This is a postprint version of the article published as:
Cao, Z, Yin, Z & Zhang, J. (2020). Recognition of cognitive load with a stacking network ensemble of denoising autoencoders and abstracted neurophysiological features. *Cognitive Neurodynamics, 15, 425-437*

Recognition of cognitive load with a stacking network ensemble of denoising autoencoders and abstracted neurophysiological features

Zixuan Cao $^{a, b}$, Zhong Yin $^{a, b,*}$, Jianhua Zhang ^{c}

a. *Engineering Research Center of Optical Instrument and System, Ministry of Education, Shanghai Key Lab of Modern Optical System, University of Shanghai for Science and Technology, Shanghai, 200093, P. R. China*

b. *School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, P. R. China*

c. *OsloMet Artificial Intelligence Lab, Department of Computer Science, Oslo Metropolitan University, Oslo, N-0130, Norway*

Abstract: The safety of human-machine systems can be indirectly evaluated based on operator's cognitive load levels at each temporal instant. However, relevant features of cognitive states are hidden behind in multiple sources of cortical neural responses. In this study, we developed a novel neural network ensemble, SE-SDAE, based on stacked denoising autoencoders (SDAEs) which identify different levels of cognitive load by electroencephalography (EEG) signals. To improve the generalization capability of the ensemble framework, a stacking-based approach is adopted to fuse the abstracted EEG features from activations of deep-structured hidden layers. In particular, we also combine multiple K-nearest neighbor and naive Bayesian classifiers with SDAEs to generate a heterogeneous classification committee to enhance ensemble's diversity. Finally, we validate the proposed SE-SDAE by comparing its performance with mainstream pattern classifiers for cognitive load evaluation to show its effectiveness.

Keywords: Human-machine system, cognitive load, neurophysiological signals, electroencephalography, neural network.

[*] Corresponding author. Tel.: +86 21 55271064. E-mail address: yinzhong@usst.edu.cn (Z. Yin).
Address: Jungong Road 516, Yangpu District, Shanghai 200093, P. R. China

1. Introduction

Human-machine (HM) systems widely exist in complex control environments for accomplishing predefined cognitive tasks (Habib et al., 2017). The HM systems have a capability to stabilize machine performance by incorporating actions of operator's supervision and decision-making (Yin et al., 2015) (Yin & Zhang, 2017). Different from machine agents who possess reliable functionalities, operators' performance can be instable or degraded because of intention distraction, mental fatigue and mental overload (Rusnock & Borghetti, 2018) (Parasuraman & Jiang, 2012). Such issues are major factors that cause many serious accidents originated by human operators. Numerous completed studies have shown that cognitive load is inversely related to the performance and operation quality of an operator in HM system (Lewis, 2019). The concept of the cognitive load is closely linked to cognition frameworks, operator emotions, and mental demand (Lewis, 2019). The distribution of the cognitive load is critical to maintain main performance under various HM task environment (Jaquess et al., 2018).

The cognitive load is vulnerable to many factors under safety critical human-machine interaction environment. There is currently no well-established definition of cognitive load (Young et al., 2015). In literature, the cognitive load can be considered as the amount of operator's mental resources taken up by operation requirements (Wilson & Eggemeier, 2020). When task demands increase and exceed general working capacity of operators, it causes excessive level of cognitive load and results in inability of information analysis and decision making of the operators (Fallahi et al., 2016) (Wilson, 2005). On the other hand, low cognitive load may cause the operators to become inefficient and inattention on current tasks (Ryu & Myung, 2005). To this end, an accurate and effective model is required to be designed for evaluating the cognitive load aiming at stabilizing the human performance within a proper range. It is particularly important for reducing risk and increasing operational safety of HM systems.

1.1 Related works of methods for cognitive load assessment

There are three main ways to assess cognitive load, i.e., subjective measures, task performance, and neurophysiological signals (Rutherford, 1987). Subjective measures are also known as subjective rating scales, of which the two most widely practical methods are Subjective Workload Assessment Technique (SWAT) and National Aeronautics and Space Administration-Task Load Index (NASA-TLX) (Fallahi et al., 2016). However, subjective measures lack objectivity and are limited by the low time resolution for data collection (Yin & Zhang, 2017). On the other hand, task performance measures are not suitable for implementing in those task environments where the performance parameters are implicit and cannot be directly collected (Hicks & Wierwille, 1979). Different from the two classical methods, the neurophysiological signals, such as electroencephalography (EEG), electrocardiogram (ECG), functional near infrared spectroscopy, and event-related potential (Di Stasi et al., 2013), are easy to be continuously acquired and processed in an online fashion. Among them, the EEG signals have high sensitivity, strong objectivity and easy implementation of task conditions and cognitive load. The EEG was also closely linked to levels of alertness and fatigue of operators (Makeig & Inlow, 1993) (Makeig & Jung, 1995) who were engaged in the task environments for nuclear power plants and transportation driving systems (Choi et al., 2018) (Borghini et al., 2014).

In this work, we employ the EEG signal as a unique indicator for assessing cognitive load. It should be noted that extensively reported works have applied pattern recognition methods to analyze the EEG signals. The machine learning based pattern classifiers can improve efficiency on modeling mappings between the EEG signals and human cognitive states. Among these works, (Wang et al., 2012) designed an EEG-based workload classifier with correct recognition rate of 80% via a hierarchical Bayesian model. Support Vector Machine (SVM) was used by (Ke et al., 2015) and they built a cross-task cognitive load identification model for n-back tasks. By incorporating the EEG features of power spectrum within 3–15 Hz frequency band, (Dornhege et al., 2007) identified mental tasks under different difficulties by using Linear Discriminant Analysis (LDA). In the work of (Vuckovic et al., 2002) found that the learning vector quantization neural network achieved the best classification performance among three different neural network frameworks for workload classification. All these papers validated effectiveness of statistical learning and neural network based approaches on the issue of the cognitive load recognition.

1.2 Motivation of the present study

Acquired EEG signals usually accompanied with various types of muscular noise that was caused by body movements, blinks, and respirations from users of brain computer interfacing devices under a HM system. Therefore, it is very crucial to extract noise-free EEG features by proper filters. However, useful frequency band for indicating cognitive load variation may be overlapped with those bands of noise. Another assessable approach is to extract critical information of the EEG by reconstructing signals at feature level. Under such framework, a Denoising AutoEncoder (DAE) which is a special type of Auto-Encoder (AE) with feedforward neural network structure is particularly suitable for feature filtering. It has a capability to learn higher-level feature abstractions from statistics of the EEG data with superimposed noise (Ting, 2015). The stability of a DAE can be further improved by adding different types of artificial noise into the training set when it has been implemented as a cognitive load classifier (Wang et al., 2016).

In the literature, the DAE is most widely used in the field of image processing. In the works of (Görgel & Simsek, 2019) and (Lan et al., 2019) applied an improved model of DAE to cope with face recognition and hyperspectral image classification problems. In the works of (D. Lee et al., 2018) and (Fang et al., 2018) exploit DAE's noise reduction capability for image filtering. A DAE based data-driven model has been also applied for fault diagnosis in processing or mechanical control systems. For instance, (Meng et al., 2018) designed an enhancement DAE based on the fault diagnosis of rolling bearing. In the works of (Fu et al., 2019), (Xu et al., 2018) and (Yu, 2019) designed modified DAE models for fault detection issues. In addition, DAEs with hierarchical network structure have shown outstanding performance in a variety of machine learning tasks in EEG data processing (Lee et al., 2020).

Although the pattern recognition methods can achieve acceptable cognitive load recognition accuracy, classical shallow machine learning methods have a difficulty in mining hidden information associated with operator's cognitive state variables. Therefore, we attempt to implement Stacked Denoising AutoEncoder (SDAE) to build a deep-learning based workload classifier. Unlike neural network framework with single hidden layer, a deep learning model possesses hierarchical feature abstraction structure (Zhou, 2016). Such model increases number of hidden layers in feedforward path. By feeding noisy input data, a layer-wise training scheme was performed with the functionality for unsupervised feature denoising (Wang et al., 2016).

