Cogn Comput (2017) 9:758–765
DOI 10.1007/s12559-017-9494-0

Motor Imagery EEG Classification Based on Kernel Hierarchical Extreme Learning Machine

Lijuan Duan[1,2] **· Menghu Bao**[1,3] **· Song Cui**[1,3] **· Yuanhua Qiao**[4] **· Jun Miao**[5]

Received: 11 October 2016 / Accepted: 18 July 2017 / Published online: 1 August 2017

Abstract As connections from the brain to an external device, Brain-Computer Interface (BCI) systems are a crucial aspect of assisted communication and control. When equipped with well-designed feature extraction and classification approaches, information can be accurately acquired from the brain using such systems. The Hierarchical Extreme Learning Machine (HELM) has been developed as an effective and accurate classification approach due to its deep structure and extreme learning mechanism. A classification system for motor imagery EEG signals is proposed based on the HELM combined with a kernel, herein called the Kernel Hierarchical Extreme Learning Machine (KHELM). Principle Component Analysis (PCA) is used to reduce the dimensionality of the data, and Linear Discriminant Analysis (LDA) is introduced to push the features away from different classes. To demonstrate the performance, the proposed system is applied to the BCI competition 2003 Dataset Ia, and the results are compared with those from state-of-the-art methods; we find that the accuracy is up to 94.54%.

Keywords Electroencephalogram classification · Motor imagery · Extreme learning machine · Hierarchical extreme learning machine · Kerne l-based extreme learning machine

Introduction

Brain science is one of the most challenging research fields and attempts to reveal and perceive the inner mechanisms of brain activities and functions. A Brain-Computer Interface (BCI) [1, 8, 28] provides a bridge between the brain and an external device for brain science studies. Electroencephalogram (EEG) [2, 4] data are collected using BCI devices to obtain information about brain activities. EEGs present the responses of the brain to external stimuli and are widely used for monitoring brain activities. Motor imagery EEG [20] has aroused the enthusiasm of many researchers and plays an important role in dyskinesia study, especially for patients with neuromuscular disorders.

EEG recognition is divided into two parts: feature extraction and classification. For feature extraction, various methods have been developed such as band power (BP) [21], power spectral density (PSD) values [26], and wavelet package (WP) [29]. For classification, support vector machines (SVM) [3], neural networks [22], and naive Bayes [15] are widely used. Gert [20] uses band power values and a neural network for the online classification of right and left motor imagery, and they also analyze offline classification using an adaptive autoregressive (AAR) model. An algorithm for epileptic seizure detection using lacunarity and Bayesian linear discriminant analysis (BLDA) [30] is proposed for

✉ Jun Miao
jmiao@bistu.edu.cn

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

[2] Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, Beijing, China

[3] National Engineering Laboratory for Critical Technologies of Information Security Classified Protection, Beijing 100124, China

[4] College of Applied Science, Beijing University of Technology, Beijing 100124, China

[5] Beijing Key Laboratory of Internet Culture and Digital Dissemination Research and School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China

long-term intracranial EEG. Luis [18] presents an adaptive semi-supervised classification algorithm for online multi-class motor imagery EEG and achieves the highest accuracy (77%) to date. A combination of linear and nonlinear EEG features and a k-nearest neighbor (k-NN) classifier [16] is used to detect mild depression in college students with a high accuracy.

Although neural networks have been widely used in EEG classification, the iteration process in the parameter fine tuning limits the learning speed. Hence, a type of single-hidden-layer feed-forward neural network (SLFN) named the Extreme Learning Machine (ELM) is proposed by Huang [5, 11, 31], in which the weight between the input and hidden layers is randomly generated, and the weight between the hidden and output layers is computed without an iteration process. Recently, the HELM [24], an approach based on the ELM, was presented and extended the ELM into a deep ELM neural network without back propagation or an iteration process. The HELM performances better than the ELM; however, the random parameters in the network influence the stability of the network. Combining with the Kernel-based Extreme Learning Machine (Kernel-ELM) [9], the KHELM is proposed. This method not only maintains the high training speed but also performs better than the HELM and Kernel-ELM. Therefore, it is applied to the classification of motor imagery EEG with PCA and LDA.

The contributions of this paper are as follows:

(1) To improve the stability of the HELM, the HELM is combined with a Gaussian Kernel to form the KHELM method.
(2) The KHELM is introduced first to the classification of motor imagery EEG signals, and a classification system based on the KHELM is proposed.
(3) The classification system performs better than state-of-the-art methods with respect to accuracy as well as training and testing speeds.

