layer in H-ELM, m for the single hidden layer in SS-ELM, and $m \times m \times 5000$ in HSS-ELM. Here, the number of the last hidden layer node was chosen empirically from our previous experiments to be 5000, since this last hidden layer is usually much larger than m to generate sparse performance [22]. By cross-validation, the best parameters for each dataset are summarized as follows: $C = 10^4$, $\lambda = 10^2$, and $m = 20$ on Waveform, $C = 10^5$, $\lambda = 10^6$, $m = 200$ on USPST, $C = 10^4$, $\lambda = 10^2$, and $m = 400$ on COIL20, and $C = 10^1$, $\lambda = 10^2$, and $m = 300$ on Yale. In addition, the number of iterations and the learning rate were set to be 500 and 0.1 respectively, for the SAE approach.

3.1.3 Comparisons with related algorithms

In this experiment, we compare the proposed HSS-ELM to four baseline algorithms, including ELM, SAE, H-ELM, and SS-ELM. The classification performance is evaluated in terms of average accuracy and standard deviation ($\text{acc} \pm \text{std}$). Table 2 summarizes the classification results.

From the results shown in Table 2, it is evidenced that the proposed HSS-ELM achieved comparable, if not better, performance with the pure supervised learning algorithms, such as single-layer ELM, SAE, and H-ELM with deep architecture, demonstrating the utilization of unlabeled data by HSS-ELM not only did not hamper but actually enhanced classification accuracy, suggesting effective exploitation of unlabeled data. It is also observed that HSS-ELM outperformed the other existing semi-supervised algorithm SS-ELM on all four datasets, indicating the successful incorporation of deep network architecture to extract compact, high-level features from the inputs, thus leading to improved overall learning performance compared to single-layer SS-ELM. Most notably, HSS-ELM yielded higher mean accuracy than its predecessors H-ELM and SS-ELM on almost all of the datasets (except COIL20). For the COIL20 dataset, HSS-ELM resulted in an average classification accuracy of 97.28–2.50% higher than the SS-ELM method, and slightly lower than H-ELM (97.83). The average classification accuracy on USPST by HSS-ELM

was 90.42%, yielding a 0.54% improvement over H-ELM and a 4.68% improvement than SS-ELM. For the Yale dataset, the proposed HSS-ELM yielded average accuracy of 76.09%, 1.95% higher than H-ELM, and 2.92% higher than SS-ELM. The average classification accuracy of Waveform using HSS-ELM was 85.08%, better than H-ELM (85.00%) and SS-ELM (84.22%).

Table 2 also presents the training time of different algorithms to evaluate the computational efficiency. Among the five classification algorithms, ELM is the most efficient one, while H-ELM, SS-ELM, and HSS-ELM are relatively comparable, and SAE is the least time efficient. Due to its complexity architecture, the training time of the proposed method is slightly more than ELM, SS-ELM, and H-ELM. However, compared with SAE, HSS-ELM can greatly accelerate the training speed, yielding an increase in time efficiency of 5.8 times, 65.6 times, 134 times, and 85 times for Waveform, USPST, COIL20, and Yale datasets, respectively. Furthermore, HSS-ELM method also exhibited more evident advantage on classification problems of higher number of classes, which is consistent with the empirical results in the ELM literature [15]. These results show that the developed HSS-ELM method can achieve excellent trade-off between classification accuracy and computational cost.