In recent years, SDAE is widely used in face recognition (Zhang et al., 2016) and industrial fault detection problem (Wang et al., 2016), many of which have been developed and commercialized successfully. In this study, we employed SDAE as a backbone to develop an accurate cognitive load estimator. In particular, ensemble learning principle has been adopted to reduce uncertainty in EEG features among multiple subjects. To improve the generalization capability and avoid potential overfitting, a stacking-based ensemble learning framework is utilized to fuse abstracted EEG features from activations of deep-structured hidden layers. For those wake learners, we combine multiple K-nearest neighbor and naive Bayesian classifiers with DAEs to generate a heterogeneous classification committee to improve ensemble's diversity. In the end, we compare the proposed Stacking-based Ensemble of SDAE (SE-SDAE) with several classical cognitive load estimators to validate its effectiveness.

The rest of the paper is organized as follows. In Section 2, the EEG database used for evaluation the performance of the workload classifier has been described. The detailed methodology of the SE-SDAE is given in Section 3. Section 4 shows the results of the performance of the cognitive load classification. The discussion and conclusion has been summarized in Section 5.

2. Experimental data

The data used in this experiment has been collected on Automation-enhanced Cabin Air Management System (Auto-CAMS) in previous studies (Zhang et al., 2015). Auto-CAMS is a simulated software system that fulfills the needs of complex HM missions by providing an air quality control task in an aircraft cabin. The EEG signals of the experimental participants were collected by the Nihon Kohden biomedical signal processor and displayed in real0time by Neurofax software. The data acquisition experiments are briefly described as follows. It should be noted that the details of the Auto-CAMS software are omitted and can be found in (Yin & Zhang, 2017).

2.1 Experimental participants and setup

Eight on-campus postgraduate students (male, aged 21-24 years) participated in the experiment as volunteers. The experiments were carried out in accordance with the guidelines issued by the Ethical Committee of East China University of Science and Technology (ECUST). All participants gave written informed consent in accordance with the Declaration of Helsinki. Each participant completed Auto-CAMS operation training prior to the experiment and was anonymized consecutively with a label from S1 to S8.

The Auto-CAMS platform controls air quality through four subsystems: oxygen concentration, carbon dioxide concentration, pressure and temperature. Participant's task is to manually adjust the parameters of the failed subsystem and stabilize it within the target range. The system can change the complexity of the task by manipulating the Number Of Failed Subsystems (NOFS) to meet different cognitive load requirements. A greater NOFS value corresponds to a higher task complexity and cognitive load level.

Participants are required to conduct two experimental sessions separately for two different days, each of which is divided into eight phases and lasts for 100 minutes. The first and eighth stages correspond to the condition of NOFS=0 and last for 5 minutes. The remaining six stages last for 15 minutes and correspond to the conditions of NOFS = 1, 3, 4, 4, 3, 1, respectively. The stages of the NOFS=0 condition do not require participants to operate and they can be used to verify whether the cognitive load level is restored or not after accomplishing an experimental session. Note that the condition of NOFS=2 is omitted to prevent a situation in which excessive experimental time leads to exhaustion of participants and affection of their cognitive load levels. This multi-day scheme with cycle task complexity comprehensively capture the cognitive state information and facilitate a fair comparison of the classifier performance on the EEG data.

2.2 Neurophysiological data preprocessing

Eleven electrodes on the scalp of the participants are used to collect the EEG data and placed at the positions of F3, F4, Fz, C3, C4, Cz, P3, P4, Pz, O1 and O2 that are specified by the 10-20 international system. Each phase of the experiment generates a dataset, which contains 450,000 data points corresponding to 11 channels with a sampling frequency of 500 Hz. Since each participant conducted two sessions of eight-phase experiments, the total number of datasets is 16.

For each raw EEG dataset, we first eliminate the high frequency muscle noise using the 4th order Butterworth IIR filter (with low-pass cutoff frequency of 40 Hz). In the second step, we use Independent Component Analysis (ICA) to separate the EEG data into Independent Components (ICs). Then, we determine the IC corresponding to the maximum r_e value as the ocular artifact source and set all its values to zeros, where we define the r_e as the ratio between the power of the delta band (1–4 Hz) and the summation of all frequential bands (1–40 Hz). To achieve sufficient temporal resolution, the EEG data is segmented with each of two seconds. Two seconds of the filtered EEG data are illustrated in Fig. 1.

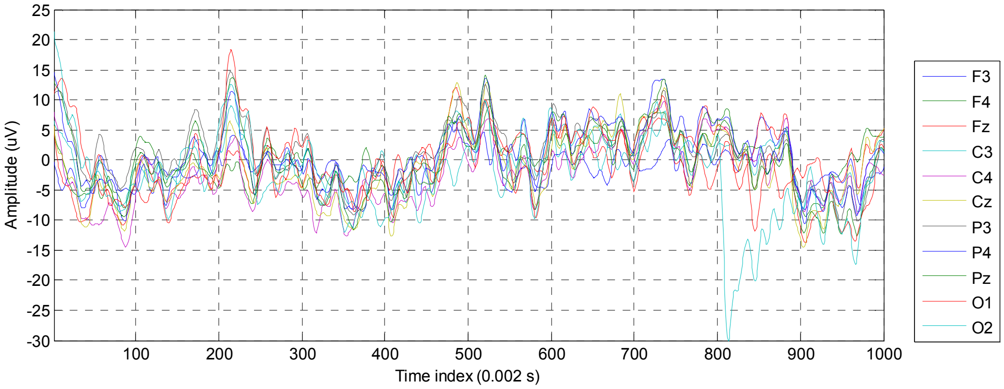

Fig. 1. Filtered EEG signals in a segment on 11 channels under the sampling frequency of 500 Hz.

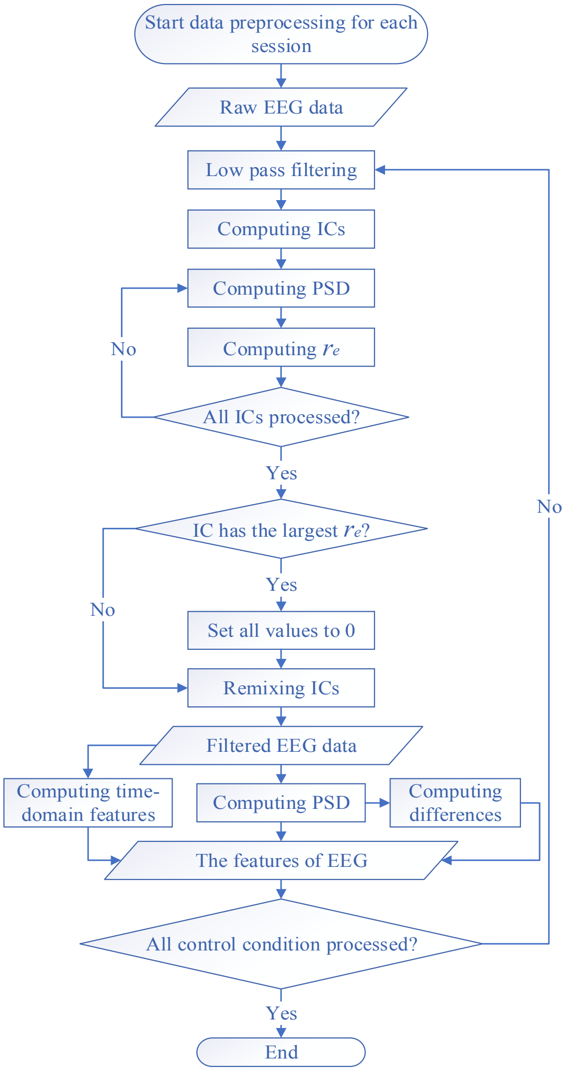

Fig. 2. Flow chart for data preprocessing

Then, the Fast Fourier Transform (FFT) with a frequency resolution of 0.5 Hz is used to calculate the PSD features of each segment. The PSD features were extracted within four frequency bands of each channel, i.e., theta (4–8 Hz), alpha (8–13 Hz), beta (14–30 Hz) and gamma (30–40 Hz). We obtained a total of 44 frequency domain features for the single-channel condition. Additional 16 features can be derived by calculating the power differences between the left and right hemispheres of the scalp, i.e., F4-F3, P4-P3, C4-C3, and O2-O1. Finally, with the time-domain features of mean, variance, zero crossing rate, Shannon entropy, spectral entropy, kurtosis and skewness calculated by 11 channels, a total of 137 features were obtained for each EEG segment. All the above EEG data preprocessing steps are shown in Fig. 2.