The remainder of this paper is organized as follows. In the “Method” section, the classification system, including feature extraction and classification, is presented. In the “Experiments” section, experiments are conducted on the BCI competition 2003 Dataset Ia, and the performance results are given. In the “Conclusion” section, the conclusions are presented.

Method

The original EEG signals are easily affected by the environment, which results in a low signal-to-noise ratio (SNR). In addition, given their complicated components, EEG signals are difficult to recognize. In the proposed approach, we first partition the sample with overlaps into segments. Because the partitioned EEG data are still characterized by high dimensionality and low SNR, it is important to extract typical and distinctive features. Then, PCA is used to extract the principal dimensions of each segment. However, PCA cannot reveal information indicating different classes. After PCA, an LDA process is introduced to decrease the coupling between the classes. To further integrate and analyze all segments in each sample, the features are rearranged as the inputs of the KHELM for classification. Therefore, the system involves two parts: (1) feature extraction with PCA and LDA and (2) classification based on the KHELM. The framework of the proposed method is shown in Fig. 1.

Feature Extraction with PCA and LDA

As a common tool in dimensional reduction, PCA projects the original high-dimensional data into lower dimensional data by relying on the maximization of the total scatter matrix of the projected samples [13]. In this way, the SNR of the signal can be increased. Then, LDA is introduced to search for the best projection direction and decrease the coupling between classes, which reduces the dimension to 1 for binary-class data [19]. Features obtained by PCA and LDA are more discriminative.

Denoting the training data as Z and the testing data as T, the procedure is described as follows. First, the eigenvectors and eigenvalues of all the data are calculated using covariance matrix decomposition. The eigenvectors corresponding to the top l eigenvalues form the basis $L_{PCA} = [\phi_1, \phi_2, \ldots, \phi_l]$. Because a larger eigenvalue represents a larger contribution rate of the principle component, we

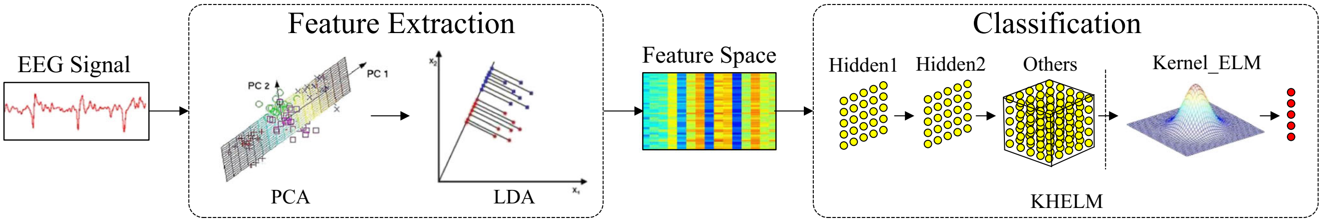


Fig. 1 The framework of the system

choose l principle components using a threshold of the accuracy contribution rate (ACR). Then, we obtain the training features $Z_{PCA} = Z \cdot L_{PCA}$ and the testing features $T_{PCA} = T \cdot T_{PCA}$. Finally, LDA is applied to Z_{PCA} and T_{PCA} to obtain the remapped matrices X_{train} and X_{test} as the final features.

Classification Based on KHELM

The ELM is advantageous as a result of its learning speed for machine learning and artificial intelligence. However, the ELM is limited to simple data due to its shallow architecture. Therefore, the Hierarchical Extreme Learning Machine is built, and the structure is extended into a more complex architecture [24]. In this paper, the KHELM is proposed based on a combination of the HELM and the Kernel-ELM. The technique is divided into two parts, as shown in Fig. 2: (1) the unsupervised feature extraction based on the ELM-based sparse auto-encoder and (2) the supervised feature classifier based on the Kernel-ELM.

For the KHELM learning algorithm, the first-layer weights of the KHELM are set as the transposition of the output weights β_1 that are learned by the ELM-based Sparse Auto-Encoder shown in Fig. 2a, and the i^{th}-layer weights of the KHELM are set as the transposition of the output weights β_{i+1} of the ELM-based Sparse Auto-Encoder shown in Fig. 2b. The framework of the KHELM includes two phases: multi-layer forward encoding followed by the Kernel-ELM classifier.

