TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Transactions of the Institute of Measurement and Control
1–17
© The Author(s) 2015
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0142331215587568
tim.sagepub.com
SAGE

Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers

Junming Zhang, Yan Wu, Jing Bai and Fuqiang Chen

Abstract

This paper presents an automatic sleep stage method combining a sparse deep belief net and combination of multiple classifiers for electroencephalogram, electrooculogram and electromyogram. The sparse deep belief net was applied to extract features from these signals automatically, and the combination of multiple classifiers, utilizing the extracted features, assigned each 30-s epoch to one of the five possible sleep stages. More importantly, we proposed a new voting principle based on classification entropy to enhance the classification performance further by harnessing the complementary information provided by the individual classifier. Differently from existing methods, our method used unsupervised feature learning to extract features automatically from raw sleep data and classification based on the learned features. The results of automatic and manual scorings were compared on an epoch-by-epoch basis. The accuracies for wake, S1, S2, SWS and REM were 98.49%, 80.05%, 91.2%, 98.22% and 95.31%, respectively, and the total accuracy of sleep stage was 91.31%. The results demonstrated that the sparse deep belief net was an efficient feature extraction method for sleep data, and the combination of multiple classifiers based on classification entropy performed well on sleep stages.

Keywords

Combination of multiple classifiers, feature extraction, sleep stage, sparse deep belief net

Introduction

Sleep diseases, such as insomnia, sleepwalking, narcolepsy and nocturnal breathing disorders, seriously affect the patient's quality of life and thus need to be treated. Effective diagnosis and treatment of patients with sleep-related complaints is currently an urgent and heavily research topic in the healthcare community. In order to diagnose sleep issues, all-night polysomnographic (PSG) recordings, including electroencephalogram (EEG), electrooculogram (EOG) and electromyogram (EMG), are usually taken from the patients, and the recordings are divided into 20- or 30-s epochs before being scored by a well-trained expert according to the Rechtschaffen and Kales (R&K) rules (Rechtschaffen and Kales, 1968). They established a method based on a set of rules to assign labels to time intervals in the PSG representing different states of sleep: Wake (W), stages 1–4 (S1, S2, S3 and S4) and rapid eye movement (REM). Because there is no clear distinction between S3 and S4, S3 and S4 were combined as slow wave sleep (SWS) stage (Charbonnier et al., 2011; Hsu et al, 2013; Liang et al., 2012; Stanus et al., 1987; Zoubek et al., 2007). In 2008, the American Academy of Sleep Medicine (AASM) discontinued the use of S3 and S4 (Schulz, 2008).

Because visual sleep scoring is a subjective process and in general a tedious task requiring much time and effort for the physician, automatic sleep stage methods based on multichannel data fusion strategy were developed. The literature provides several examples involving different techniques: rule-based methods (Krakovská and Mezeiová, 2011; Kurihara and Watanabe, 2012), artificial neural networks (Charbonnier et al., 2011; Ronzhina et al., 2012), support vector machines (SVMs) (Crisler et al., 2008; Gudmundsson et al., 2005; Koley and Dey, 2012; Yιlmaz et al., 2010), spectral analysis (Álvarez-Estévez et al., 2013; Jha et al., 2014; Khalighi et al., 2012) and non-linear feature analysis (Acharya et al., 2010; Kannathal et al., 2005; Krakovská and Mezeiová, 2011). A detailed review of the literature can be found in Hamida and Ahmed (2013), and Penzel and Conradt (2000), in which it is stated that automation of hypnogram generation is an open area of research still unsolved. However, some problems still exist in those methods, as follows:

1) The precision of the sleep stage estimation needs to be improved.
2) Traditionally, almost all the research methods were based on handcrafted features and then the features were used for classification, but the latent information of sleep data does not play a key role in automatic sleep stage classification.

College of Electronics & Information Engineering, Tongji University, Shanghai 201804, China

Corresponding author:
Yan Wu, College of Electronics & Information Engineering, Tongji University, Shanghai 201804, China.
Email: yanwu@tongji.edu.cn

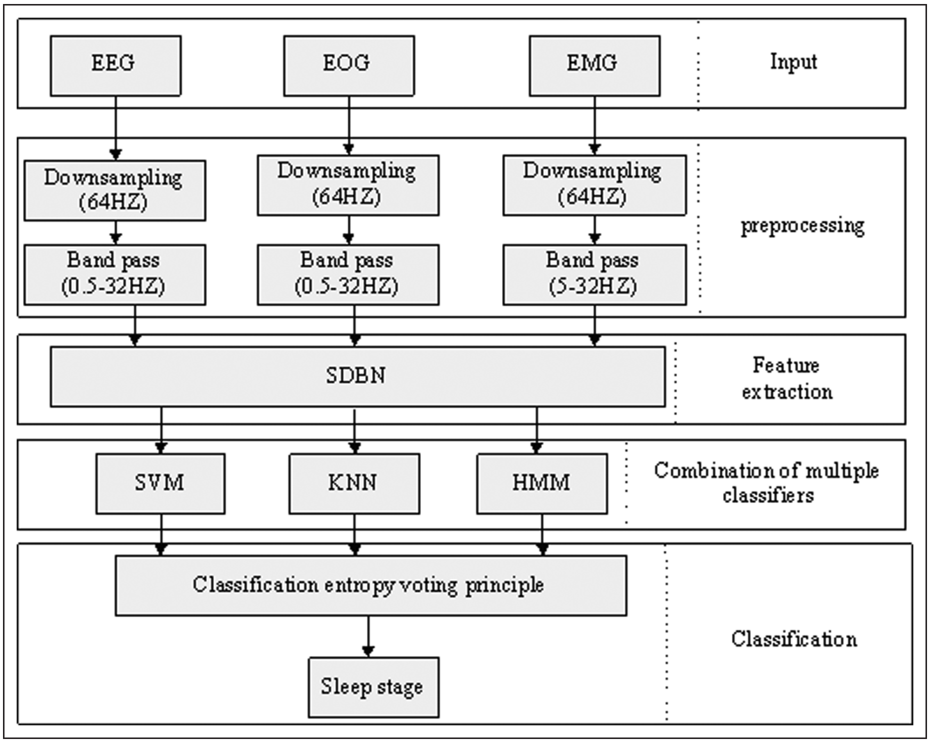


Figure 1. Flowchart of multi-parameter sleep-staging method.