Note that the EEG features of the second, fourth, fifth and seventh phases of each experimental session are selected for training and testing the proposed cognitive load classifier. The EEG feature sets of the four phases of a session and a subject are merged to form a feature matrix of 1800×137, wherein the 900 data points corresponding to the second and seventh phases are determined as low cognitive-load level and the remaining data points are determined as high cognitive-load level. Finally, 16 EEG feature sets are prepared for further analysis.

3. Methodology

In this section, we first introduce the algorithms used for building the weak learners in the ensemble model and then describe the details of the proposed SE-SDAE approach.

3.1 Methods used for building weak learners of the classifier committee

3.1.1 Stacked Denoising AutoEncoder

The reconstruction of the data in DAE is achieved through its basic component Autoencoder (AE). The architecture of the AE is a shallow neural network with single hidden layer and is known as a three-layer multi-layer perceptron. By training the equivalent transformation of the input and target output, the hidden layer could represent the input features in a different abstraction space. In this study, the dimensionality of the hidden layer of the AE is smaller than the dimensionality of the EEG features. The reduced dimensionality expression of input can be obtained through an DAE model. The transformation between each two layers in AE is a linear transformation combined with a mapping of nonlinear activation function. The output of the activation in the hidden layer can be computed as follows,

$$\mathbf{x}_h = f(\mathbf{W}\mathbf{x}+\boldsymbol{\theta}) = \frac{1}{1+\exp(-\mathbf{W}\mathbf{x}-\boldsymbol{\theta})}. \tag{1}$$

In the equation, $\mathbf{x}\in R^D$ is defined as the input EEG feature or feature abstraction fed to the AE and $\mathbf{x}_h \in R^d$ is the vector of activation values in the hidden layer. The bias vector and weight matrix correspond to $\boldsymbol{\theta}\in R^d$ and $\mathbf{W}\in R^{d\times D}$, respectively. The terms D and d represent the dimensionalities of the input features and hidden feature abstractions.

Then, the hidden layer to the output layer mapping can be defined by the function expressed as,

$$\mathbf{x}_o = f(\mathbf{W}'\mathbf{x}_h+\boldsymbol{\theta}') = \frac{1}{1+\exp\left[-\frac{\mathbf{W}'}{1+\exp(-\mathbf{W}\mathbf{x}-\boldsymbol{\theta})}-\boldsymbol{\theta}'\right]}. \tag{2}$$

In the equation, the term $\mathbf{x}_o \in R^D$ is the output of the AE and it denotes the reconstruction of the input EEG feature. Here, the tied weights are employed with $\mathbf{W}' = \mathbf{W}^T \in R^{D\times d}$. The loss of the input and output is represented by the square error cost function,

$$L(\mathbf{x},\mathbf{x}_o) = \left\|\mathbf{x}-\mathbf{x}_o\right\|_2^2. \tag{3}$$

The Back Propagation (BP) algorithm can be utilized to minimize the above cost function and determines the trained values of $\mathbf{W}$, $\boldsymbol{\theta}$, and $\boldsymbol{\theta}'$. The BP algorithm could update the weights of the AE by calculating the partial derivative of the error function with respect to the parameters in each neuron until the preset precision or maximum number of iterations is reached. By defining the optimized parameters as $\mathbf{W}^*$, $\boldsymbol{\theta}^*$, and $\boldsymbol{\theta}'^*$, the trained parameter set of the AE can be represented by the following equations with N denoting the number of training data points.

$$\begin{aligned}\{\mathbf{W}^*,\boldsymbol{\theta}^*,\boldsymbol{\theta}'^*\} &= \arg\min \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x},\mathbf{x}_o) \\ &= \arg\min \frac{1}{N}\sum_{i=1}^{N} L\left(\mathbf{x}, f(\mathbf{W}' f(\mathbf{W}\mathbf{x}+\boldsymbol{\theta})+\boldsymbol{\theta}')\right)\end{aligned} \tag{4}$$

In order to improve the ability of the AE to reduce the noise in multidimensional features, we randomly set the input $\mathbf{x}^{(i)}$ to zero based on a uniform distribution to cause some interference before performing the training stage. This procedure can be denoted as follows,

$$\bar{\mathbf{x}}^{(i)} \sim p\left(\bar{\mathbf{x}}_0^{(i)} \middle| \mathbf{x}_0^{(i)}\right), \tag{5}$$

where the uniform probability density distribution of elicit $\bar{\mathbf{x}}^{(i)}$ for $i = 1, 2, ..., D$ is defined by p.

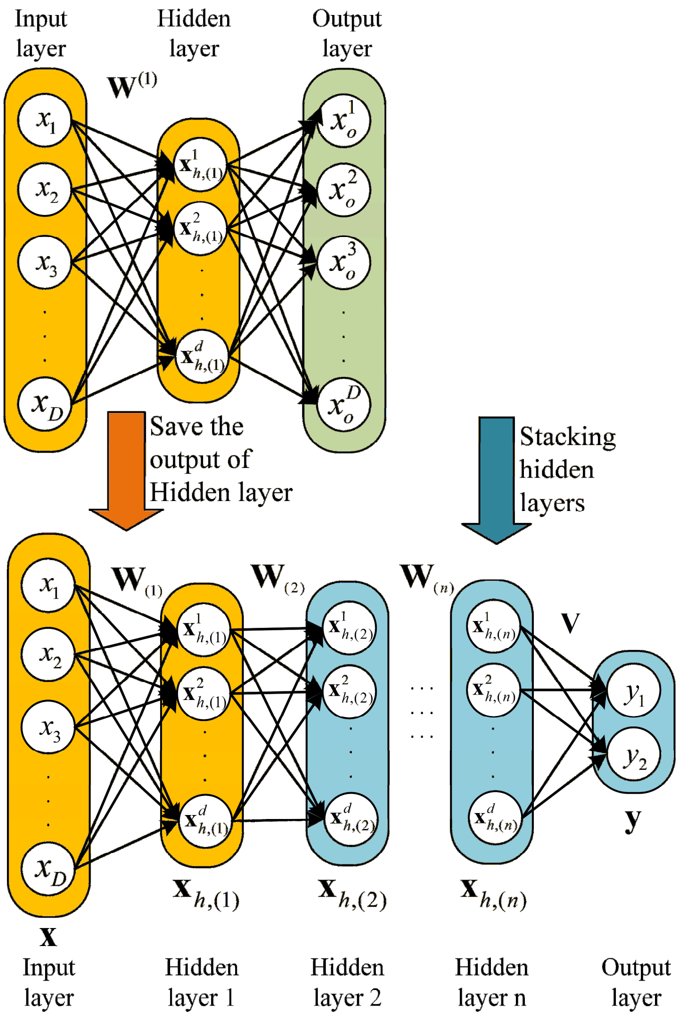

Fig. 3. Architecture of the SDAE used for EEG modeling.