For the training features $\{(x_i, y_i)\}_{i=1}^{N} \epsilon R^d \times R^m$ of the EEG data, x_i is the input vector and y_i is the class label. The output of each hidden layer is represented as follows:

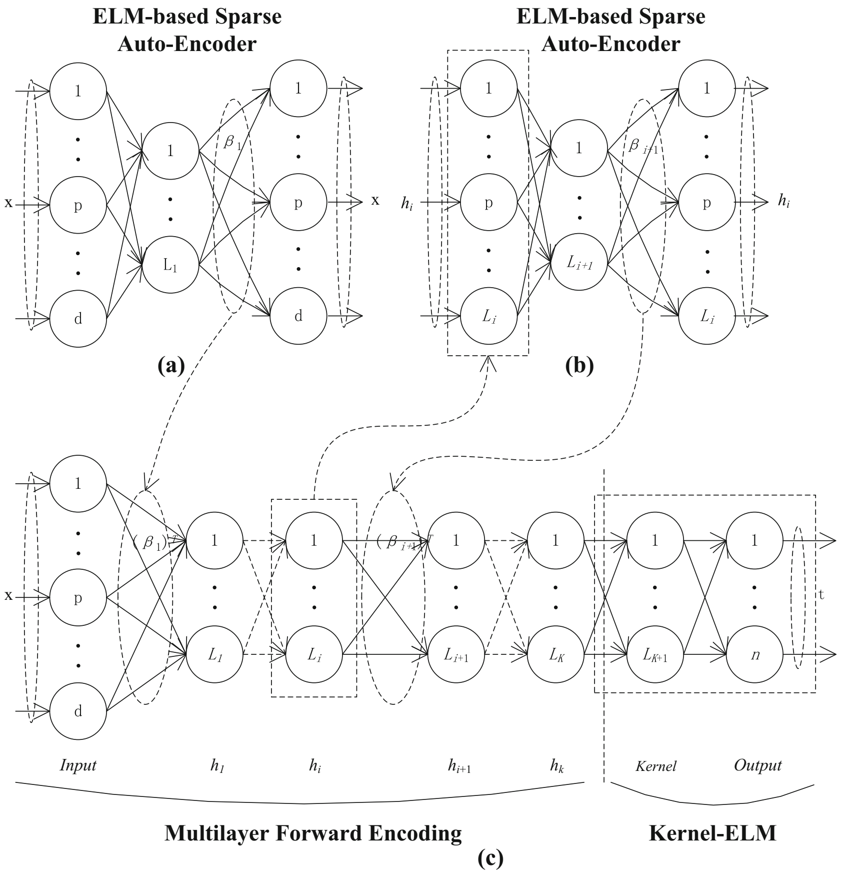
$$H_i = g(H_{i-1} \cdot \beta^T), \tag{1}$$

where H_i is the i^{th}-hidden-layer output matrix, and the input x is considered as the 0^{th} hidden layer, where $i = 0$. g is the activation function of the hidden layers, and β is the output weight. It is shown in Fig. 2 that once the feature of the previous hidden layer is extracted, the weights of the current hidden layer can be learned without fine tuning.

To extract the abstract representation of the input features, the ELM-based Sparse Auto-Encoder is chosen to model the feature representation part of the KHELM, and its structure is shown in Fig. 2a.

The ELM is a promising technology with a high learning speed and trivial human intervention requirement [10]. Compared with other techniques, the ELM provides a better generalization performance and can obtain globally optimal solutions [12]. We use the ELM to build an auto-encoder with the target replaced by the input, called the ELM-AE.

Fig. 2 The architecture of the KHELM learning algorithm [24]



To generate more sparse and compact features of the inputs, an l_1 constraint is added to extend it into the ELM-based Sparse Auto-Encoder. Its optimization model is extended as follows:

$$O_\beta = \arg\min_\beta \{\|H\beta - X\|^2 + \|\beta\|_{l1}\}, \tag{2}$$

where X denotes the input data and output target and β is the weight between the hidden and output layers. With the l_1 and l_2 constraints, the ELM-based Sparse Auto-Encoder is less sensitive to the input, and thus, its generalization performance is enhanced.

The method of the ELM-based Sparse Auto-Encoder is summarized in Algorithm 1, and its structure is shown in Fig. 2a. In the ELM-based Sparse Auto-Encoder, with the weight β, the output H of the hidden layer is decoded into the input X. It is proved that the input X can be transformed into H^* with the weight β^T, where H^* is the output of the hidden layer in the KHELM, being an approximation to H, and β^T is the transposition of β.