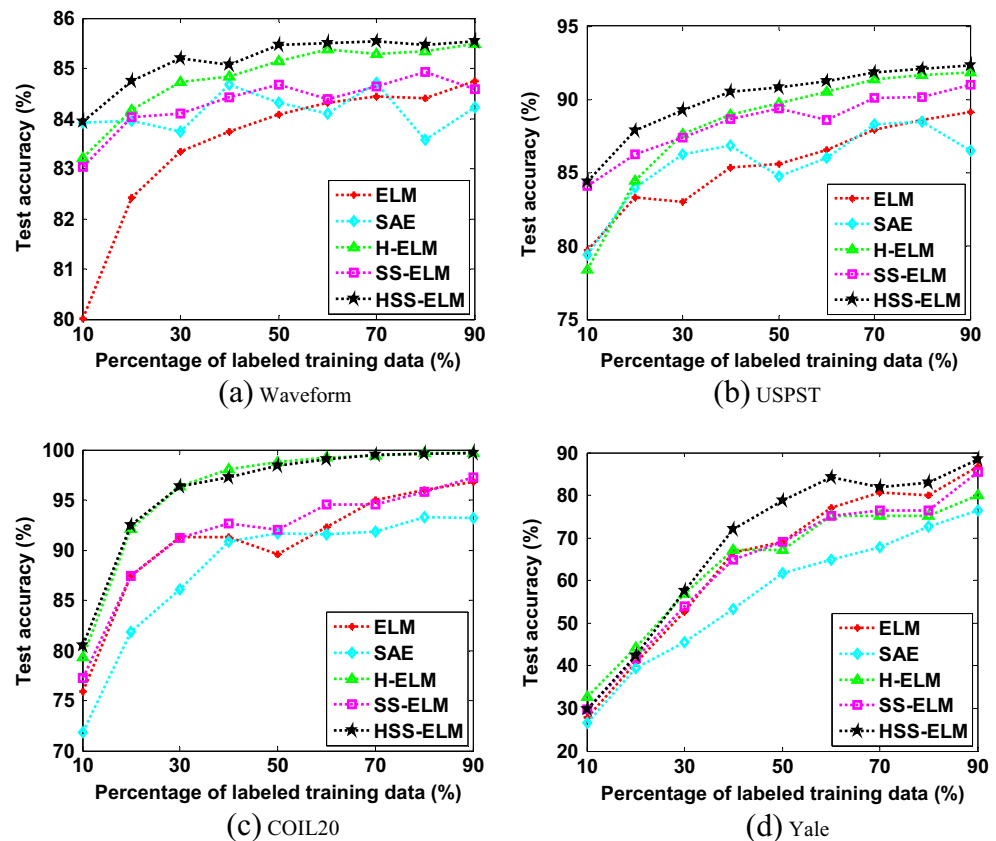
3.1.4 Performance with different ratios of labeled training data

In accordance with the experimental setup described in [28], we seek to evaluate the performance of different algorithms under different proportion of labeled to unlabeled data for training. Particularly, in this experiment, 20% of the total samples were first randomly selected as test set, and another 20% as validation set. Then training set was selected from the remaining samples, and the ratio of labeled-to-unlabeled data was set to vary systematically from 10:90 to 90:10%. The process of allocating samples is repeated ten times. Finally, the performance of different algorithms is evaluated in terms of the average classification accuracy on test set. The results for all the benchmark datasets, as shown in Fig. 3, indicated that low accuracy was found for all methods when there is a

Table 2 Comparisons of classification results and training time on each dataset using different methods

Method	Waveform		USPST		COIL20		Yale	
	acc \pm std	Time (s)	acc \pm std	Time (s)	acc \pm std	Time (s)	acc \pm std	Time (s)
ELM	83.62 \pm 1.17	0.0468	84.39 \pm 1.89	0.0156	90.50 \pm 1.51	0.0624	71.70 \pm 9.85	0.0624
SAE	84.64 \pm 2.12	75.38	89.78 \pm 3.02	160.71	97.22 \pm 2.11	241.25	74.39 \pm 2.04	395.93
H-ELM	85.00 \pm 0.97	0.1248	89.88 \pm 2.19	0.0624	97.83 \pm 1.68	0.2340	74.14 \pm 2.78	2.4180
SS-ELM	84.22 \pm 1.17	12.8233	85.74 \pm 1.86	2.2620	94.78 \pm 1.66	1.3260	73.17 \pm 8.27	0.1404
HSS-ELM	85.08 \pm 0.68	13.0729	90.42 \pm 1.79	2.4492	97.28 \pm 0.77	1.7940	76.09 \pm 7.40	4.6488

Fig. 3 Testing accuracy with respect to different proportion of labeled data using five methods on each dataset



small proportion of labeled training data. Overall, increases in the amount of labeled training data resulted in higher classification accuracy. While this trend was true for all studied methods, it is worth to note that, the proposed method yielded highest accuracy at every proportion of available labeled data for training (with some exceptions in the COIL20 dataset). Furthermore, in the case of less labeled data, HSS-ELM has classification advantage over H-ELM and other supervised methods, and also outperformed the semi-supervised algorithm SS-ELM. Consider the case of Waveform dataset, at labeled data ratio of 0.1, the average accuracy of HSS-ELM is 83.94%, outperforming H-ELM (83.21%) and SS-ELM (83.03%). When the ratio of labeled data is 0.9, the average accuracy of HSS-ELM is 85.58%, higher than both H-ELM (85.48%) and SS-ELM (84.58%). We observed similar trend on the COIL20, USPST, and Yale datasets.