By implementation of Eqn. (5) for the input features of the AE, a DAE network is derived. The essential functionality of DAE is to extract the noise-free, low-dimensional embeddings of the input data. Moreover, multiple DAEs could be connected together to build a SDAE. That is, after training a DAE, the output of the hidden layer is used as input to train another DAE network. The outputs of the high-level DAE could abstract compact feature representations of the original inputs. Based on such stacked, hierarchical architecture, the output of the n^{th} hidden layer of a SDAE can be expressed as,

$$\mathbf{x}_{h,(n)} = f\left(\mathbf{W}_{(n)} \ldots f(\mathbf{W}_{(2)} f(\mathbf{W}_{(1)} x_{(0)} + \boldsymbol{\theta}_{(1)}) + \boldsymbol{\theta}_{(2)}) \ldots + \boldsymbol{\theta}_{(n)}\right). \tag{6}$$

Finally, a top layer containing two nodes is added to form a network that can indicate binary cognitive load levels. To facilitate supervised machine learning using BP algorithm, the output of the SDAE used as a weak classifier can be defined as,

$$\mathbf{y} = f\left(\mathbf{V}\mathbf{x}_{h,(n)} + \mathbf{v}\right) = s\left(\mathbf{x}_{(0)}\right). \tag{7}$$

In the equation, the value of $\mathbf{y} = [1\ 0]^T$ and $\mathbf{y} = [0\ 1]^T$ represent the low and high cognitive load levels, respectively. Given $\mathbf{x}_{h,(n)} \in R^{d_n}$, the term $\mathbf{V} \in R^{2 \times d_n}$ is defined as the output weight matrix and $\boldsymbol{v} \in R^2$ is the bias vector. The architecture of a SDAE is shown in Fig. 3.

3.1.2 K-nearest Neighbors weak learner

The K-nearest Neighbors (KNN) classifier first finds k training samples that are closest to the testing sample. Then it predicts the category of the testing samples based on their information of class labels of these neighbor instances. The testing sample shown in Fig. 4 is classified into the category ω3 according to its neighbor instances, where k = 7.

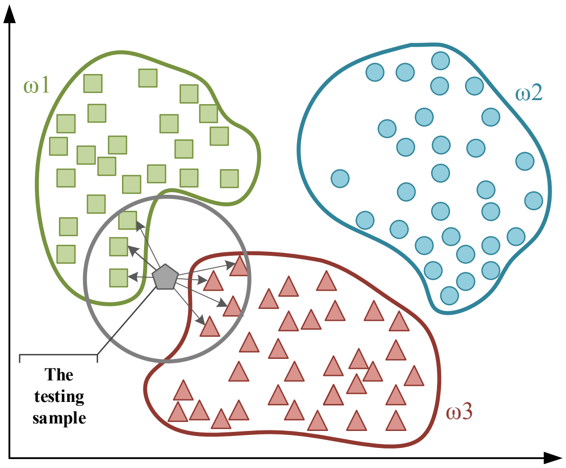

Fig. 4. Schematic of K-nearest Neighbors algorithm.

The closeness is evaluated by using the Euclidean distance between each two samples as follows,

$$d(\mathbf{X}^{(i)},\mathbf{X}^{(j)})=\sqrt{\sum_{p=1}^{D}\left|x_p^{(i)}-x_p^{(j)}\right|^2}. \tag{8}$$

In the equation, $\mathbf{X}=(\mathbf{x}^{(1)},\mathbf{x}^{(2)},...,\mathbf{x}^{(n)})$ represents the training dataset with n instances defined. The k data points that are closest to the testing sample can be obtained by sorting the $d(\mathbf{X}^{(i)},\mathbf{X}^{(j)})$ from low to high. The target class labels that defines cognitive load levels can be used to estimate the class of the testing sample by majority voting. The choice of k value is critical to the generalization capability of the model. To determine the optimal k, we compute the training error separately with different k adopted and find the parameter corresponding to the minimum error.

3.1.3 Naive Bayesian classifier

The principle of Naive Bayesian classifier adopted is the Bayesian theorem in probability theory. Its role is to describe the relationship between two conditional probabilities as shown as follows,

$$P(B\mid A)=\frac{P(A\mid B)\,P(B)}{P(A)}. \tag{9}$$

In the equation, the term $P(B|A)$ indicates the probability of event B when event A has occurred while $P(A|B)$ denotes an opposite case. Naive Bayesian model assumes that all the EEG features are independent of each other. By defining the features and cognitive load levels as $\mathbf{x}=\{x_1,x_2,...,x_D\}$ and $\mathbf{y}=\{y_1,y_2\}$, the class conditional probability density can be described as follows,

$$p\ (\mathbf{x}\mid y_n)=\prod_{i=1}^{D}p\,(x_i\mid y_n). \tag{10}$$

In the equation, D denotes the EEG feature dimensionality and n is the cognitive load level.

Based on Eqn. (9), the conditional probability of each class can be arranged as follows,

$$p\ (y_n \mid \mathbf{x}) = \frac{p\ (y_n) \prod_{i=1}^{D} p\,(x_i \mid y_n)}{p\ (\mathbf{x})}. \tag{11}$$

In the equation, $p\ (y_n)$ and $p\ (y_n \mid \mathbf{x})$ are the prior probability and posterior probability of class y_n. The difference between them is that the prior probability does not need to consider factors of current observations from instances while the calculation of the posterior probability predicts an event based on the effect of the training instances.

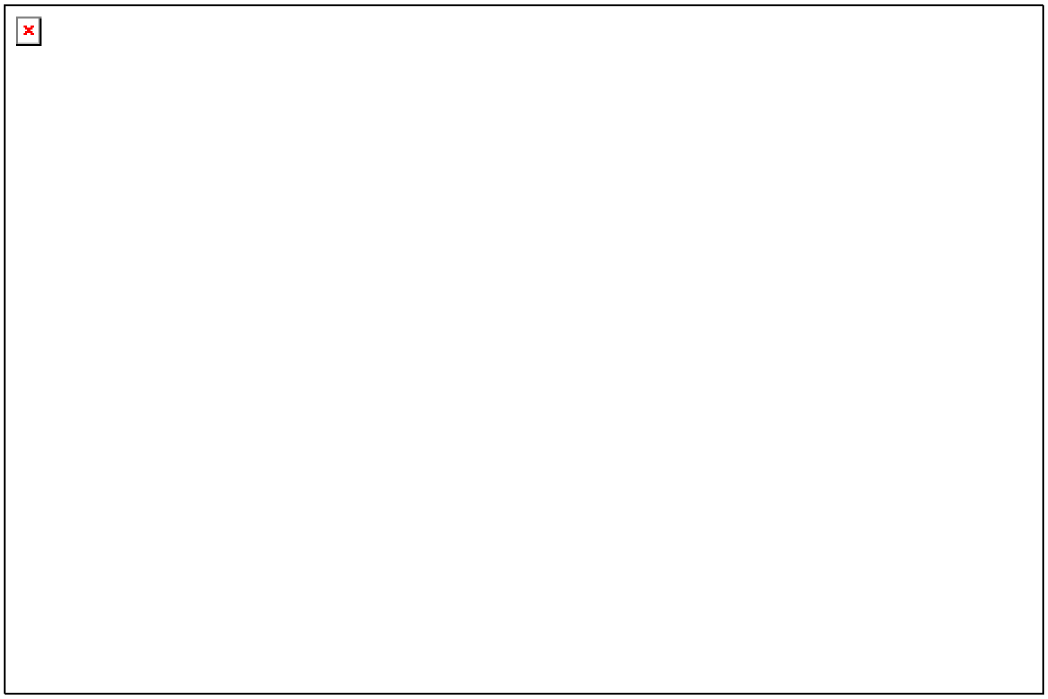

Fig. 5. Architecture of the SE-SDAE based on EEG modeling for the cognitive load recognition.

According to Eqn. (11), we can get the probability that a sample belongs to cognitive load level y_n,

$$p\ (y_n \mid x_1, x_2, \ldots, x_D) = \frac{p\ (y_n\ ) \prod_{i=1}^{D} p\,(x_i \mid y_n)}{\prod_{i=1}^{D} p\,(x_i\ )}. \tag{12}$$

For a sample to be tested, we calculate the posterior probability corresponding to the different classification labels of cognitive load. The label with the largest posterior probability is the predicted category of the testing sample of multidimensional EEG features.

3.2 Stacking-based ensemble of SDAE

The primitive methodologies of ensemble learning can be summarized into three basic categories, i.e., bagging, boosting and stacking. Bagging and boosting employ parallel and serial calculation methods for training the classifier committee, respectively. On the other hand, the core idea of stacking is to implement *k*-fold cross-validation technique to learn different member classifiers. The merit of cross-certification is to effectively avoid over-fitting problem caused by limited size of training samples. In this study, we applied five-fold cross validation to build our SE-SDAE model for cognitive load recognition.