Recently, a new signal and information processing method called deep learning (Bengio et al., 2006; Hinton et al., 2006; Lecun et al., 1998; Lee et al., 2007) has been proposed for unsupervised feature learning. Deep learning has been applied to many domains, such as biomedical signals EEG (Wang et al., 2013; Wulsin et al., 2010; Wulsin et al., 2011), EMG, EOG (Wang and Shang, 2013), and EEG, EMG and EOG (Langkvist et al., 2012). These studies showed that deep learning can be applied to raw physiological data to learn relevant features effectively.

Langkvist et al. (2012) explored the feasibility of applying a deep belief network (DBN) to sleep data. In general, DBN tends to learn distributed, non-sparse representations. However, sparse representation models resemble biological visual system characteristics, and they are able to learn more complex features than simple oriented bars. Motivated in part by the biological visual system characteristics, Lee et al. (2007) posed a sparse variant of the deep belief network (SDBN) model.

In this paper, SDBN is first applied to analyse the EEG, EMG and EOG signals of different sleep stages. We present a system based on SDBN that can automatically extract features from the raw physiological data and then build a combination of multiple classifiers to predict the sleep stages. The rest of the paper is organized as follows: the signal preprocessing, feature extraction and combination of multiple classifiers are presented in the next section. Then the experimental dataset, results and performance comparison with some existing methods are presented. Discussion of the experimental results is given, finally followed by the conclusions.

Methods

Sleep stage system

Figure 1 shows the flowchart of the multi-parameter (EEG, EOG and EMG) sleep stage method that includes four parts: 1) preprocessing; 2) feature extraction; 3) combination of multiple classifiers; and 4) voting classification.

Preprocessing

All signals are preprocessed by notch filtering at 50 Hz in order to cancel out power line disturbances. Although brain activities of EEG signals are divided into many frequency rhythms, the most important frequency range for sleep EEG is 0.5–32 Hz. Similarly, that for EOG is 0.5–32 Hz and that for EMG is 5–32 Hz (Langkvist et al., 2012; Zoubek et al., 2007). Based on these frequency ranges, after downsampling the EEG, EOG and EMG signals to 64 Hz, band-pass filters of 0.5–32 Hz for EEG and EOG, and 5–32 Hz for EMG are used. Then, the EEG, EOG and EMG signals are segmented (divided) into epochs of 30 s, each epoch corresponding to a single sleep stage.

Feature extraction

Each channel of the data is divided into segments of 30 s with zero overlap for feature extraction. The SDBN contains two major points: 1) DBN and 2) sparse coding. The DBN is important for our system, and the sparse coding enables DBN to learn sparse representations.

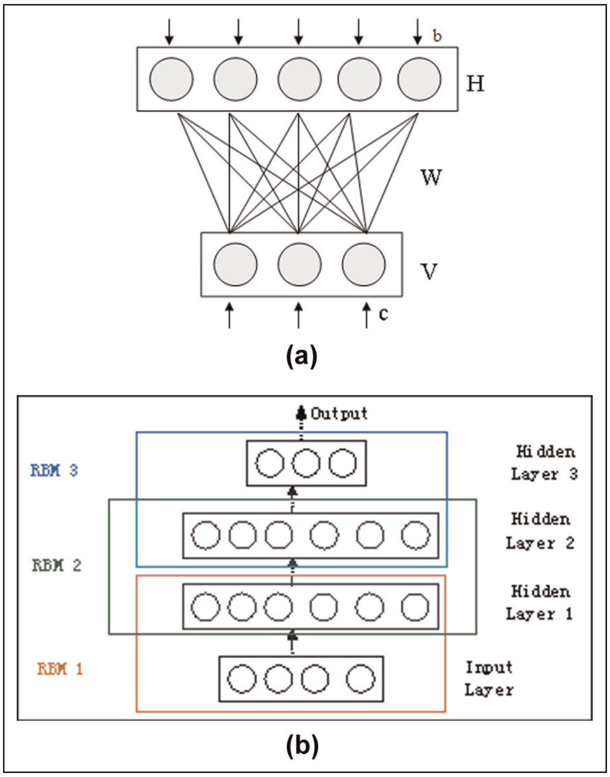


Figure 2. A graphical depiction of (a) a restricted Boltzmann machine (RBM) and (b) deep belief network (DBN).

DBN. DBN is a probabilistic generative model with deep architecture that is capable of learning high dimensional manifolds of the data. It consists of a multilayer neural network with each layer an RBM containing visible units, $\boldsymbol{v}$, and hidden units, $\boldsymbol{h}$ (in this paper, bold italic lowercase letter is used to denote a vector, bold uppercase letter to denote a matrix, and italic lowercase letter to denote a number, unless otherwise stated). There are no connections between the nodes in the same layer. The visible and hidden units have a bias vector, $\boldsymbol{c}$ and $\boldsymbol{b}$, respectively. The visible and hidden units are connected by a weight matrix, **W**. A graphical depiction of an RBM is shown in Figure 2(a). To form a DBN, a user-defined number of RBMs are trained one after another and then stacked on top of each other, where the visible layers of higher RBM are the hidden units of the previous RBM (Figure 2b). The output from a lower-level RBM is the input to a higher-level RBM.

An RBM has a joint distribution and an energy function for a given visible and hidden vector. They are defined as:

$$p(\boldsymbol{v},\boldsymbol{h}|\theta) = \frac{1}{z}\exp^{-\mathrm{E}(\boldsymbol{v},\boldsymbol{h}|\theta)} \qquad (1)$$

$$E(\boldsymbol{v},\boldsymbol{h}|\theta) = -\sum_{\substack{i\in\text{visible}\\ j\in\text{hidden}}} v_i w_{ij} h_j - \sum_{i\in\text{visible}} c_i v_i - \sum_{j\in\text{hidden}} b_j h_j \qquad (2)$$

where $\theta = \{\mathbf{W},\boldsymbol{c},\boldsymbol{b}\}$, v_i, h_j are the binary states of visible unit i and hidden unit j, c_i, b_j are their biases and w_{ij} is the weight of the edge connecting v_i and $h_{j.}$ Z is the partition function that ensures that the distribution is normalized.

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)} \qquad (3)$$

In fact, we often care about the distribution of input data. The probability that the network assigns to a visible vector, $\boldsymbol{v}$, is given by summing over all possible hidden vectors.

$$p(\boldsymbol{v}|\theta) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)} \qquad (4)$$

To obtain the distribution, we need to compute Z, which is difficult. Fortunately, because there are no connections between the nodes in the same layer and there are only connections between the two layers, the states of all the hidden units of h_j are independent given a specific visible vector $\boldsymbol{v}$ and so are the visible units v_i given a specific hidden vector $\boldsymbol{h}$. Given visible vector $\boldsymbol{v}$, the binary state h_j, of each hidden unit, j, is set to 1 with probability

$$\mathrm{p}(h_j = 1|\boldsymbol{v},\theta) = \sigma(c_j + \sum_i w_{ij} v_j) \qquad (5)$$

Given a hidden vector $\boldsymbol{h}$, it is also very easy to obtain an unbiased sample of the state of a visible unit

$$\mathrm{p}(v_i = 1|\boldsymbol{h},\theta) = \sigma(b_i + \sum_j w_{ij} h_j) \qquad (6)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$.