Algorithm 1 The ELM-based Sparse Auto-Encoder

```
Function Name: $sparse\_elm\_ae(X, H)$
Input: input data $X = \{x_i\}_{i=1}^N$, output matrix of hidden nodes $H = \{h_i\}_{i=1}^L$
01. Calculate the Lipschitz constant $\gamma$ of the gradient $\nabla p$ of the function
    $\|H\beta - X\|^2$
02. Calculate $\beta$ iteratively, $y_1 = \beta_0, t_1 = 1, i \geq 1$
    Step 1. $\beta_i = s_\gamma(y_i)$,
            where $s_\gamma = \arg\min_\beta\{\frac{\gamma}{2}\|\beta - (\beta_{i-1} - \frac{1}{\gamma}(\beta_{i-1}))\|^2 + \|\beta\|_{l1}\}$
    Step 2. $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$
    Step 3. $y_{i+1} = \beta_i + (\frac{t_{i-1}}{t_{i+1}})(\beta_i - \beta_{i-1})$
Output: output weight $\beta$
```

The KELM method is the combination of the ELM and a kernel. The output of the hidden layer HH^T is replaced by the kernel function

$$HH^T = \Phi_{ELM} = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_L) \\ \vdots & \ddots & \vdots \\ K(x_L, x_1) & \cdots & K(x_L, x_L) \end{bmatrix}, \tag{3}$$

where $K(\cdot)$ is the kernel function; the most popular kernel of the Kernel-ELM is the Gaussian kernel $K(x_i, x_j) = exp(-\|x_i - x_j\|/K)$, in which K is the kernel parameter.

The output of the Kernel-ELM is as follows:

$$\begin{aligned} f(x) &= h(x)\beta = h(x)H^T(HH^T + \frac{I}{C})^\dagger Y \\ &= \begin{bmatrix} K(x, x_1) \\ \ddots \\ K(x, x_L) \end{bmatrix}^T (\Phi_{ELM} + \frac{I}{C})^\dagger T, \end{aligned} \tag{4}$$

where C is the regularization coefficient, $(\cdot)^\dagger$ represents the Moore-Penrose generalized inverse, and T is the label of the samples.

The procedures for training the KHELM are given in Algorithm 2.

Using the ELM-based Sparse Auto-Encoder, the KHELM is extended into a deep neural network, and the features are extracted in a more compact and sparse form again, which improves the KHELM performance in classification. Meanwhile, because the hidden layer is replaced by a kernel function in the Kernel-ELM, the randomness of the parameters in the KHELM is lower than in the HELM, which provides a more stable network for the KELM compared to the HELM.

Algorithm 2 The Kernel Hierarchical Extreme Learning Machine

```
Input: input data $H_0 = \{x_i\}_{i=1}^N$, label $T = \{y_i\}_{i=1}^N$,
       number of hidden neurons $L_i$, weights of random mapping $b_i$
       regularization coefficient $C$ and kernel function $K$
01. Train the first hidden layer
    Step 1. Calculate the output of the hidden nodes of the ELM-based Sparse
            Auto-Encoder:
            $A_1 = g(H_0 \cdot b_1)$
    Step 2. Calculate the output weights $\beta_1$:
            $\beta_1 = sparse\_elm\_ae(A_1, H_0)$
    Step 3. Calculate the output matrix of the hidden nodes:
            $H_1 = g(H_0 \cdot (\beta_1)^T)$
02. Train the $i^{th}$ hidden layer
    Step 1. Calculate the output of the hidden nodes of the ELM-based Sparse
            Auto-Encoder:
            $A_i = g(H_{i-1} \cdot b_i)$
    Step 2. Calculate the output weights $\beta_i$:
            $\beta_i = sparse\_elm\_ae(A_i, H_{i-1})$
    Step 3. Calculate the output matrix of the hidden nodes:
            $H_i = g(H_{i-1} \cdot (\beta_i)^T)$
03. Train the Kernel ELM
     Calculate the output:
            $O = K(H_i, H_i) \cdot (\Phi_{ELM} + \frac{I}{C})^\dagger T$
Output: All the output weights $\beta_i$
```

Experiments

Dataset Description

In this section, we first introduce the EEG dataset for the motor imagery. The dataset is from the BCI competition 2003 Dataset Ia taken from a healthy subject. The subject is asked to imagine the movement of a cursor up and down on a computer screen. Meanwhile, his cortical potentials are recorded, and he receives visual feedback of his slow cortical potentials. Cortical positivity leads to a downward movement of the cursor on the screen, whereas cortical negativity leads to an upward movement of the cursor. All the