The architecture of the proposed SE-SDAE is illustrated in Fig. 5. The training of the SE-SDAE model mainly includes the following steps:

In the first step, the training set $\mathbf{A} = \{\mathbf{X}_{tr}, \mathbf{Y}_{tr}\}$ is divided into *J*, non-overlapped subsets with equal size according to the temporal order of samples. Select one of them as a validating subset, the rest are used to learning the SDAE model. The subsets of the EEG features are denoted as $\{\mathbf{A}_{(1)}, \mathbf{A}_{(2)}, ..., \mathbf{A}_{(J)}\}$. The value of *J* is predetermined as five.

Then, the five-fold cross validation is carried out. In total $J-1$ subsets are used to train SDAE, the output can be computed as,

$$\mathbf{y} = \tilde{f}^{(j)}(\mathbf{x}) = f\left(\mathbf{V}^*\mathbf{x}^*_{h,(M)} + \mathbf{v}^*\right), \tag{13}$$

$$\mathbf{x}^*_{h,(M)} = f\left(\mathbf{W}^*_{(M)} ... f(\mathbf{W}^*_{(2)} f(\mathbf{W}^*_{(1)}\mathbf{x} + \boldsymbol{\theta}^*_{(1)}) + \boldsymbol{\theta}^*_{(2)}) ... + \boldsymbol{\theta}^*_{(M)}\right). \tag{14}$$

Based on the learned model $\tilde{f}^{(j)}$, cognitive load levels of the validating subset *j* and the testing set are separately predicted as,

$$\mathbf{Z}^{(j)}_s = \tilde{f}^{(j)}(\mathbf{X}_{tr(j)}), \tag{15}$$

$$\mathbf{G}^{(j)}_s = \tilde{f}^{(j)}(\mathbf{X}_{te}). \tag{16}$$

In the equation, $\mathbf{X}_{tr(j)}$ and $\mathbf{X}_{te}$ are the feature vectors of the validating subset j and the testing set, respectively. The terms $\mathbf{Z}_s^{(j)}$ and $\mathbf{G}_s^{(j)}$ are their predicted values of the cognitive load. Since the value of J is set to five, the above steps are repeated five iterations in total. That is, a set of the predicted values on the validating sets and the testing set are derived as $\{\mathbf{Z}_s^{(1)}, \mathbf{Z}_s^{(2)}, ..., \mathbf{Z}_s^{(J)}\}$ and $\{\mathbf{G}_s^{(1)}, \mathbf{G}_s^{(2)}, ..., \mathbf{G}_s^{(\mathrm{j})}\}$, respectively. Finally, the predicted values $\mathbf{Z}_s^{(j)}$ are concatenated to generate a new feature matrix $\mathbf{A}^{'}$. At the same time, the mean of the five-dimensional predicted values of the testing EEG feature set is used to build another feature matrix $\mathbf{B}^{'}$.

Table 1. Pseudo codes for learning a cognitive load classifier based on the SE-SDAE

```
**Algorithm:**  **SE-SDAE**
**Inputs:**     Training set, $\mathbf{A} = \{\mathbf{X}_{tr}, \mathbf{Y}_{tr}\}$
                Testing set, $\mathbf{B} = \{\mathbf{X}_{te}, \mathbf{Y}_{te}\}$
                Learning algorithms excluding SDAE, $\ell_1, \ell_2, ..., \ell_T$
                Number of SDAE's hidden layers, $M$
                Dimensionality of the EEG features, $D$
                Number of training data points, $N$
                Number of base classifiers, $T$
**Output:**     Ensemble classifier, $h_s(\tilde{f}^{(1)}, \tilde{f}^{(2)}, ..., \tilde{f}^{(J)}, \hat{f}_{(1)}^{(1)}, \hat{f}_{(1)}^{(2)}, ..., \hat{f}_{(1)}^{(J)}, ..., \hat{f}_{(T)}^{(J)})$

Randomly assign network parameters
**For** $j = 1 : J$, $J = 5$
    $\mathbf{x}_{h,(0)} \leftarrow \mathbf{X}_{tr(j)}$
    **For** $n = 1 : M$
        **For** $i = 1 : D$
            $\overline{\mathbf{x}}_{(n)}^{(i)} \sim p\left(\overline{\mathbf{x}}_{0,(n)}^{(i)} \middle| \mathbf{x}_{0,(n)}^{(i)}\right)$
        $\mathbf{x}_{h,(n)} \leftarrow f(\mathbf{W}_{(n)}\mathbf{x}_{h,(n\text{-}1)} + \boldsymbol{\theta}_{(n)})$
        $\{\mathbf{W}_{(n)}^{*}, \boldsymbol{\theta}_{(n)}^{*}\} \leftarrow \arg\min 1/N \cdot \sum^{N} L\left(\mathbf{x}_{h,(n\text{-}1)}, \mathbf{x}_{h,(n+1)}\right)$
    $\mathbf{y}^{(j)} = f\left(\mathbf{V}\mathbf{x}_{h,(\mathrm{M})} + \mathbf{v}\right)$
    **For** $n = 1 : M$
        Compute $\{\mathbf{W}_{(n)}^{*}, \boldsymbol{\theta}_{(n)}^{*}, \mathbf{V}^{*}, \mathbf{v}^{*}\}$ via Eqn.(4)
    $\mathbf{Z}_s^{(j)} \leftarrow \tilde{f}^{(j)}(\mathbf{X}_{tr(j)})$
    $\mathbf{G}_s^{(j)} \leftarrow \tilde{f}^{(j)}(\mathbf{X}_{te})$
$\mathbf{A}^{'} \leftarrow \left\lceil \mathbf{Z}_s^{(1)}; \mathbf{Z}_s^{(2)}; ...; \mathbf{Z}_s^{(J)} \right\rceil$
$\mathbf{B}^{'} \leftarrow \sum_{i=1}^{J} \mathbf{G}_s^{(j)}$
**For** $t = 1 : T$
    **For** $j = 1 : J$
```

```
            Build base leaner $\hat{f}_{(t)}^{(j)} \leftarrow \ell_{(t)}(\mathbf{A}_{(j)})$
            $\hat{\mathbf{Z}}_{s}^{(j)} \leftarrow \hat{f}_{(t)}^{(j)}(\mathbf{X}_{tr(j)})$
            $\hat{\mathbf{G}}_{s}^{(j)} \leftarrow \hat{f}_{(t)}^{(j)}(\mathbf{X}_{te})$
        $\hat{\mathbf{A}}_{(t)}^{'} \leftarrow \left\lceil \hat{\mathbf{Z}}_{(t)}^{(1)}; \hat{\mathbf{Z}}_{(t)}^{(2)}; \ldots; \hat{\mathbf{Z}}_{(t)}^{(J)} \right\rceil$
        $\hat{\mathbf{B}}_{(t)}^{'} \leftarrow \sum_{i=1}^{J} \hat{\mathbf{G}}_{s}^{(j)}$
        $\mathbf{A}^{'} \leftarrow \mathbf{A}^{'} \cup \hat{\mathbf{A}}_{(t)}^{'}$
        $\mathbf{B}^{'} \leftarrow \mathbf{B}^{'} \cup \hat{\mathbf{B}}_{(t)}^{'}$
    $\mathbf{A}^{'} = \mathbf{A}^{'} \cup \mathbf{Y}_{tr}$
    Learning secondary classifier, $h_s \leftarrow \ell_s(\mathbf{A}^{'})$
    Return model $h_s(\tilde{f}^{(1)}, \tilde{f}^{(2)}, \ldots, \tilde{f}^{(J)}, \hat{f}_{(1)}^{(1)}, \hat{f}_{(1)}^{(2)}, \ldots, \hat{f}_{(1)}^{(J)}, \ldots, \hat{f}_{(T)}^{(J)})$
```

After the above computations are completed, we build T additional classifiers $\ell_1, \ell_2, \ldots, \ell_T$ with heterogeneous structure excluding the SDAE as the base classifiers. During each iteration, the training, validating subsets and the testing set remain unchanged. Then, the T paired feature matrices generated by the heterogeneous base classifiers are connected in parallel with $\mathbf{A}^{'}$ and $\mathbf{B}^{'}$. The target cognitive load labels are added for supervised learning. The feature matrices $\mathbf{A}^{'}$ and $\mathbf{B}^{'}$ are considered as input features during training and testing of the whole SE-SDAE, respectively. The functionality of the two matrices is to indicate cognitive load levels for the secondary classifier. The pseudo codes of all the above steps is shown in Table 1.