From Equations (1) and (4), we can obtain the conditional probability:

$$p(\boldsymbol{h}|\boldsymbol{v},\theta) = \frac{p(\boldsymbol{v},\boldsymbol{h}|\theta)}{p(\boldsymbol{v}|\theta)} = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)}}{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)}} \qquad (7)$$

To obtain $p(\boldsymbol{v},\boldsymbol{h}|\theta)$, the joint distribution of visible and hidden units is much more difficult. To solve the problem, we can resort to Gibbs sampling. For the detailed information of Gibbs sampling, readers are referred to Gelfand (2000). Although Gibbs sampling can solve the problem, it is ineffective, especially in high-dimensional feature space. A much faster learning procedure contrastive divergence (CD) was proposed in Hinton (2002). CD is proposed by Hinton and it can be used to train RBM. Initially, we are given v_i then we can obtain h_j by (5), and the value of h_j is determined by comparing a random value γ ranging from 0 to 1 with the conditional probability $\mathrm{p}(h_j = 1|\boldsymbol{v},\theta)$. Then we can reconstruct $\boldsymbol{v}$ by $\mathrm{p}(v_i = 1|\boldsymbol{h},\theta)$. We can repeat the above process backward and forward until the reconstruction error is small enough or it has reached the maximum number of iterations, which is set beforehand. Then the weight and biases can be updated according to the following rule:

$$\Delta W_{ij} = \kappa(<v_i h_j>_{input} - <v_i h_j>_{recog}) \qquad (8)$$

$$\Delta c_i = \kappa(<v_i>_{input} - <v_i>_{recog}) \qquad (9)$$

$$\Delta b_j = \kappa(<h_j>_{input} - <h_j>_{recog}) \qquad (10)$$

where κ is a learning rate, which influences the speed of convergence. For the much more detailed information of CD, readers are referred to Hinton (2002).

Sparse coding. RBM distributed representation means that given a case, many units in the hidden layer are activated. This might seem unrelated to the selectivity of a unit, but the concept of sparse coding can guarantee the selectivity. For training the parameters of the model, the objective is to maximize the log-likelihood of the data. In order to perform the sparsity, a regularization term that penalizes a deviation of the expected activation of the hidden units from a fixed level p was added. Given a training set $\{v(1), ..., v(m)\}$ with m examples, Lee et al. (2007) pose the following optimization problem:

$$\text{minimize}\{w_{ij}, c_i, b_j\} - \Sigma_{l=1}^{m} \log \Sigma_h \text{P}(\boldsymbol{v}^l, \boldsymbol{h}^l) + \lambda \Sigma_{j=1}^{n} |p - \frac{1}{m} \Sigma_{l=1}^{m} E[h_j^l | \boldsymbol{v}^i]|^2,$$

where E[] is the conditional expectation given the data, λ is a regularization constant, and p is a constant controlling the sparseness of the hidden units h_j. Thus, the objective is the sum of a log-likelihood term and a regularization term. The training process resembles that of RBM. For the detailed information of SDBN, the readers are referred to Lee et al. (2007).

Combination of multiple classifiers

Although many classifiers have been used to classify the sleep data into sleep stages, the most common classifiers are K-nearest neighbour (KNN), decision tree learning (DTL), hidden Markov model (HMM), artificial neural network (ANN) and support vector machine (SVM). From Hamida and Ahmed (2013), we can draw the following conclusions: 1) although KNN requires a large computation time and a great memory space, it is highly simple and usually produces good results; (2) DTL is not suitable due to the high inter-subject variability in the sleep signals; (3) HMM has the advantage of being suitable due to the multivariate temporal nature of the PSG data; (4) SVM is more straightforward to implement and has a higher accuracy level than that of ANN for the classification of sleep stages (Lee and Yoo, 2013). Because KNN and SVM are simple to implement and DTL is not suitable for sleep data, SVM, KNN and HMM were chosen to classify the sleep stages. The principle of combination of multiple classifiers is a voting principle based on classification entropy.

SVM. SVM is a very popular type of classifier and has been used in sleep stages (Crisler et al., 2008; Lee and Yoo, 2013). Much of the work on classifications in high-dimensional feature space pointed out the superiority of SVM over traditional statistical and neural classifiers (Guyon et al., 2002; Melgani and Bruzzone, 2004). Suppose training vectors $\boldsymbol{x}_i \in R^n$, $i = 1, 2, \ldots, N$, $y_i \in \{1, -1\}$. The basic idea behind SVM is to introduce a mapping $\boldsymbol{x} -> \Gamma(\boldsymbol{x})$ that maps the data into a linear space LS where they are linearly separable. One of the most important properties of SVM is that the mapping Γ does not need to be explicitly known, and only the kernel function KF($\boldsymbol{x_1}$, $\boldsymbol{x_2}$) is needed. There are many kernels can be chosen. In this paper, the radial basis function (RBF) kernel is used. The reason is that the kernel non-linearly maps samples into a higher dimensional space, so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is non-linear. In addition, the sigmoid kernel behaves like RBF for certain parameters, and the polynomial kernel has more hyperparameters than the RBF kernel (Hsu et al., 2003). Almost all the papers with sleep stage related to SVM make use of RBF (Gudmundsson et al., 2005; Khalighi et al., 2012; Lee and Yoo, 2013).

$$\text{KF}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \text{e}^{-\gamma||x_i - x_j||},$$

where γ is a parameter to be specified. If the data is not linearly separable in LS, usually we should solve the following optimization problem (Chang and Lin, 2011).

$$\min_{\boldsymbol{w}, b, \xi} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{N} \xi_i \tag{11}$$

subject to $y_i(\boldsymbol{w}^T \Gamma(\boldsymbol{x}_i) + b) > = 1 - \xi_i$

$$\xi_i > = 0, \; i = 1, 2, \ldots, N,$$

where $C > 0$ is the regularization parameter. Due to the possible high dimensionality of the vector variable $\boldsymbol{w}$, we usually solve the following dual problem.