3.2 Experiment on BCI datasets

3.2.1 BCI datasets

This section evaluates the performance of the proposed method on MI EEG datasets. BCI Competition IV Dataset 2a [37] comprises of MI EEG measurements from nine subjects with four classes of movement, namely, left hand, right hand, feet, and tongue. Two sessions were recorded from each subject, in

which one for training and the other for evaluation. Each session consisted of 288 trials of data recorded with 22 EEG channels and three monopolar electrooculogram (EOG) channels (with left mastoid serving as reference) [8].

3.2.2 Protocol

For each trial in the BCI Competition IV Dataset 2a, EEG data from 0 to 2 s after the initiation of MI were segmented and processed with 8–30 Hz band-pass filter. Preliminary features was extracted from the filtered data using the common spatial pattern (CSP) approach, yielding feature dimension of 24, as performed in [38]. Finally, the proposed HSS-ELM method was used to learn and classify the compact feature representations which can help to remove redundancy of the input feature data. The setting of parameters C and λ were same as those of section 3.1.2, and the setting of the hidden nodes m was $\{10, 20, \dots, 100\}$, and then all the optimal values were selected by tenfold cross-validation.

Note that there are three experimental setups for EEG classification in [39]: (1) a shallow method that uses prior knowledge; (2) a deep architecture that also uses prior knowledge; (3) a deep architecture that does not use any prior knowledge. Experimental results have demonstrated that the second method can perform better in extracting suitable EEG features for classification tasks [39]. The similar results have been verified

in our previous work [38]. According to the two-stage extraction strategy in [38], the features are first extracted from raw EEG by the common CSP algorithm, and then their higher level representations are further obtained by different deep learning methods to classification in our experiment.

3.2.3 Cross-validation results

The classification performances of the six approaches (SVM, ELM, SAE, H-ELM, SS-ELM, and our proposed HSS-ELM) were first investigated on the training data. The performance was evaluated in terms of the mean kappa value using ninefold cross-validations [10]. The samples were split into the training and test subsets: eight subsets for training and one for testing. Specially, in the semi-supervised methods SS-ELM and HSS-ELM, the testing subset ignoring the labels was added into the training data for learning a semi-supervised classifier. The results by averaging the ninefold accuracies on the testing subset are shown in Table 3. It showed that HSS-ELM yielded the best averaged mean kappa value (0.7945), and SS-ELM also obtained good result (0.7721).

3.2.4 Unseen evaluation data results

We compared the proposed method with SVM, ELM, SAE, H-ELM, and SS-ELM on the evaluation data of BCI Competition IV Dataset 2a, and the averaged classification accuracies of all six algorithms were shown in Table 4. In general, HSS-ELM outperformed the other five algorithms in terms of classification accuracy. Specifically, our method gained the best mean accuracy on subjects A01, A02, A06, and A07, while SVM performed best on subjects A03 and A08, and SS-ELM achieved the best result on subject A05, and SAE performed best on subject A04. Taking subject A01 for example, the average classification accuracy by our method was 81.14%, better than H-ELM (76.38%) and SS-ELM (79.27%). For all nine subjects, our algorithm yielded the

highest average accuracy (67.76%), a 2.03% improvement over H-ELM, a 0.49% improvement over SS-ELM, a 0.95% improvement over SAE, a 2.71% improvement over ELM, and a 1.39% improvement over SVM.

3.2.5 Performance with different ratios of labeled training data

In this section, we aimed to analyze the performance of different algorithms with different ratio of labeled training data on BCI Competition IV Dataset 2a. For the 288 trials in training session, 28 samples were randomly selected as validation set, and 10–90% of the remaining 260 samples were randomly chosen as labeled training set while the remaining, combined with 288 samples from evaluation session, were used as unlabeled training set. At the same time, the 288 evaluation samples were still used as test set. We repeated the process of allocating samples for ten times. The results for four representative subjects (A01, A02, A07, A08) are shown in Fig. 4.