Stacking essentially employed a multilayered structure that is similar to the neural network with each node replaced by a base learner or secondary classifier. The effectiveness of stacking mainly comes from feature abstraction from different training subsets based on cross validation. Different base classifiers express the representation of different features through a heterogeneity classification committee. Therefore, the novelty of the proposed SE-SDAE lies not only in the effect of multi-layer stacking but also the enhanced learning ability from different types of the training algorithms. It implies the SE-SDAE has a potential to achieve superior performance. However, with the increase of the number of layers in the member SDAE classifier, SE-SDAE faces a serious over-fitting risk. It is the reason that only two hidden layers has been adopted.

4. Results

In this study, eighty percent of the EEG data is used to build a training set with a size of 23040×137 and the rest is used for testing the classifier's performance 5760×137. To identify the optimal structure of the SDAE, we first examine the number of hidden layers in Fig. 6. In the figure, SEN, SPE, NPV, PRE, FH, FL, and ACC denote the average sensitivity, specificity, negative predicting value, precision, F1-score of high class, F1-score of low class, and accuracy, respectively. For the SDAE classifier, the batch size and the learning rate are selected as 32 and 1, respectively. It is shown that the two hidden layers could achieve sufficient generalization capability.

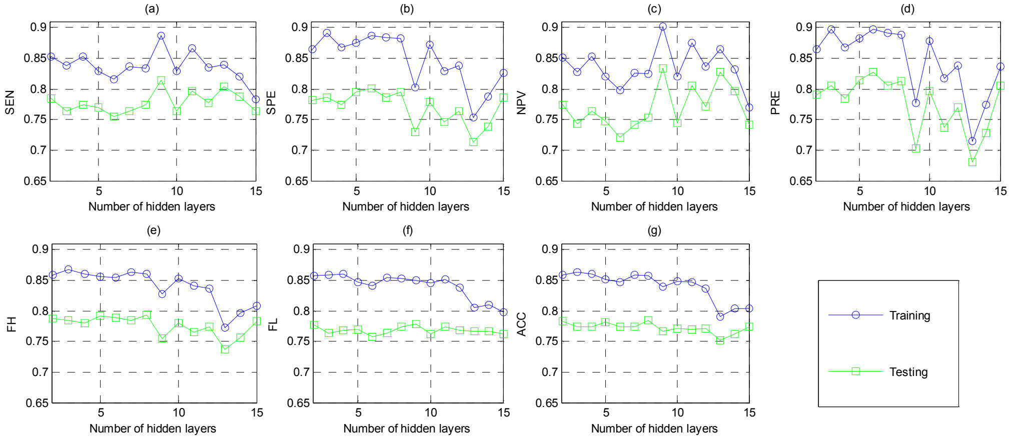

Fig. 6. Classification performance of the SDAE with different number of hidden layers.

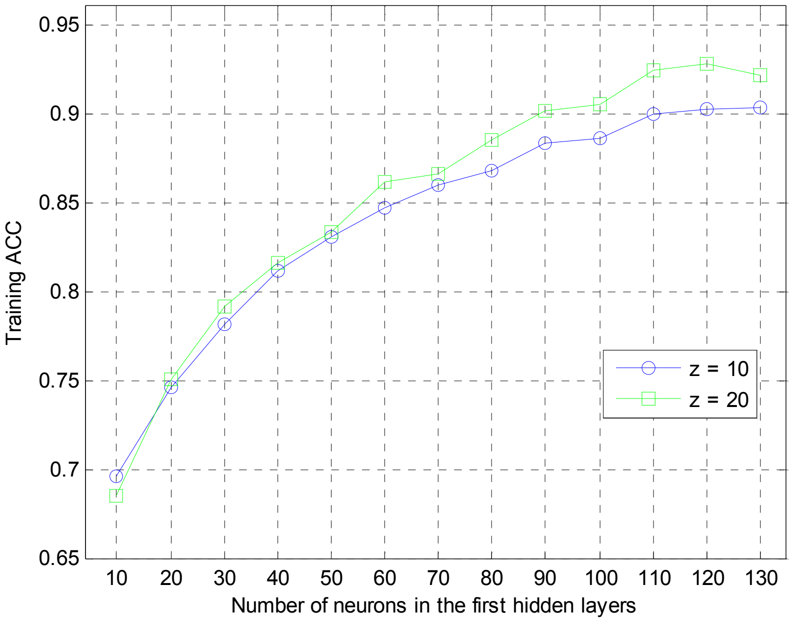

Fig. 7. Training performance of different hidden layer neurons in SDAE. The value of *z* denotes the number of hidden neurons in the second hidden layer.

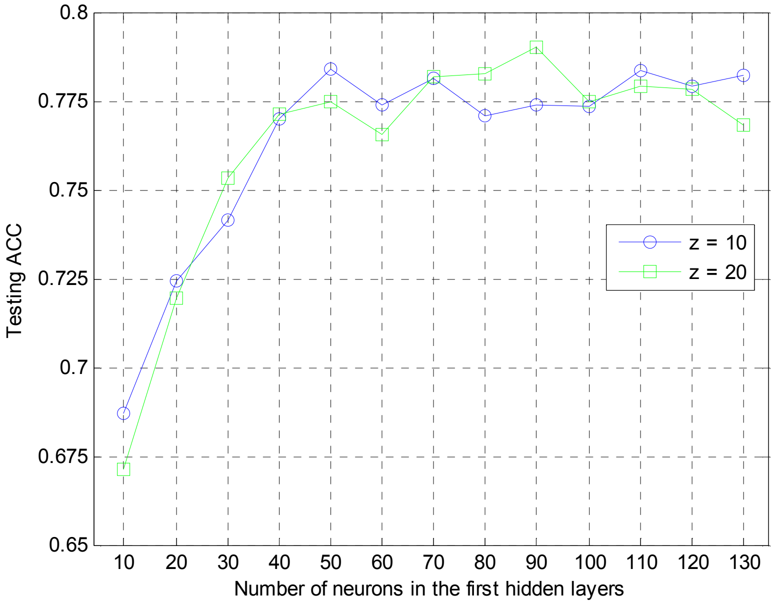

Fig. 8. Testing performance of different hidden layer neurons in SDAE. The value of *z* denotes the number of hidden neurons in the second hidden layer.

By varying the number of first hidden layer neurons of the SDAE, we have also investigated different training ACC and testing ACC to determine the optimal hidden neurons in the SE-SDAE. The corresponding results are shown in Figs. 7 and 8. The value of z represents the number of second hidden layer neurons, where two different values have been examined.

From the figure, it is shown that both the training and testing ACC improve along with the increase of the number of hidden neurons. However, when such value is greater than 40, the testing performance remains stable. The reason behind is that the potential overfitting of the deep neural network for coping with high dimensional EEG features. It is noted that the SDAE has the best training ACC when the number of neurons in the first hidden layer is selected at 120 with z = 20. Therefore, we adopted the value as the optimal hyper-parameter of the SDAE network for the cognitive load classification. The training ACC and test ACC corresponding to this parameter are 0.9280 and 0.7783, respectively.

Table 2. Detail classification performance of the cognitive load by the SE-SDAE and the classical SDAE.

Classification Metrics	*SDAE*		*One-dimensional stacking of the SE-SDAE*		*Two-dimensional stacking of the SE-SDAE*	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*
SEN	0.9169	0.7722	0.7753	0.8207	0.7742	**0.8212**
SPE	0.9395	0.7851	0.7538	0.8023	0.7590	**0.8069**
NPV	0.9156	0.7549	0.7815	**0.8343**	0.7776	0.8336
PRE	0.9405	0.8009	0.7471	0.7867	0.7554	**0.7931**
FH	0.9286	0.7863	0.7609	0.8033	0.7647	**0.8069**
FL	0.9274	0.7697	0.7674	0.8180	0.7682	**0.8201**
ACC	0.9280	0.7783	0.7642	0.8109	0.7664	**0.8137**

Note: The optimal testing performance is marked in bold.