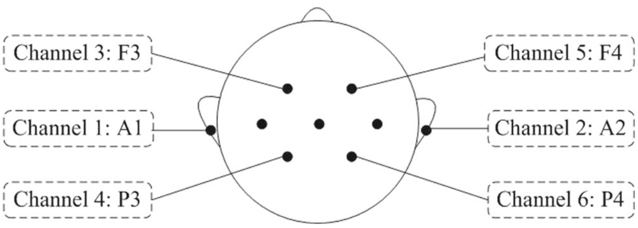


Fig. 3 Distribution of EEG electrodes for 6 channels

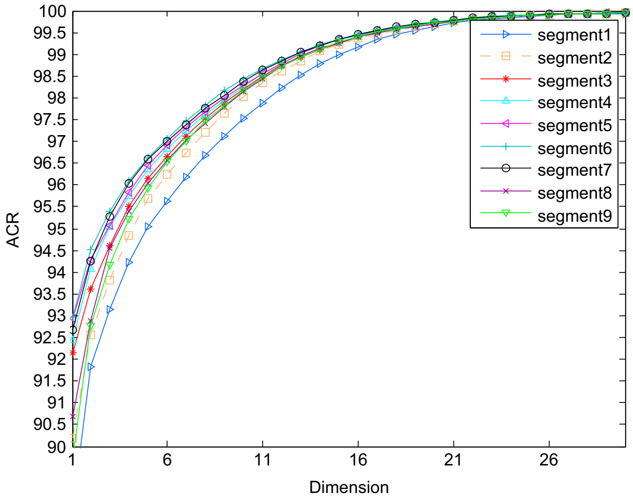


Fig. 4 ACR for PCA of the segments of channel A1

trails are composed of a training set (268 trials, with 135 for class 0 and 133 for class 1) and a testing set (293 trials, with 147 for class 0 and 146 for class 1), and each trial lasts 6 s. During each trial, the task is visually presented by a highlighted goal at either the top or bottom of the screen to indicate negativity or positivity from 0.5 s until the end of the trial. The visual feedback is presented from 2 s to 5.5 s. Only this 3.5-s interval for each trial is used for training and testing. The 256 Hz sampling rate and 3.5-s recording length result in 896 samples per channel for each trial. Here, the central parietal region electrode (Cz-Mastoids) is chosen as the reference electrode, and the remaining six electrodes, which are located as shown in Fig. 3, are used to collect the signals.

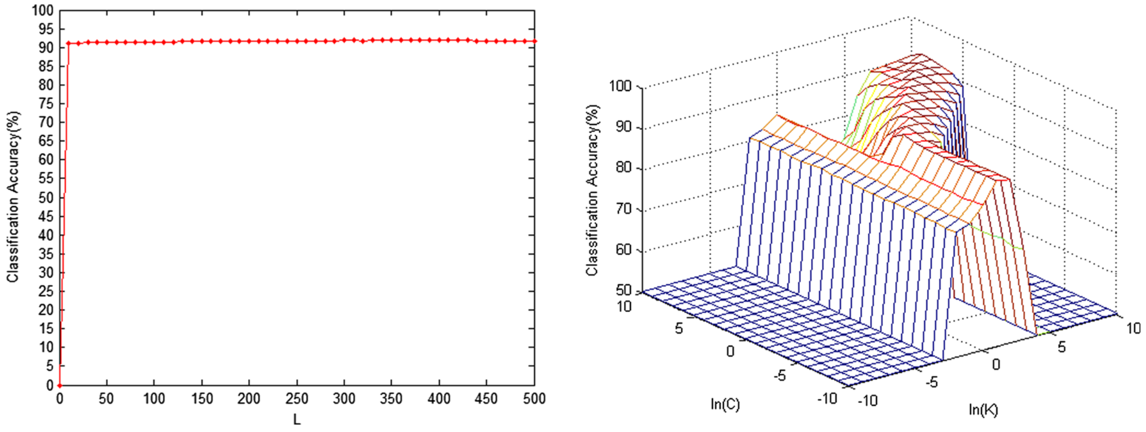
In the experiment, first, we only choose the signals from channel 1 and channel 2 (A1 and A2) to generate the discriminative feature sets based on previous work [6]. Then, the continuous recording samples are partitioned to subepochs of 500 ms, with a 125 ms overlap. Therefore, the raw data are split into 9 segments for each channel, and each segment has 128 dimensions. Thus, there are 18 segments in the end.

Discussion of Results

Parameter Selection for Feature Extraction For the feature extraction, we use a combination of PCA and LDA. First, PCA is used for the dimensional reduction. Although the Accuracy Contribution Rate (ACR) is greater than 99% for all segments shown in Fig. 4, the corresponding feature dimension of 16 is selected. Then, the 16-dimensional features of each segment are reduced into 1-dimensional features using LDA. Finally, the features of channels A1 and A2 are combined to form the input of the KHELM in 18 dimensions.