Taking advantages of the semi-supervised learning theory, HSS-ELM outperformed H-ELM significantly in any instances of limited labeled data and more abundant unlabeled data. For example, considering subject A01T at the labeled training set proportion of 0.1, the proposed method yielded the best average accuracy of 70.28%, a 11.32% improvement over H-ELM and 1.32% over SS-ELM. On the other hand, at labeled training ratio of 0.9, our method also resulted in the best average classification accuracy of 80.28%, 4.37% better than H-ELM, and 1.04% higher than SS-ELM. The same is true for the other subjects.

4 Discussion

In these experiments, the proposed HSS-ELM method exhibited an excellent performance in both classification and computational efficiency, as demonstrated in several

Table 3 Classification accuracy on the training data of BCI Competition IV Dataset 2a

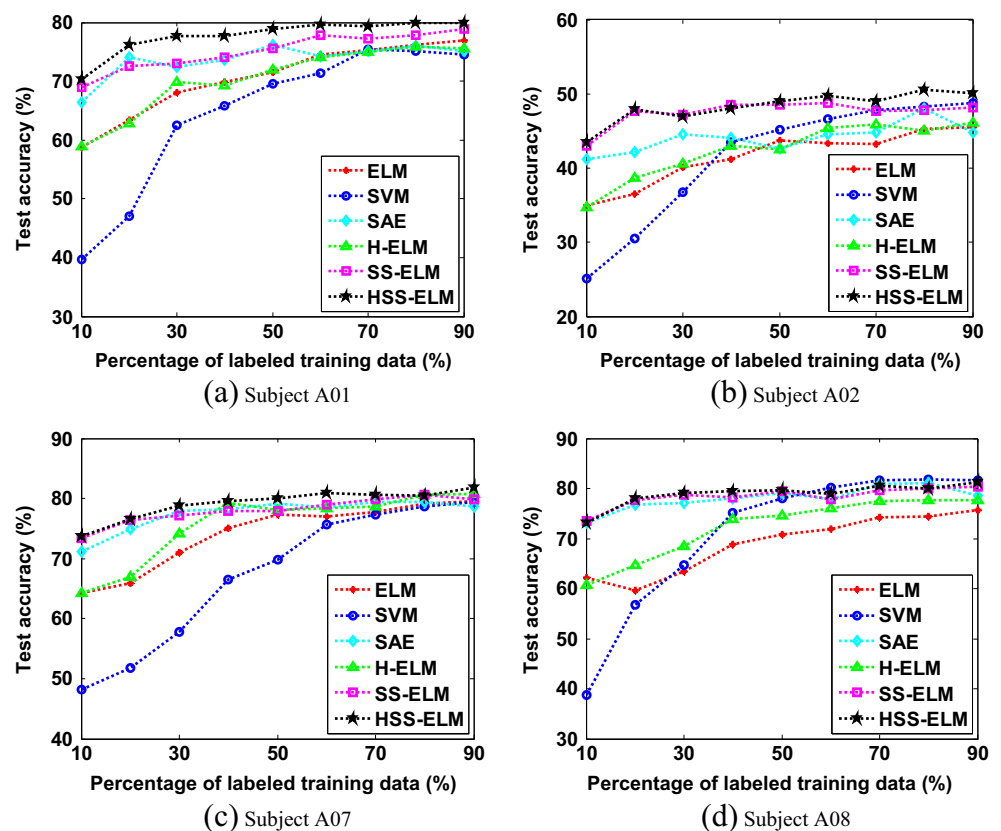
Methods	ELM	SVM	SAE	H-ELM	SS-ELM	HSS-ELM
A01	89.08	82.76	84.22	87.35	89.80	92.10
A02	62.06	59.77	63.42	64.94	66.81	70.26
A03	91.95	89.08	88.12	93.10	92.68	94.40
A04	64.94	67.81	68.71	70.11	77.16	80.86
A05	63.22	64.36	66.86	64.36	67.39	71.41
A06	67.81	68.96	62.65	68.96	74.86	78.31
A07	89.65	90.80	90.16	89.08	93.25	92.10
A08	88.51	88.48	88.68	90.80	93.25	89.22
A09	87.35	88.50	89.35	89.65	90.95	92.67
Mean accuracy	78.29	77.83	78.02	79.81	82.91	84.59
Mean kappa	0.7105	0.7044	0.7069	0.7308	0.7721	0.7945

Table 4 Classification accuracy on evaluation data of BCI Competition IV Dataset 2a