Table 3. Performance comparison between the SE-SDAE and classical cognitive load classifiers.

Cognitive classifiers	*Training accuracy*	*Testing accuracy*
SDAE	0.9280	0.7783
KNN	0.7916	0.7658
Naive Bayes	0.6978	0.6977
Logistic regression	0.7812	0.7828

Cognitive classifiers	*Training accuracy*	*Testing accuracy*
Extreme learning machine	0.8256	0.7724
Discriminant analysis classifier	0.7694	0.7677
One-dimensional stacking of the SE-SDAE	0.7642	0.8109
Two-dimensional stacking of the SE-SDAE	0.7664	**0.8137**

Note: The optimal testing performance is marked in bold.

After we selected the SDAE as the base classifier, the KNN algorithm is used as the secondary classifier in based on the stacking scheme to training the SE-SDAE model (denoted as one-dimensional SE-SDAE). When comparing the model performance of the SE-SDAE with the classical SDAE, KNN, Naive Bayesian model, logistic regression, extreme learning machine, and discriminant analysis classifier, we found that the testing accuracy of the one-dimensional model of the SE-SDAE has increased. The corresponding results are listed in Table 2.

To further improve the performance of the SE-SDAE, we have tried several different architectures of the stacking network. For instance, we have used SDAE as the secondary classifier and KNN as the base classifier. We also tried to use Naive Bayesian model as the base classifier and KNN as the second order classifier. But these attempts have produced a negative results. In the end, we designed a two-dimensional integration model with KNN as the secondary classifier was while the SDAE and Naive Bayesian classifiers are simultaneously used as the base learners. The last trained model achieved the desired results, as shown in Table 3. The potential reason that the two-dimensional model achieves a higher accuracy is the improvement of the diversity of the ensemble classification committee.

Table 4. Performance comparison of different base classifiers in the SE-SDAE.

Index of the SDAE model	*Training accuracy*	*Testing accuracy*	*Index of the NB*	*Training accuracy*	*Testing accuracy*
1	0.9192	0.7987	1	0.6988	0.6989
2	0.9268	0.7945	2	0.7051	0.7050
3	0.9283	0.7967	3	0.6974	0.6892
4	0.9209	0.7988	4	0.6948	0.6929
5	0.9218	0.7979	5	0.6905	0.6992

Note that the ensemble model of the SE-SDAE requires to train its base classifier on five iterations because of the implementation of the cross validation technique. In order to observe the performance difference of the SE-SDAE and each member classifiers. We preserved the experimental precision of all five training iterations of the base classifiers for the naive Bayesian (NB) model and SDAE in Table 4. The performance improvement of the SE-SDAE can be found by comparing the results of Tables 2 and 3. It is noted that each base SDAE can achieve significantly higher performance than that of the NB classifier. It partially validates the effectiveness to employ the SDAE and the base learner.

In the end, we list the performance of the proposed SE-SDAE classifier of each subject for binary cognitive load classification in Table 5. In general, the training accuracy is significantly higher than the testing accuracy. For specific subject, e.g., Subject #4, relatively low performance is observed. The results indicate individual difference may severely impair the generalization capability of the ensemble model. The testing classification accuracy on Subject #1, #2, #6, and #7 achieves satisfactory performance. The observation indicates the neurophysiological responses of these individuals are consistent with the variation of the task demand in the simulated human machine systems. The cortical activities can be well modeled based on the deep and ensemble learning approaches.

Table 5. Performance of the SE-SDAE classifier on each subject.

Index of the Subject		*SEN*	*SPE*	*NPV*	*PRE*	*FH*	*FL*	*ACC*
1	*Training*	0.9937	0.9945	0.9938	0.9944	0.9941	0.9941	0.9941
	Testing	0.9311	0.9160	0.9290	0.9185	0.9248	0.9224	**0.9236**
2	*Training*	0.9965	0.9986	0.9965	0.9986	0.9976	0.9976	0.9976
	Testing	0.9505	0.9635	0.9501	0.9638	0.9571	0.9568	**0.9569**
3	*Training*	0.9814	0.9930	0.9813	0.9930	0.9872	0.9871	0.9872
	Testing	0.8624	0.8889	0.8539	0.8956	0.8787	0.8711	0.8750
4	*Training*	0.9790	0.9800	0.9794	0.9797	0.9793	0.9797	0.9795
	Testing	0.7493	0.7304	0.7283	0.7513	0.7503	0.7294	0.7403
5	*Training*	0.9923	0.9876	0.9924	0.9875	0.9899	0.9900	0.9899
	Testing	0.8636	0.8478	0.8667	0.8444	0.8539	0.8571	0.8556
6	*Training*	0.9894	0.9870	0.9897	0.9866	0.9880	0.9884	0.9882
	Testing	0.9010	0.9018	0.8886	0.9129	0.9069	0.8951	**0.9014**
7	*Training*	0.9896	0.9958	0.9896	0.9958	0.9927	0.9927	0.9927

Index of the Subject		*SEN*	*SPE*	*NPV*	*PRE*	*FH*	*FL*	*ACC*
	Testing	0.9008	0.9135	0.8955	0.9180	0.9093	0.9044	**0.9069**
8	*Training*	0.9909	0.9896	0.9910	0.9896	0.9903	0.9903	0.9903
	Testing	0.8781	0.8719	0.8768	0.8733	0.8757	0.8743	0.8750

According to the classification accuracy, the SE-SDAE can be effectively constructed, by using stacking principle on three types of classifiers with different learning structure, i.e., SDAE, NB and KNN models, to recognize variations of low and high cognitive levels indicated by multidimensional EEG features. The results comparison demonstrates that the SE-SDAE can further improve the testing accuracy against several classical shallow and deep workload estimators. At the level of the base classifiers, the two-dimensional fusion scheme integrating both SDAE and KNN achieved the best testing accuracy.

For the computational cost, the SDAE requires the lowest training time while the SE-SDAE with two-dimensional scheme took the highest training time. The potential reason is that the ensemble model with stacking architecture need to optimize multiple base classifiers as well as the secondary fusion classifier. It is also known that the KNN is a non-parametric learning machine and requires examining all training instances to provide a predicted class. This procedure has also occupied more computational resource.

We also compared the accuracy of the SE-SDAE with a single subject's data used for training and testing. It is found that the subject-specific performance of most classifiers is superior to the accuracy of the subject-specific case. The reason that the subject-specific model of cognitive load possess better accuracy may be the higher correlation between the EEG features and the control conditions of the same subject. On the other hand, Naive Bayesian model has the worst accuracy. It implies that Naive Bayesian classifier is not suitable for samples with non-independent EEG features. However, adding Naive Bayesian model to the classifier committee of the SE-SDAE based on the two-dimensional structure can improve the final classification accuracy. It can be seen that the two-dimensional structure itself has the ability to combine the different learning advantages of each model.

5. Conclusions

In this study, we present a new approach, SE-SDAE, to evaluate the human cognitive load based on the cortical EEG signals. The EEG has been carefully filtered and preprocessed to remove the muscular noise and find the neural responses under different conditions of the cognitive task. Both of the frequency and temporal features of the EEG data have been extracted. In the SE-SDAE framework, the stacking-based ensemble method has been employed to effectively model the mappings between the high-dimensional EEG features and the cognitive load levels. In particular, a heterogeneous classification committee including KNN and naive Bayesian classifiers has been introduced to improve the model generalization performance. The limitation of the present study may lies in two aspects. On one hand, the improvement in classification accuracy of the SE-SDAE is not significant compared to the SDAE. The essential of the stacking-based fusion scheme is the cross-validation, which avoids the over-fitting of data but decrease the size of training sample in each training iteration. In the future work, we will look for the ways to improve the shortcomings of the stacking fusion strategy in the SE-SDAE.