Parameter Selection of KHELM For the KHELM, three parameters are required for tuning: the parameter C for the regularized least mean square calculation, the number of hidden nodes L, and the parameter of the kernel function K. In the experiment, C is confined to $\{10^{-10}, 10^{-9}, \ldots, 10^{9}, 10^{10}\}$, L is confined to $\{500, 600, \ldots, 1400, 1500\}$, and K is confined to $\{10^{-10}, 10^{-9}, \ldots, 10^{9}, 10^{10}\}$. The influences of L, C and K on the performance of the KHELM are shown in Fig. 5.

Figure 5a shows the impact of L on the performance of the KHELM, and the parameters C and K are fixed beforehand. It is shown in Fig. 5a that the accuracy slightly fluctuates around a central line, and the amplitude of the fluctuation is less than 1%. Therefore, it is demonstrated that the KHELM is not sensitive to the node number of the hidden layers. In Fig. 5b, the 3-D accuracy curves of the KHELM in terms of C and K are shown. Clearly, C and K are the main influential factors for the KHELM. Therefore,



(a)

(b)

Fig. 5 Classification accuracy of KHELM. **a** Accuracy of the KHELM with respect to L. **b** Accuracy of the KHELM with respect to C and K

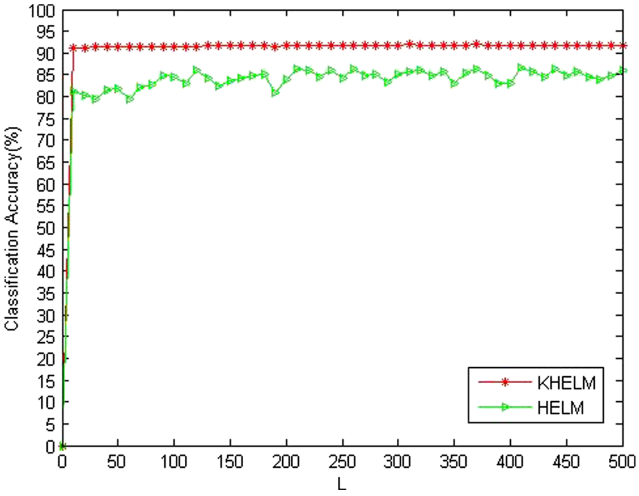


Fig. 6 Comparison of the KHELM and the HELM

one must be careful when selecting C and K, and the best choice for C and K is 10^4 and 10^5, respectively.

Evaluation of the Classification System To demonstrate the effectiveness of the classification system, first, the performance of the KHELM is compared with other classifiers based on the ELM using the same features. The performance of the state of the art is also given on the same dataset as a comparison.

For all experiments, the hardware and software specifications are as follows: PC, Intel-i5 3.30 GHz CPU, 8.00 GB RAM, Windows 7, and MATLAB R2012a.

To certify the superiority of the KHELM, the accuracies of 50 experiments are averaged for comparison with the HELM based on a similar structure. The results are shown in Fig. 6. It is shown that the average accuracy of the KHELM is higher than that of the HELM, and the amplitudes of the fluctuations of the KHELM are substantially smaller than those of the HELM. Clearly, the HELM is more sensitive to the node number of the hidden layers than is the KHELM. Therefore, the KHELM is more stable than the HELM.

We compare the performance of the KHELM with other ELM-based methods such as the avg-ELM, Kernel-ELM, V-ELM [7], ML-ELM, and HELM. The result from the avg-ELM is an average over 50 ELMs. To obtain the results of the V-ELM, we use 50 ELMs to vote. The ML-ELM is a simple stacked layer-by-layer architecture with the ELM-AE; here, we set it to have three hidden layers, and the numbers of hidden nodes are (20, 20, 2000). The HELM has three hidden layers, similar to the KHELM. Table 1 gives the results of the comparison. The average accuracy of the KHELM is 91.99%, while those of the HELM and Kernel-ELM are 90.94 and 91.81%, respectively. Meanwhile, the best accuracies of the KHELM and HELM are equal: 94.54%. The training and testing speeds of the KHELM are approximately five times higher than those of the HELM. Specifically, in terms of both accuracy and speed, the performance of the KHELM is superior to that of the Kernel-ELM and the HELM. It is obvious (from Table 1) that the best accuracy achieved by the KHELM is the highest accuracy, and the speed is higher than that of the other methods based on the ELM, except for the avg-ELM. Because the avg-ELM has only one hidden layer, it obviously obtains a high speed. The results demonstrate that the sparse weights and kernel function have an important effect on the performance of the system with respect to speed and accuracy.