$$\min_{\alpha} \frac{1}{2} \alpha^T \mathbf{Q} \alpha - \boldsymbol{e}^T \alpha \tag{12}$$

subject to $\boldsymbol{y}^T \alpha = 0$,

$$0 < \alpha_i < C, \; i = 1, 2, \ldots, N,$$

where e = [1;...; 1]$^{\text{T}}$ is the vector of all ones, $\mathbf{Q}$ is an $N \times N$ positive semi-definite matrix, $Q_{ij} = y_i y_j KF(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Solving the dual problem avoids the need for computing the features Γ(x) explicitly. Many other methods are available for multi-class SVM classification. In this paper, the strategy used here follows Chang and Lin (2011). It implements the 'one-against-one' approach for multi-class classification. The software is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

KNN. KNN has also been used in sleep stages (Yιlmaz et al., 2010). In this paper, Euclidian distance is used as a measure of distance, because the Euclidian distance calculates the straight line between objects. The Manhattan distance calculates the fold line distance, and the Minkowski distance can be considered a generalization of both the Euclidean distance and the Manhattan distance. For sleep stage classification, the linear distance is more suitable. In addition, most papers with sleep stage related to KNN make use of Euclidian distance (Gudmundsson et al., 2005; Kanoje and Shingare, 2014; Yιlmaz et al., 2010). KNN works based on the minimum distance from the query instance to the training samples. The discriminant function of the nearest neighbour can be obtained as follows:

$$g_i(x) = \min ||x - x_i^n||,$$
$$\text{if } g_j(x) = \min g_i(x), \text{then } x \in \Theta_j.$$

where $i = 1, \ldots, C$, C is the total numbers of class, $n = 1, \ldots, N$, N is the total number of input samples. Θ_i denotes the sample set of the class i. After we gather KNNs, we take simple majority of these KNNs to be the prediction of the query instance.

HMM. HMM allows analysis of non-stationary multivariate time series by modelling both the probability density functions of locally stationary multivariate data and the transition probabilities between these stable states. It assumes a number of discrete hidden states, discrete or continuous valued observation, and the probability of transition from one state to the other. In this paper, as the signals and the features are real valued observations, continuous HMM were used. For real valued observation, an HMM can be defined by the following elements:

a) Suppose $\boldsymbol{Q}$ is the set of all states, $\boldsymbol{V}$ is the set of observable states.

$$\mathbf{Q} = \{\mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_\mathrm{N}\}, \mathbf{V} = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_\mathrm{M}\},$$

where N is the number of states, observable states is M. $\boldsymbol{I} = (i_1, i_2, \ldots, i_T)$ is the state sequence where the length is T, and the corresponding observation sequence is $\boldsymbol{O} = (o_1, o_2, \ldots, o_T)$.

b) State transition probability distribution, $\mathbf{A} = [a_{ij}]_{N\mathrm{x}N}$, where $a_{ij} = p(i_{t+1} = q_j | i_t = q_i)$, $1 <= i, j <= N$.

c) Emission probability distribution in state j, $\mathbf{B} = [b_j(k)]_{N\mathrm{x}M}$, where $b_j(k) = p[o_t = v_k | i_t = q_j]$.

d) Initial state distribution $\pi = \{\pi_i\}$ where $\pi_i = p\{i_1 = q_i\}$.

The classification using HMM typically includes the following steps (Rabiner, 1989; Rabiner and Juang, 1986):

a) Initialization of state transition probability $\mathbf{A}$ and initial state distribution π.

b) Training of HMM for each class.

c) Computation of log-likelihood that each model gives to the test signal.

d) Selection of the most likely model.

Voting principle. Classifier combination methods have been proved an effective tool for improving the performance of a single classifier. The difficulty is to find the combination function accepting M-dimensional score vectors from M classifiers and outputting a final classification. In order to solve the problem and improve the performance of combination of multiple classifiers, we proposed a new voting principle based on classification entropy (CE).

Traditionally, if one object is identified as the same class by most base classifiers, the object is labelled to this class, which is named the majority voting method. The majority voting method has many extensions, such as weighted majority voting and Borda count method (Xie and Minn, 2012). However, almost all the weighted majority voting methods neglect either total performance[1] or local performance[2] of each classifier. We present methods considering not only the total performance but also the local performance. When the local performance of one classifier is less controlling constant, the weight will be penalized. Otherwise, the weight will be enhanced. So, the weights of classifiers will be determined automatically and adaptively with this method. The value of controlling constant is important. It determines which values of classifiers will be penalized.

Let $\boldsymbol{C} = \{c_1, c_2, c_3, \ldots, c_M\}$ be a set of trained classifiers and all objects are represented by feature vectors $\boldsymbol{x} \in R^n$, where R^n denotes n-dimensional feature space, and M is the number of classifiers. The set of label set is $\boldsymbol{B} = \{b_1, b_2, b_3, \ldots, b_L\}$, where L is the number of classes. Each object belongs to one class and each classifier obtains a feature vector $\boldsymbol{x}$ as its input and assigns it to label in $\boldsymbol{B}$. $\mathrm{R}(\boldsymbol{x}) = b_j$ denotes the label of $\boldsymbol{x}$ is b_j and $\mathrm{p_i}(b_j|\boldsymbol{x})$ denotes the probability of $\boldsymbol{x}$ belongs to b_j based on c_i, where $i = 1, 2, \ldots, M$, and $j = 1, 2, 3, \ldots, L$. The arithmetic of CE consists of the following steps:

1) According to the training set $\boldsymbol{X}$ and classifiers set $\boldsymbol{C}$, we calculate the confusion matrix ($\mathbf{CM}_{\boldsymbol{i}}$) of c_i based on $\boldsymbol{X}$, $\boldsymbol{x} \in \boldsymbol{X}$.

$$\mathbf{CM}_i = \begin{bmatrix} p^i_{11} & p^i_{12} & \cdots & p^i_{1L} \\ p^i_{21} & p^i_{22} & \cdots & p^i_{2L} \\ \vdots & \cdots & p^i_{jk} & \vdots \\ p^i_{L1} & p^i_{L2} & \cdots & p^i_{LL} \end{bmatrix},$$

where p^i_{jk} denotes the probability $\mathrm{p}(b_k|\boldsymbol{x})$, $\mathrm{R}(\boldsymbol{x}) = b_j$.