Methods	ELM	SVM	SAE	H-ELM	SS-ELM	HSS-ELM
A01	77.60	75.21	78.75	76.38	79.27	81.14
A02	45.56	47.88	45.87	46.35	48.68	49.86
A03	78.75	79.61	78.92	77.18	77.95	78.02
A04	59.34	64.17	64.34	61.67	62.99	63.33
A05	43.82	42.85	44.03	44.58	44.86	44.03
A06	46.84	46.49	48.47	47.95	48.95	49.44
A07	80.14	79.93	79.97	80.83	80.83	81.11
A08	80.17	82.11	81.56	80.34	81.21	81.49
A09	73.26	79.09	79.44	76.25	80.69	81.38
Mean accuracy	65.05	66.37	66.81	65.73	67.27	67.76
Mean kappa	0.5340	0.5516	0.5575	0.5431	0.5636	0.5701

benchmark datasets and MI EEG data. When compared with the pure supervised learning algorithms, i.e., SVM and ELM with shallow architecture as well as SAE and H-ELM with deep architecture, HSS-ELM achieved relatively better performance by exploring the unlabeled data effectively. Furthermore, HSS-ELM had superiority over the semi-supervised single-layer SS-ELM on all four datasets in section 3.1, and it was evidenced that HSS-ELM can boost the overall learning performance by using deep architecture to obtain multilayer sparse representations from the input data. In terms of computational efficiency, the slight increase in training time (several seconds) by HSS-ELM compared to H-ELM and

SS-ELM was negligible in practice, especially when considering the added improvement in classification accuracy. In particular, for the cases of data with small sample size but large dimension, i.e. Yale dataset, the training time of HSS-ELM is largely impacted by the deep architecture. In contrast, for cases of large sample size and small dimension, i.e. Waveform dataset, long training time is due to the calculation cost of the graph Laplacian in semi-supervised learning. In addition, in comparison with SVM, ELM, SAE, H-ELM and SS-ELM, our algorithm yielded favorable results when the number of labeled training data is small, since the incorporated semi-supervised learning can utilize the unlabeled data to improve

Fig. 4 Performance comparisons of six methods on different subjects

classification accuracy. With the increasing number of labeled training data, the classification accuracy increases for all the methods, albeit at different rate, the proposed method performed the best in general owing to the fact that its deep architecture allows for better information extraction.

We also proved the effectiveness of our method for identifying multi-class MI tasks on BCI Competition IV Dataset 2a. Due to the fact that EEG data typically exhibit low signal-to-noise ratio, applying the proposed HSS-ELM or other DL algorithms directly for feature extraction and classification on raw data usually yielded inadequate performance results for our research. As a result, handmade features are first extracted from the input raw EEG using the CSP approach, and then the different DL architectures are used to further learn compact feature representations. For subjects A01, A07, A08 and A09, our method resulted in acceptably high classification accuracy of above 80%. However, all the methods performed poorly in subject A05, and the mean accuracy of our method was 44.03%, slightly lower than that of SS-ELM.

Finally, we compare our algorithm to other existing relevant state-of-the-art methods: FBCSP [8] and PSRT [10]. The results are presented in Table 5. The average accuracy on the training data obtained by HSS-ELM was 84.59% (its corresponding kappa value was 0.7945), while the result for all nine subjects on the evaluation data using HSS-ELM was 67.76% (its kappa value was 0.5701), highest among all tested methods and comparable with the results reported in [8, 10].

5 Conclusion

In this paper, we have proposed a novel hierarchical semi-supervised framework via extreme learning machine (ELM), HSS-ELM, by virtue of the advantages of both SS-ELM and H-ELM. Different from SS-ELM, the proposed method adopted the multilayer architecture of DL and learned from the input raw data using the unsupervised sparse ELM-AE algorithm to obtain higher level features, allowing more hidden information to be extracted. On the other hand, in contrast with the supervised algorithm H-ELM, HSS-ELM used the unlabeled data to improve the classification performance according to the semi-supervised theory. The proposed HSS-ELM achieved high-level representation with multilayer encoding from both labeled data and unlabeled data, and had

the superiority in train efficiency and classification accuracy in various simulations. Our empirical results have demonstrated that the proposed HSS-ELM is effective at learning high-level feature representation and capable of achieving excellent classification performance in several benchmark datasets as well as MI EEG datasets, compared with SVM, ELM, SAE, H-ELM and SS-ELM. However, there are still several questions to be further investigated in future work. Compared with H-ELM and SS-ELM, we enhanced the recognition precision while increased the complexity of the model, i.e. adding the graph Laplacian in objective function for semi-supervised learning. It will be feasible to use the online sequential version to solve the problem. In classification experiments on MI EEG data, the classical common spatial pattern (CSP) approach is usually employed to extract the low-level features from raw EEG data. Recently, some advanced CSP variants, such as augmented complex common spatial pattern (ACCSP) [9] and probabilistic common spatial pattern (P-CSP) [40], have yielded promising results. By incorporating these advancements with the presented algorithm in this paper, it will serve to improve the classification and evaluation of MI EEG signals.