We also compare our system with some state-of-the-art methods on the same dataset. Table 2 gives the final results of the selected methods. It is shown that the proposed system performs better than the state-of-the-art methods. The accuracy increase is at least 2.39%. The results demonstrate that the classification system is more effective for motor imagery EEG data, and introducing the KHELM to the classification of motor imagery EEG data is thus necessary.

Table 1 Comparison with other methods based on the ELM

Method	♯	Accuracy (%)	Training time (s)	Testing time (s)
avg-ELM	Average	89.20±1.63	0.0006±0.0044	0.0025±0.0086
	Best	92.83	–	–
Kernel-ELM	–	91.81	0.0024±0.0006	0.0017±0.0002
V-ELM [7]	Average	92.30±0.43	0.1033±0.0478	0.0118±0.0177
	Best	93.52	–	–
ML-ELM	Average	89.83±1.34	0.3136±0.0694	0.0059±0.0007
	Best	94.20	–	–
HELM	Average	90.94±1.06	0.3182±0.021	0.0173±0.001
	Best	94.54	–	–
KHELM	Average	91.99±0.72	0.0640±0.0024	0.0089±0.0004
	Best	94.54	–	–

Table 2 Comparison with various related methods

Classification method	Feature extraction	Accuracy (%)
Linear (2004) [17]	Gamma band power combined with SCP	88.70
Bayes (2005) [23]	Combing SCP with the spectral centroid	90.44
Neural Network (2005) [27]	SCP and beta band specific energy	91.47
Neural Network (2008) [25]	Wavelet package	90.80
k-NN (2010) [14]	Coefficients of the second-order polynomial	92.15
KHELM	PCA and LDA	94.54

Conclusion

In this paper, a new system is proposed for motor imagery EEG signal classification based on PCA, LDA, and the KHELM. To improve the stability of the HELM, the KHELM is proposed in this paper; the results demonstrate that it is a successful improvement on the HELM. Compared with various other feature extraction methods, features obtained by PCA and LDA are suitable for motor imagery signals with low dimensionality and high discrimination. Compared with other classification methods, the KHELM is easy to implement and requires trivial human intervention. When deep and compact feature information is extracted, the KHELM achieves a high accuracy and high speed in classification. The best results from the experiment demonstrate that the system is effective and efficient when applied to the binary-class signals of motor imagery.

Acknowledgements This research is partially sponsored by the Natural Science Foundation of China (Nos. 61672070, 81471770, 61572004 and 61650201), the Beijing Municipal Natural Science Foundation (4152005, 4162058), the Science and Technology Program of Tianjin (15YFXQGX0050), and the Qinghai Natural Science Foundation (2016-ZJ-Y04).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflicts of interest.

Informed Consent Informed consent was not required, as no humans or animals were involved.