2) Calculate the total accuracy ($\boldsymbol{TAC}$) of the ith classifier.

$$Num = \sum_{k=1}^{L} \sum_{j=1}^{L} p^i_{kj}, \; TAC_i = \frac{\sum_{j=1}^{L} p^i_{jj}}{Num},$$

$$TAC = [TAC_1 \; TAC_2 \; \ldots \; TAC_L].$$

3) Calculate the local accuracy ($\boldsymbol{LAC}$) of every classifier and classification matrix of the classifier combination.

$$LAC_i = [p^i_{11} \, p^i_{22} \, \cdots \, p^i_{LL}],$$

$$\mathbf{LAC} = \begin{bmatrix} e_{11} & \ldots & e_{1L} \\ \vdots & e_{ij} & \vdots \\ e_{M1} & \ldots & e_{ML} \end{bmatrix} = \frac{[\boldsymbol{LAC}_1 \; \boldsymbol{LAC}_2 \; \ldots \; \boldsymbol{LAC}_M]^T}{Num}$$

4) According to **LAC** and ***TAC***, we calculate the weights of every classifier for the labels (***WCCL***). ***WCCL*** is very important for our methods (Table 1).

$$WCCL_{ij} = \exp(2(e_{ij} - \beta))TAC_i,$$

$WCCL_{ij}$ denotes the weight of ith classifier with the class j, where β is a controlling constant. When e_{ij} is larger than β, the weight will be enhanced and the opposite is true.

5) $s^i \in B$ denotes the output of the ith classifier. Let $y_j^i = WCCL_{ij}k_j^i$, where $k_j^i = \begin{cases} 1 & \text{if } s^i = j \\ 0 & \text{otherwise} \end{cases}$.

6) The final voting classification: $\max\{CE_j\}$

$$CE_j = \frac{\sum_{m=1}^{M} y_j^m}{\sum_{m=1}^{M}\sum_{j=1}^{L} y_j^m}.$$

Experimental results

Dataset

The dataset we used is obtained from the UCD database, which is available online at http://physionet.org/physiobank/database/ucddb. The dataset consists of 25 sleep-disordered–breathing subjects' full overnight PSG recordings (21 males and four females with an average age of 50, average weight 95 kg and average height 173 cm). Each recording contains two EEG channels (C3–A2 and C4–A1), two EOG channels and 1 EMG channel, as well as an annotation file with detailed onset time and duration of every hypopnoea event. In this paper, the single-channel EEG signal (C3–A2), left EOG and EMG signals were used. The sample rates of EEG, EOG and EMG are 128, 64 and 64 Hz, respectively.

Table 1. Weights of every classifier for the class labels.

Classifier	Class 1	Class 2	…	Class L
c_1	w_{11}	w_{12}	$\cdots$	w_{1L}
c_2	w_{21}	w_{22}	$\cdots$	w_{2L}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
c_M	w_{M1}	w_{M2}	$\cdots$	w_{ML}

To reduce unbalanced data and train the model better, we selected recordings with sleep efficiency larger than 82%. In the end, 10 recordings were selected. A more detailed description of the 10 recordings is listed in Table 2. The 10 all-night PSG sleep recordings were scored by the sleep expert according to the R&K rules. Each 30-s epoch was classified into one of the five sleep stages. In our experiments, only epochs belonging to the five sleep stages were used, artifact and indeterminate epochs were rejected. The distribution of the five sleep stages was shown in Table 3.

Statistical evaluation methods

To evaluate our system performance, we used accuracy and F_1 score as the evaluation criteria. The definition of F_1 score is as follows (Wulsin et al., 2010):

$$F1 = (2 * \text{precision} * \text{recall})/(\text{precision} + \text{recall}).$$

As the sleep stages are multi-class problem, we reduced each class's precision and recall calculations to a one-against-others problem.

Note that all the experiment results were based on leave-one-out cross-validation of the 10 recordings. One recording was left out for testing each time while the other nine recordings were used for training. The testing results were averaged over 10 recordings and then as the final cross-validation results.

Results

The input data consisted of the concatenation of EEG, EOG and EMG, in which the window width of EEG, EOG and EMG is 1920, respectively. Hence, the input dimension is 5760. A three-layer SDBN with architecture of 5760–200–200

Table 2. A detailed description of the 10 recordings.

Study number	Height (cm)	Weight (kg)	Gender	PSG AHI	BMI	Age	ESS	Study duration (h)	Sleep efficiency (%)	PSG duration (s)
002	172	100.3	M	23	33.9	54	16	6.2	84	22470
006	185	103.5	M	31	30.2	52	3	6.7	89	24267
007	183	84	M	12	25.1	47	15	6.8	90	24405
010	174	119	M	34	39.3	38	2	7.6	92	27211
012	179	97.5	M	25	30.4	51	12	7.2	85	25941
017	176	117	M	12	37.8	53	7	6.6	87	23684
019	178	97.8	M	16	30.9	49	18	7.1	92	25573
024	172	99.9	M	24	33.8	54	19	7.6	83	27250
026	175	84	M	14	27.4	49	9	7	87	25160
027	182	93	M	55	28.1	45	10	7.4	86	26791

PSG, polysomnographic; AHI, apnea hypopnea index; ESS, epworth sleepiness score.

Table 3. The distribution of the sleep stages in each subject.

Study number	S1 (%)	S2 (%)	SWS (%)	REM (%)	Wake (%)
002	28.48	22.99	11.63	20.72	16.18
006	22.4	11.76	30.69	23.76	11.39
007	6.89	50.92	15.62	16.24	10.33
010	13.23	51.27	8.71	18.63	8.16
012	12.02	28.8	18.59	21.54	19.05
017	10.04	39.89	10.96	4.36	34.74
019	28.22	26.38	15.24	9.62	20.54
024	10.77	30.39	6.30	16.71	37.41
026	17.63	21.33	13.69	8.26	39.09
027	13.36	40.53	13.91	14.02	18.18

For abbreviations, see text.

Table 4. Confusion matrix for support vector machine (SVM) (%).

	SWS	S2	S1	REM	Wake
SWS	98.4	0.44	0.89	0	0.27
S2	5.11	83.09	10.98	0.41	0.41
S1	0.73	21	62.09	12.24	3.94
REM	0	6.11	0	91.7	2.18
Wake	0	0.89	0.63	0.36	98.12

For abbreviations, see text.

Table 5. Confusion matrix for K-nearest neighbor (KNN).

%	SWS	S2	S1	REM	Wake
SWS	94.58	1.11	4.31	0	0
S2	6.50	76.31	16.64	0.55	0
S1	0.73	9.56	65.52	18.03	6.16
REM	0.30	5.77	0.21	91.2	2.53
Wake	0	0	1.51	1.67	96.82

For abbreviations, see text.

Table 6. Confusion matrix for hidden Markov model (HMM).