Acknowledgments This work is supported by National Nature Science Foundation under Grant (No.61201302, 61671197 and 61601162), Zhejiang Province Natural Science Foundation (LY15F010009), Guangdong Provincial Work Injury Rehabilitation Center and the University of Houston. The authors would like to acknowledge the BCI Competition IV Dataset 2a which was used to test the algorithms proposed in this study.

Compliance with ethical standards

Conflict of interests The authors declare that they have no conflict of interests.

References

1. Wolpaw J, Wolpaw E W (2012) Brain-computer interfaces: principles and practice. OUP USA
2. Pfurtscheller G, Neuper C (2001) Motor imagery and direct brain-computer communication. *Proc IEEE* 89(7):1123–1134
3. Sanei S, Chambers J A (2013) EEG signal processing. John Wiley & Sons
4. Samek W, Kawanabe M, Müller KR (2014) Divergence-based framework for common spatial patterns algorithms. *IEEE Rev Biomed Eng* 7:50–72
5. She Q, Gan H, Ma Y et al (2016) Scale-dependent signal identification in low-dimensional subspace: motor imagery task classification. *Neural Plast* 7431012
6. Thomas E, Fruitet J, Clerc M (2013) Combining ERD and ERS features to create a system-paced BCI. *J Neurosci Methods* 216(2): 96–103
7. Park C, Looney D, ur Rehman N et al (2013) Classification of motor imagery BCI using multivariate empirical mode decomposition. *IEEE Trans Neural Syst Rehab Eng* 21(1):10–22

Table 5 Comparison between our algorithm and other methods on BCI Competition IV Dataset 2a

Methods	FBCSP [8]	PSRT [10]	HSS-ELM
Cross-validation results	0.663	0.78	0.7945
Unseen evaluation data results	0.569	–	0.5701