Human and Animal Rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Blankertz B, Muller K-R, Curio G, Vaughan TM, Schalk G, Wolpaw JR, Schlogl A, Neuper C, Pfurtscheller G, Hinterberger T, Schroder M, Birbaumer N. The BCI, competition 2003: progress and perspectives in detection and discrimination of EEG single trials. IEEE Trans Biomed Eng. 2004;51(6):1044–1051.
2. Buzsáki G, Anastassiou CA, Koch C. The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes. Nat Rev Neurosci. 2012;13(6):407–420.
3. Camps-Valls G, Bruzzone L. Kernel-based methods for hyperspectral image classification. IEEE Trans Geosci Remote Sens. 2005;43(6):1351–1362.
4. da Silva FL. EEG and MEG: relevance to neuroscience. Neuron. 2013;80(5):1112–1128.
5. Deng CW, Huang GB, Jia Xu, Tang JX. Extreme learning machines: new trends and applications. Sci Chin Inf Sci. 2015;58(2):1–16.
6. Duan L, Qi Z, Yang Z, Miao J. Research on heuristic feature extraction and classification of eeg signal based on bci data set. Res J Appl Sci Eng Technol. 2013;5(3):1008–1014.
7. Duan L, Zhong H, Miao J, Yang Z, Ma W, Zhang X. A voting optimized strategy based on ELM for improving classification of motor imagery BCI data. Cogn Comput. 2014;6(3):477–483.
8. Guest editorial the third international meeting on brain-computer interface technology. Making a difference. IEEE Trans Neural Syst Rehab Eng. 2006;14(2):126–127.
9. Huang GB. An insight into extreme learning machines: random neurons, random features and kernels. Cogn Comput. 2014;6(3):376–390.
10. Huang G-B, Wang DH, Lan Y. Extreme learning machines: a survey. Int J Mach Learn Cybern. 2011;2(2):107–122.
11. Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE international joint conference on neural networks (IEEE Cat. No.04CH37541). Institute of Electrical and Electronics Engineers (IEEE).
12. Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: theory and applications. Neurocomputing. 2006;70(1–3):489–501.
13. Jolliffe IT. Principal component analysis and factor analysis. In: Principal component analysis. Springer Nature; 1986 p 115–128.
14. Kayikcioglu T, Aydemir O. A polynomial fitting and k-NN based approach for improving classification of motor imagery BCI data. Pattern Recogn Lett. 2010;31(11):1207–1215.
15. LeVan P, Urrestarazu E, Gotman J. A system for automatic artifact removal in ictal scalp EEG based on independent component analysis and bayesian classification. Clin Neurophysiol. 2006;117(4):912–927.
16. Li X, Hu B, Shen J, Xu T, Retcliffe M. Mild depression detection of college students: an EEG-based solution with free viewing tasks. J Med Syst. 2015; 39(12).
17. Mensh BD, Werfel J, Seung HS. BCI competition 2003—data set ia: Combining gamma-band power with slow cortical potentials to improve single-trial classification of electroencephalographic signals. IEEE Trans Biomed Eng. 2004;51(6):1052–1056.
18. Nicolas-Alonso LF, Corralejo R, Gomez-Pilar J, Àlvarez D, Hornero R. Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces. Neurocomputing. 2015;159:186–196.
19. Panahi N, Shayesteh MG, Mihandoost S, Varghahan BZ. Recognition of different datasets using PCA, LDA, and various classifiers. In: 2011 5th International conference on application of information and communication technologies (AICT). Institute of Electrical and Electronics Engineers (IEEE); 2011.

20. Pfurtscheller G, Neuper C, Schlogl A, Lugger K. Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. IEEE Trans Rehab Eng. 1998;6(3):316–325.
21. Pfurtscheller G, Neuper Ch, Flotzinger D, Pregenzer M. EEG-based discrimination between imagination of right and left hand movement. Electroencephalogr Clin Neurophysiol. 1997;103(6):642–651.
22. Subasi A, Erċelebi E. Classification of EEG signals using neural network and logistic regression. Comput Methods Programs Biomed. 2005;78(2):87–99.
23. Sun S, Zhang C. Assessing features for electroencephalographic signal categorization. In: Proceedings. (ICASSP 05). IEEE international conference on acoustics, speech, and signal processing. Institute of Electrical and Electronics Engineers (IEEE); 2005.
24. Tang J, Deng C, Huang G-B. Extreme learning machine for multilayer perceptron. IEEE Trans Neural Netw Learn Syst. 2016;27(4):809–821.
25. Ting W, Guo-zheng Y, Bang-hua Y, Sun H. EEG feature extraction based on wavelet packet decomposition for brain computer interface. Measurement. 2008;41(6):618–625.
26. Vecchiato G, Susac A, Margeti S, De Vico Fallani F, Maglione AG, Supek S, Planinic M, Babiloni F. High-resolution EEG analysis of power spectral density maps and coherence networks in a proportional reasoning task. Brain Topogr. 2012;26(2):303–314.
27. Wang B, Jun L, Bai J, Peng L, Li G, Li Y. EEG recognition based on multiple types of information by using wavelet packet transform and neural networks. In: 2005 IEEE Engineering in medicine and biology 27th annual conference. Institute of Electrical and Electronics Engineers (IEEE); 2005.
28. Wolpaw JR, Birbaumer N, Heetderks WJ, McFarland DJ, Peckham PH, Schalk G, Donchin E, Quatrano LA, Robinson CJ, Vaughan TM. Brain-computer interface technology: a review of the first international meeting. IEEE Trans Rehab Eng. 2000;8(2):164–173.
29. Zandi AS, Javidan M, Dumont GA, Tafreshi R. Automated real-time epileptic seizure detection in scalp EEG recordings using an algorithm based on wavelet packet transform. IEEE Trans Biomed Eng. 2010;57(7):1639–1651.
30. Zhou W, Liu Y, Yuan Q, Li X. Epileptic seizure detection using lacunarity and bayesian linear discriminant analysis in intracranial EEG. IEEE Trans Biomed Eng. 2013;60(12):3375–3381.
31. Zong W, Huang G-B, Chen Y. Weighted extreme learning machine for imbalance learning. Neurocomputing. 2013;101:229–242.