%	SWS	S2	S1	REM	Wake
SWS	97.78	0.40	1.56	0	0.27
S2	7.05	82.79	9.86	0.30	0
S1	0.73	9	69.76	16.79	3.72
REM	0.32	5.29	1.1	91.75	1.54
Wake	0	0.57	3.02	0.16	96.25

For abbreviations, see text.

and learning rate of 0.05 were used in this paper, where $\lambda = 1/p$ in each layer, $p = 0.02$ and 0.05 for the first and second layers. The SVM experiment was carried out using the LIBSVM package (Chang and Lin, 2011). The parameters of SVM are: C = 8 (regularization parameter), $\gamma = 0.5$, e = 0.0001(termination criterion). The parameter of KNN is: $k = 33$. The controlling constant $\beta = 0.85$. Training was done on Windows XP, a 32-bit machine with a dual-core Intel E5300, 2.6 GHz. All the experiments in this paper were performed based on Matlab (2012a).

The confusion matrices for sleep stages using each individual classifier and classifier combination were presented in Tables 4–7. From these tables, we can see that the misclassification is among S1, REM and S2. Table 4 shows that the poorest performance was S1, with an accuracy of 62.09%, while the accuracies for the other stages were 83.09–98.4%.

Table 7. Confusion matrix for combination of multiple classifiers.

%	SWS	S2	S1	REM	Wake
SWS	98.22	0.4	1.11	0	0.27
S2	3.72	91.2	4.92	0.16	0
S1	0.36	8.03	80.05	9.54	2.02
REM	0.32	2.71	0.21	95.31	1.45
Wake	0	0.57	0.57	0.36	98.49

For abbreviations, see text.

Table 8. F_1 score and total accuracy for each classifier and classifiers combination.

	SWS	S2	F_1, S1	REM	Wake	Total accuracy
SVM	0.946	0.7741	0.7193	0.9	0.9231	0.8398
KNN	0.9135	0.7478	0.7137	0.9055	0.8976	0.8205
HMM	0.9259	0.7942	0.7667	0.9148	0.9254	0.8555
SVM + KNN + HMM	0.9557	0.8868	0.8630	0.9381	0.9543	0.9131

For abbreviations, see text.

Twenty-one percent of S1 epochs were misclassified as S2, and 12.24% of S1 epochs were misclassified as REM. Table 5 provides the accuracy for each stage using KNN, where the poorest performance was S1, with an accuracy of 65.52%, while the accuracies for the other stages were 76.31–96.82%, and 18.03% of S1 epochs were misclassified as REM. Table 6 shows the classification performance for the HMM, where the poorest performance was S1, with an accuracy of 69.76%, and 16.79% S1 epochs were misclassified as REM. The classification performances of each sleep stage using the combination of multiple classifiers were shown in Table 7. For the combination of multiple classifiers, the lowest accuracy was observed for S1 (80.05%) with significantly less than the accuracies of other sleep stages (91.2–98.49%). Tables 4–7 show that combination of multiple classifiers improves the accuracies of all the sleep stages, where the accuracy was greater than 80.05%. The performance of classifier combination is significantly greater than that of an individual classifier.

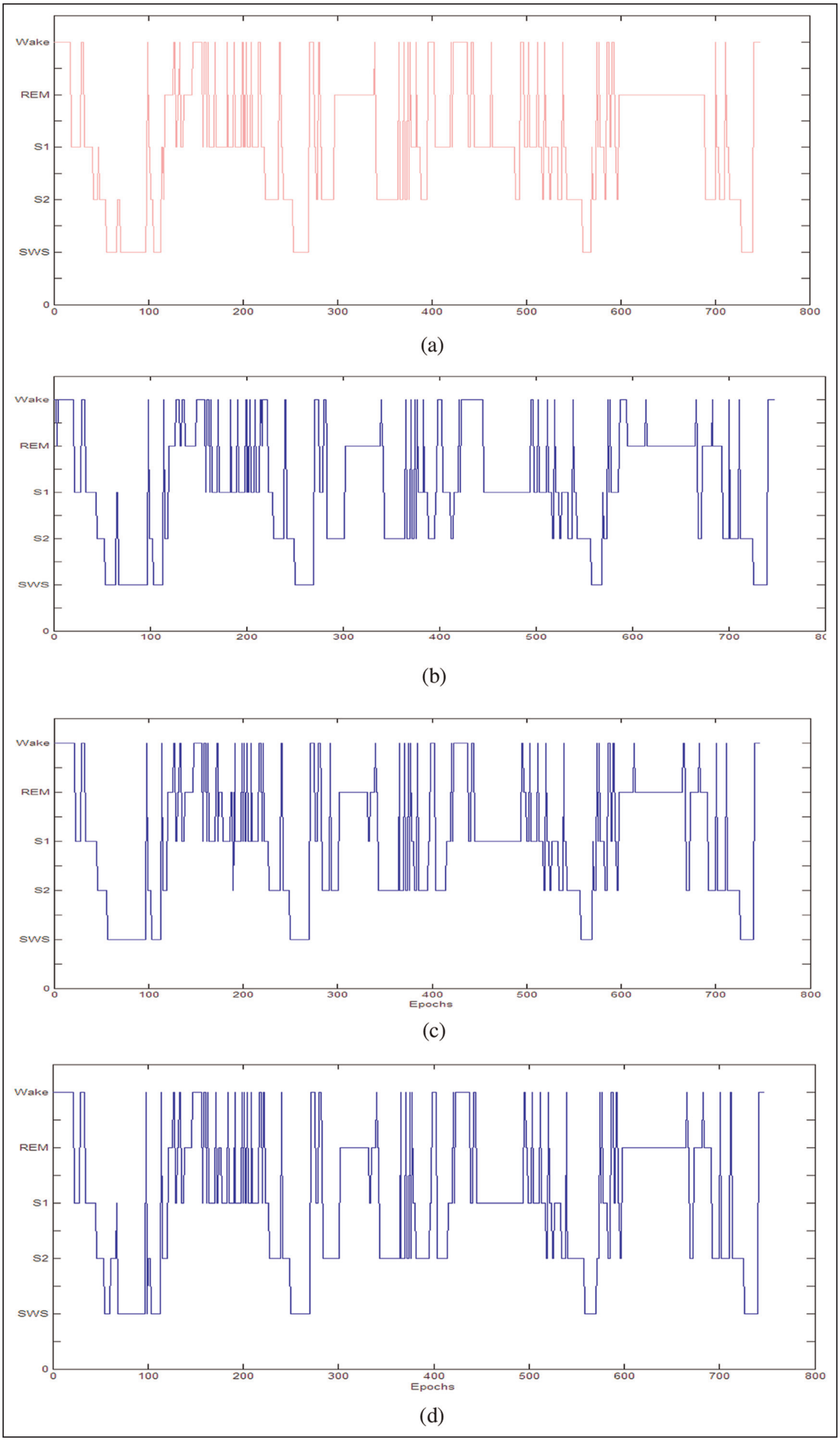
Table 8 shows the F_1 score of each sleep stage and the total accuracy of each classifier and classifier combination. Comparing the F_1 score for individual sleep stage, the classifier combination performs better than the other three classifiers: 1) the F_1 scores of the SWS, S2, S1, REM and Wake with classifier combination were 0.97–4.22%, 9.26–13.9%, 9.63–14.96%, 2.3–3.26% and 2.89–5.67% higher than that of single classifier, respectively; 2) the total accuracies of SVM, KNN and HMM were similar, while the performance of classifier combination was 5.76–9.26% higher than those of individual classifier.