Conflict of interest

None declared.

Acknowledgements

This work is sponsored by the National Natural Science Foundation of China under Grant No. 61703277 and the Shanghai Sailing Program (17YF1427000).

References

Borghini, G., Astolfi, L., Vecchiato, G., Mattia, D., & Babiloni, F. (2014). Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. *Neuroscience & Biobehavioral Reviews*, *44*, 58–75.

Choi, M. K., Lee, S. M., Ha, J. S., & Seong, P. H. (2018). Development of an EEG-based workload measurement method in nuclear power plants. *Annals of Nuclear Energy*, *111*, 595–607.

Di Stasi, L. L., Antolí, A., & Cañas, J. J. (2013). Evaluating mental workload while interacting with computer-generated artificial environments. *Entertainment Computing*, *4*(1), 63–69.

Dornhege, G., del R. Millán, J., Hinterberger, T., Mcfarland, D. J., & Müller, K.-R. (n.d.). Improving Human Performance in a Real Operating Environment through Real-Time Mental Workload Detection. *Toward Brain-Computer Interfacing*.

Fallahi, M., Motamedzade, M., Heidarimoghadam, R., Soltanian, A. R., & Miyake, S. (2016). Effects of mental workload on physiological and subjective responses during traffic density monitoring: A field study. *Applied Ergonomics*, *52*, 95–103.

Fang, Z., Jia, T., Chen, Q., Xu, M., Yuan, X., & Wu, C. (2018). Laser stripe image denoising using convolutional autoencoder. *Results in Physics*, *11*, 96–104.

Fu, X., Luo, H., Zhong, S., & Lin, L. (2019). Aircraft engine fault detection based on grouped convolutional denoising autoencoders. *Chinese Journal of Aeronautics*, *32*(2), 296–307.

Görgel, P., & Simsek, A. (2019). Face recognition via Deep Stacked Denoising Sparse Autoencoders (DSDSA). *Applied Mathematics and Computation*, *355*, 325–342.

Habib, L., Pacaux-Lemoine, M. P., & Millot, P. (2017). A method for designing levels of automation based on a human-machine cooperation model. *IFAC-PapersOnLine*, *50*(1), 1372–1377.

Hicks, T. G., & Wierwille, W. W. (1979). Comparison of five mental workload assessment procedures in a moving-base driving simulator. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *21*(2), 129–143.

Jaquess, K. J., Lo, L.-C., Oh, H., Lu, C., Ginsberg, A., Tan, Y. Y., Lohse, K. R., Miller, M. W., Hatfield, B. D., & Gentili, R. J. (2018). Changes in Mental Workload and Motor Performance Throughout Multiple Practice Sessions Under Various Levels of Task Difficulty. *Neuroscience*, *393*, 305–318.

Ke, Y., Qi, H., Zhang, L., Chen, S., Jiao, X., Zhou, P., Zhao, X., Wan, B., & Ming, D. (2015). Towards an effective cross-task mental workload recognition model using electroencephalography based on feature selection and support vector machine regression. *International Journal of Psychophysiology*, *98*(2, Part 1),

Lan, R., Li, Z., Liu, Z., Gu, T., & Luo, X. (2019). Hyperspectral image classification using k-sparse denoising autoencoder and spectral–restricted spatial characteristics. *Applied Soft Computing*, *74*, 693–708.

Lee, D., Choi, S., & Kim, H.-J. (2018). Performance evaluation of image denoising developed using convolutional denoising autoencoders in chest radiography. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *884*, 97–104.

Lee, S., Kim, H. J., & Kim, S. B. (2020). Dynamic dispatching system using a deep denoising autoencoder for semiconductor manufacturing. *Applied Soft Computing*, *86*, 105904.

Lewis, M. M. (2019). Cognitive Load, Anxiety, and Performance During a Simulated Subarachnoid Block. *Clinical Simulation in Nursing*, *36*, 30–36.

Makeig, S., & Inlow, M. (1993). Lapse in alertness: coherence of fluctuations in performance and EEG spectrum. *Electroencephalography and Clinical Neurophysiology*, *86*(1), 23–35.

Makeig, S., & Jung, T. P. (1995). Changes in alertness are a principal component of variance in the EEG spectrum. *Neuroreport*, *7*(1), 213.

Meng, Z., Zhan, X., Li, J., & Pan, Z. (2018). An enhancement denoising autoencoder for rolling bearing fault diagnosis. *Measurement*, *130*, 448–454.

Parasuraman, R., & Jiang, Y. (2012). Individual differences in cognition, affect, and performance: Behavioral, neuroimaging, and molecular genetic approaches. *NeuroImage*, *59*(1), 70–82.

Rusnock, C. F., & Borghetti, B. J. (2018). Workload profiles: A continuous measure of mental workload. *International Journal of Industrial Ergonomics*, *63*, 49–64.

Rutherford, A. (1987). Handbook of perception and human performance. Vol 1: Sensory processes and perception. Vol 2: Cognitive processes and performance.: K. R. Boff, L. Kaufman and J.P. Thomas (eds) John Wiley and Sons, 1986, ISBN 0-47 1-85061-6, £90.75. *Applied Ergonomics*, *18*(4), 340.

Ryu, K., & Myung, R. (2005). Evaluation of mental workload with a combined measure based on physiological indices during a dual task of tracking and mental arithmetic. *International Journal of Industrial Ergonomics*, *35*(11), 991–1009.

Ting, L. I. (2015). A Deep Learning Method for Braille Recognition. *Computer & Modernization*.

Vuckovic, A., Radivojevic, V., Chen, A. C. N., & Popovic, D. (2002). Automatic recognition of alertness and drowsiness from EEG by an artificial neural network. *Medical Engineering & Physics*, *24*(5), 349–360.

Wang, Z., Hope, R. M., Wang, Z., Ji, Q., & Gray, W. D. (2012). Cross-subject workload classification with a hierarchical Bayes model. *NeuroImage*, *59*(1), 64–69.

Wilson, G F. (2005). Operator functional state assessment for adaptive automation implementation. *Conference on Biomonitoring for Physiological and Cognitive Performance*.

Wilson, Glenn F, & Eggemeier, F. T. (2020). *Psychophysiological assessment of workload in multi-task environments*.

Xian-Bao, W., Wen-Xiu, H. E., Xin-Gang, W., Ming-Hai, Y., & Yun-Tao, Q. (2016). Capsule Defects Detection Based on Stacked Denoising Autoencoders. *Computer Ence*.

Xu, F., Tse, W. tai P., & Tse, Y. L. (2018). Roller bearing fault diagnosis using stacked denoising autoencoder in deep learning and Gath–Geva clustering algorithm without principal component analysis and data label. *Applied Soft Computing*, *73*, 898–913.

Yin, Y. H., Nee, A. Y. C., Ong, S. K., Zhu, J. Y., Gu, P. H., & Chen, L. J. (2015). Automating design with intelligent human–machine integration. *CIRP Annals*, *64*(2), 655–677.

Yin, Z., & Zhang, J. (2017). Cross-session classification of mental workload levels using EEG and an adaptive deep learning model. *Biomedical Signal Processing and Control*, *33*, 30–47.

Young, M. S., Brookhuis, K. A., Wickens, C. D., & Hancock, P. A. (2015). State of science: mental workload in ergonomics. *Ergonomics*, *58*(1), 1–17.

Yu, J. (2019). A selective deep stacked denoising autoencoders ensemble with negative correlation learning for gearbox fault diagnosis. *Computers in Industry*, *108*, 62–72.

Zhang, Jian, Hou, Z., Wu, Z., Chen, Y., & Li, W. (2016). Research of 3D face recognition algorithm based on deep learning stacked denoising autoencoder theory. *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*.

Zhang, Jianhua, Yin, Z., & Wang, R. (2015). Recognition of Mental Workload Levels Under Complex Human–Machine Collaboration by Using Physiological Features and Adaptive Support Vector Machines. *IEEE Transactions on Human-Machine Systems*, *45*(2), 200–214.

Zhou, Z. (2016). Machine Learning. *Tsinghua University Press*.