8. Ang KK, Chin ZY, Wang C et al (2012) Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Front Neurosci* 6:39
9. Park C, Took CC, Mandic DP (2014) Augmented complex common spatial patterns for classification of noncircular EEG from motor imagery tasks. *IEEE Trans Neural Syst Rehab Eng* 22(1):1–10
10. Djemal R, Bazyed AG, Belwafi K et al (2016) Three-class EEG-based motor imagery classification using phase-space reconstruction technique. *Brain Sci* 6(3):36
11. Bhattacharyya S, Khasnobish A, Chatterjee S et al (2010) Performance analysis of LDA, QDA and KNN algorithms in left-right limb movement classification from EEG data. *Proc IEEE Int Conf Syst Med Biol* 126–131
12. Übeyli ED (2009) Combined neural network model employing wavelet coefficients for EEG signals classification. *Digital Signal Process* 19(2):297–308
13. Wipf D, Nagarajan S (2009) A unified Bayesian framework for MEG/EEG source imaging. *NeuroImage* 44(3):947–966
14. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70(1):489–501
15. Huang GB, Zhou H, Ding X et al (2012) Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man Cybern B (Cybernetics)* 42(2):513–529
16. Zong W, Huang GB, Chen Y (2013) Weighted extreme learning machine for imbalance learning. *Neurocomputing* 101:229–242
17. Lendasse A, He Q, Miche Y et al (2014) Advances in extreme learning machines (ELM2012). *Neurocomputing* 128:1–3
18. Duan L, Bao M, Miao J et al (2016) Classification based on multi-layer extreme learning machine for motor imagery task from EEG signals. *Procedia Comput Sci* 88:176–184
19. Ding S, Zhang N, Xu X et al (2015) Deep extreme learning machine and its application in EEG classification. *Math Probl Eng* 2015
20. Peng Y, Lu BL (2016) Discriminative manifold extreme learning machine and applications to image and EEG signal classification. *Neurocomputing* 174:265–277
21. Zhang Y, Jin J, Wang X et al (2016) Motor imagery EEG classification via Bayesian extreme learning machine. *Proc IEEE Int Conf Inf Sci Technol (ICIST)* 27–30
22. Tang J, Deng C, Huang GB (2016) Extreme learning machine for multilayer perceptron. *IEEE Trans Neural Netw Learn Syst* 27(4):809–821
23. Kasun LLC, Zhou H, Huang GB et al (2013) Representational learning with ELMs for big data. *IEEE Intell Syst* 28(6):31–34
24. Uzair M, Shafait F, Ghanem B et al (2015) Representation learning with deep extreme learning machines for efficient image set classification. *arXiv preprint arXiv:1503.02445*
25. Zhou Y, Liu B, Xia S et al (2015) Semi-supervised extreme learning machine with manifold and pairwise constraints regularization. *Neurocomputing* 149:180–186
26. Huang G, Song S, Gupta JND et al (2014) Semi-supervised and unsupervised extreme learning machines. *IEEE Trans Cybern* 44(12):2405–2417
27. Yu W, Zhuang F, He Q et al (2015) Learning deep representations via extreme learning machines. *Neurocomputing* 149:308–315
28. Gan H, Luo ZZ, Meng M et al (2016) A risk degree-based safe semi-supervised learning algorithm. *Int J Mach Learn Cybern* 7(1):85–94
29. Belkin M, Niyogi P, Sindhvani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J Mach Learn Res* 7(Nov):2399–2434
30. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507
31. Vincent P, Larochelle H, Bengio Y et al (2008) Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning*, 1096–1103
32. Vincent P, Larochelle H, Lajoie I et al (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 11(Dec):3371–3408
33. Salakhutdinov R, Larochelle H (2010) Efficient learning of deep Boltzmann machines. *Proc Int Conf Artif Intell Stat* 693–700
34. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
35. Blake CC, Merz CJ UCI repository of machine learning databases. Available online: <http://archive.ics.uci.edu/ml/>
36. The Yale Face Database Available online: <http://vision.ucsd.edu/content/yale-face-database>
37. Graz University BCI Competition IV Datasets 2a. Available online: <http://www.bbc.de/competition/iv/#dataset2a>
38. Meng M, Zhu J, She Q et al (2016) Two-level feature extraction method for multi-class motor imagery EEG. *Acta Automat Sin* 42(12):1915–1922
39. Långkvist M, Karlsson L, Loutfi A (2012) Sleep stage classification using unsupervised feature learning. *Adv Artif Neural Syst* 2012:5
40. Wu W, Chen Z, Gao X et al (2015) Probabilistic common spatial patterns for multichannel EEG analysis. *IEEE Trans Pattern Anal Mach Intell* 37(3):639–653

Qingshan She received the B.S. and M.S. degrees in materials science and engineering from Lanzhou University of Technology, Lanzhou, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China. He is currently an Associate Professor with the School of Automation, Hangzhou Dianzi University. His research interests include machine learning and pattern recognition, biomedical signal processing, brain-computer interface, and its applications.

Bo Hu received the B.S. degree in electrical engineering and automation from Hangzhou Dianzi University, Hangzhou, China. He is a M.S. candidate in the School of Automation, Hangzhou Dianzi University. His research interests include semi-supervised learning and EEG signal processing.

Zhizeng Luo received the B.S. degree in mechatronic engineering from the University of Electronic Science and Technology of China, Chengdu, and the Ph.D. degree in industrial automation from Zhejiang University, Hangzhou, China. He is currently a Professor with the School of Automation, Hangzhou Dianzi University. He is also the director with the Institute of Intelligent Control and Robotics. His research interests include pattern recognition and intelligent systems, rehabilitation robot, and detection and processing of biological information.

Thinh Nguyen received his B.S. degree in biomedical engineering from the University of Houston. Currently, he is a PhD student in the Department of Biomedical Engineering at the University of Houston. His research interests include multimodal neural imaging with concurrent EEG and fMRI, image processing, and signal processing.

Yingchun Zhang is currently an Assistant Professor in the Department of Biomedical Engineering at the University of Houston. He received his PhD in Electrical Engineering at Zhejiang University, China, in collaboration with the Department of Biomedical Engineering at the University of Minnesota. His research interests focus on understanding the mechanisms of electromagnetic activities in biological tissue and systems, computational modeling and analysis of organ systems, and developing non-invasively functional imaging technologies to aid clinical diagnosis of dysfunction in the human body.