As an illustration example, Figure 3(a) shows the hypnograms of a subject that was scored by the expert, (b)–(d) performed by individual classifier and (e) generated by classifier combination without using any handcrafted features. Compared with the hypnogram of sleep stages, we can see that the five hypnograms were different. The best classification performance of the selected subject using the classifier combination was 91.31%, with accuracy of 83.98% for the SVM, 82.05% for the KNN and 85.55% for the HMM.

Learned features

According to the R&K rules, sleep stage classification are typical pattern recognition task. Firstly, physicians look at the polysomnogram and then recognize the features of signals, such as alpha, beta, theta, spindle and so on. Lastly, physicians classify successive epochs from the features. Feature extraction from the polysomnogram is very important for sleep stage classification. In order to improve the performance of the classification, many features, such as time, non-linearity, frequency and entropy have been used. However, all these features are handcrafted features, which require expert knowledge. Our purpose in this paper was to extract features automatically from the raw sleep data and then classify based on the learned features.

Ito and Komatsu (2004) have pointed out that angles and junctions embedded within contours are important features to represent the shape of objects. In natural scenes, angles are presented as part of the contours of the objects and not as isolated visual stimuli. It therefore seems reasonable to take into account contextual modulation by contour lines to which angles are connected. In addition, from the point of view of Hubel and Wiesel (1959, 1968), at area V1, the earliest visual stage, many neurons are highly selective for the orientation of line segments. At area V2, a higher visual stage, many neurons are also selective for line segment orientation; moreover, Kobatake and Tanaka (1994) demonstrated that a few V2 neurons selectively respond to complex stimuli, including a sharp triangle, and suggested that the representation of complex stimulus features may begin to emerge in area V2. Many V2 neurons show angle selectivity that is dependent on the



(a)

(b)

(c)

(d)

Figure 3. Sleep stage classification results of the 002 subject: (a) Scored by the expert; (b) support vector machine (SVM) classifier; (c) K-nearest neighbour (KNN) classifier; (d) hidden Markov model (HMM) classifier; (e) classifier combination.

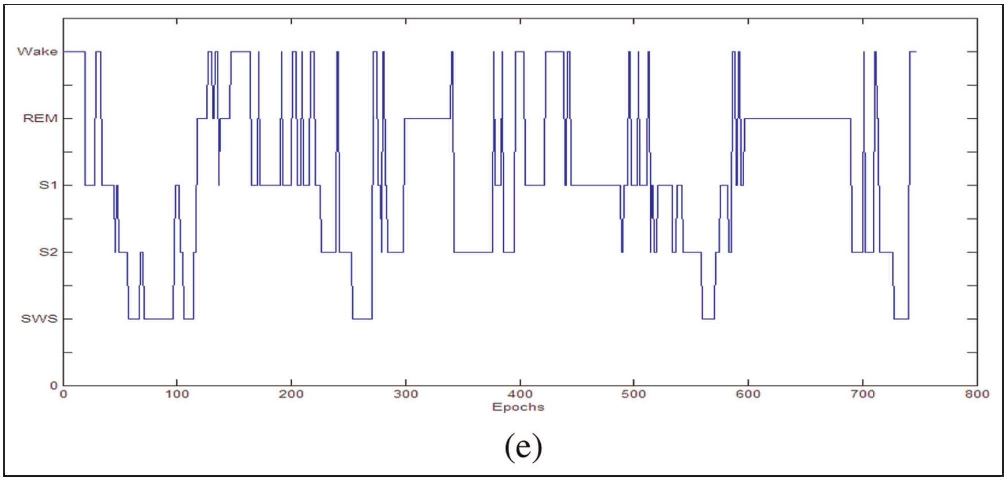


Figure 3. (Continued)

responses to the individual line components of the angle stimuli as well as to the combination of two line components (Ito and Komatsu, 2004). By making several axial measurements within the profile, Ito and Komatsu were able to compute various statistics about each neuron's selectivity for angle width, angle orientation and for each separate line component of the angle. Almost 80% of the neurons responded to specific angle stimuli, and several neurons exhibited a high amount of selectivity for its peak angle producing angle profiles. Inspiration from the work of van Hateren and van der Schaaf (1998), Ito and Komatsu (2004) and Lee et al. (2007) proposed an unsupervised learning model named SDBN, which faithfully mimics certain properties of visual area V2.

In order to visualize what the units in the first hidden layer are responding to, Figure 4–8 shows some features that SDBN has learned from EEG, EOG, and EMG. These figures show the learned features for different sleep stages are different from each other largely. The details of W stage features were shown in Figure 4. For example, the features of alpha rhythm, eye blink, rapid-rolling eye movement and higher amplitude EMG could be observed.

Figure 5 shows the features of REM stage, which are different from those of W stage. The obvious differences are that the amplitude of EEG and EMG feature with respect to REM are lower than that of the W stage, which is in accordance with the R&K rules. In addition, the feature of saw-tooth, which is the typical characteristic of the REM stage, was learned. Although the features of the alpha rhythm could be seen, the frequency of alpha rhythm is slower than that of the W stage.

The features of the S1 stage are shown in Figure 6. Some features of relatively low voltage mixed frequency waves, which are similar to those of REM stage, could be seen. Moreover, the feature of vertex incipient K-complexes is also displayed in Figure 6(a). Figure 6(b) shows the amplitude of EMG, which is similar to that of REM, but remarkably lower than that of the W stage. From Figure 6(c), we can draw the conclusion that the EOG activity is mainly slow-rolling eye movement, which is different from that of the W and REM stages. The features of EEG activity of the W stage and REM are not very obvious. The EEG activity features of S2 and SWS, however, are remarkable.

In Figure 7(a), the features of sleep spindle and K complex wave may be seen sharply, which are the primary characters of the S2 stage. Additionally, the features of some relatively low voltage mixed frequency waves are displayed. The frequencies of the relatively low voltage mixed frequency waves, however, are lower than those of S1 stage and REM stage. From Figure 7(b) and (c), we can see the features of EMG and EOG activities are obviously less than those of the S1 stage.

The feature of the delta wave, which is the main feature of the SWS stage, can be notably seen from Figure 8(a). The EMG and EOG activities, which are shown by Figure 8(b) and (c), are less than those of any other stages. All these learned features are useful for automatic classification, which also demonstrate that SDBN is able to extract related features with respect to different sleep stages.

Because the second hidden layer concatenates all the features of the first hidden layer, the features of the second hidden layer are very abstract. We only analyse the contour features of EEG, EMG and EOG in the first hidden layer. Nevertheless, there are still some features in which we do not realize the meaning of their representations, especially the EEG features. Maybe some of these features are non-linear, entropy or other categories those cannot be analysed from only the aspect of contour.

Comparison

Many traditional feature extraction methods, such as principle component analysis (PCA) and independent component analysis (ICA) are shallow architectures. Human information processing mechanisms (e.g. vision and speech), however, suggest the need for deep architectures for extracting complex structure and building internal representation from rich sensory inputs (Yu and Deng, 2011). Deep architectures attempt to learn hierarchical structure, and hold the promise of being

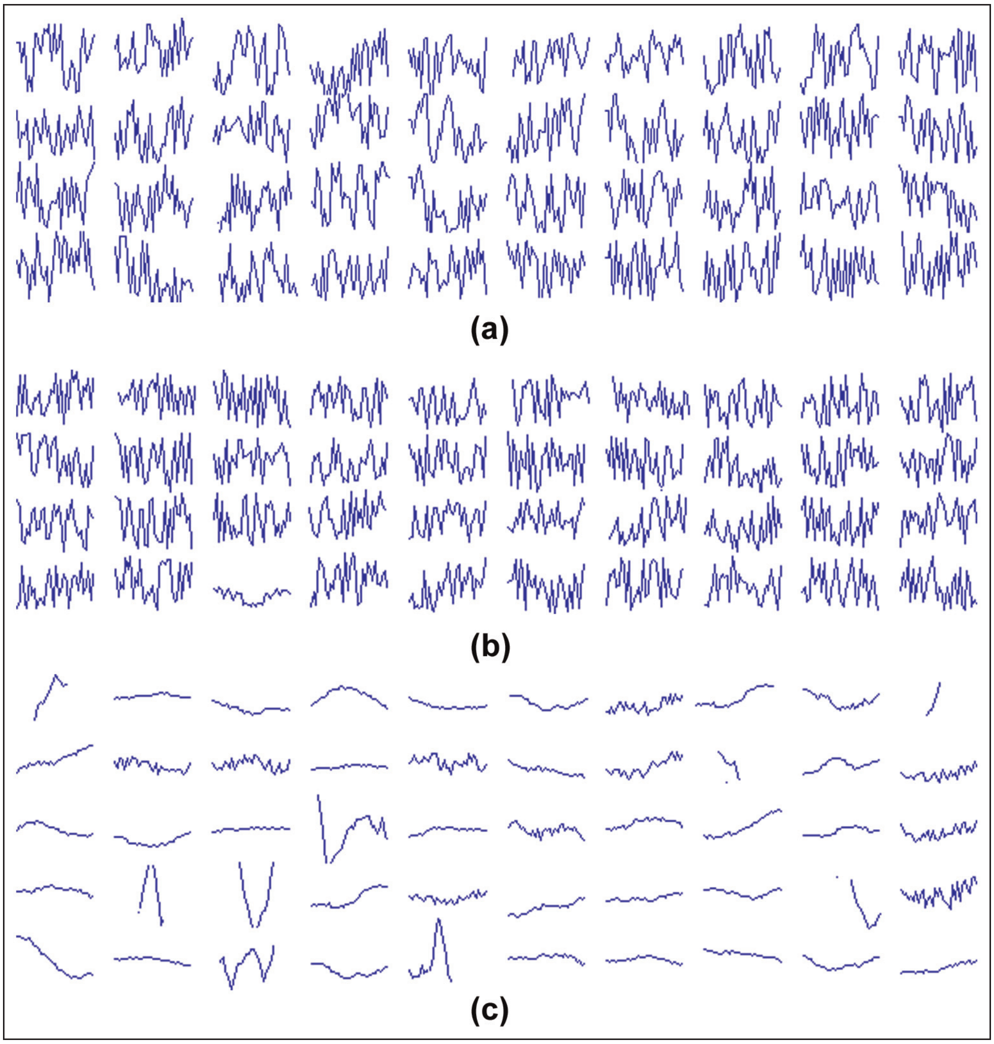


Figure 4. Some learned features with W stage at the first hidden layers: (a) electroencephalogram (EEG) features; (b) electromyogram (EMG) features; (c) electro-oculogram (EOG) features.

able to learn simple features firstly, and then build up more complex features successfully by composing together the simpler ones. Many deep learning models (Bengio et al., 2006; Hinton et al., 2006; Ranzato et al., 2007) have been proposed based on deep architectures, and some studies compared these models with the response properties of neurons in area V1. Compared with SDBN, these models have not directly evaluated the higher-order features in the deeper areas of cortical hierarchy, such as visual areas V2. In contrast, Lee et al. (2007) have analysed SDBN in visual area V2, and then pointed out that the SDBN picks up not only collinear ('contour') features but also corners and junctions. His study also suggested that the encoding of these more complex 'corner' features matches well with the results from the Ito and Komatsu's study of biological V2 responses. Additionally, sparse restricted Boltzmann machines (RBMs) can be seen as a model of the mirror neuron system (MNS) (Marijnissen, 2011). Rizzolatti and Craighero (2004) pointed that MNS can respond to multiple features and fire as soon as the perception contains enough clues to infer an action.

Compared with SDBN, under the limitation number of sleep signals, the ICA methods can hardly obtain the good splitting performance. In addition, ICA is sensitive to whitening, and scales poorly to large sets of features or large inputs (Lee et al., 2007). Compared with PCA, SDBN is more suitable for processing sleep data that is multimodal, temporal asymmetry, non-linear, non-stationary and random. Additionally, for overcomplete features, PCA codes do not generally satisfy high dispersal, as the codes that correspond to the largest eigenvalues are almost always active (Ngiam et al., 2011). From the first five experiments of Table 9, we can see the blindness of handcrafted feature selection. The accuracy of the sixth experiment with HMM (Langkvist et al., 2012), which consists of feature extraction, feature selection, PCA, A Gaussian mixture model (GMM) and HMM (Feat-GOHMM), is only 63.4%. When HMM instead

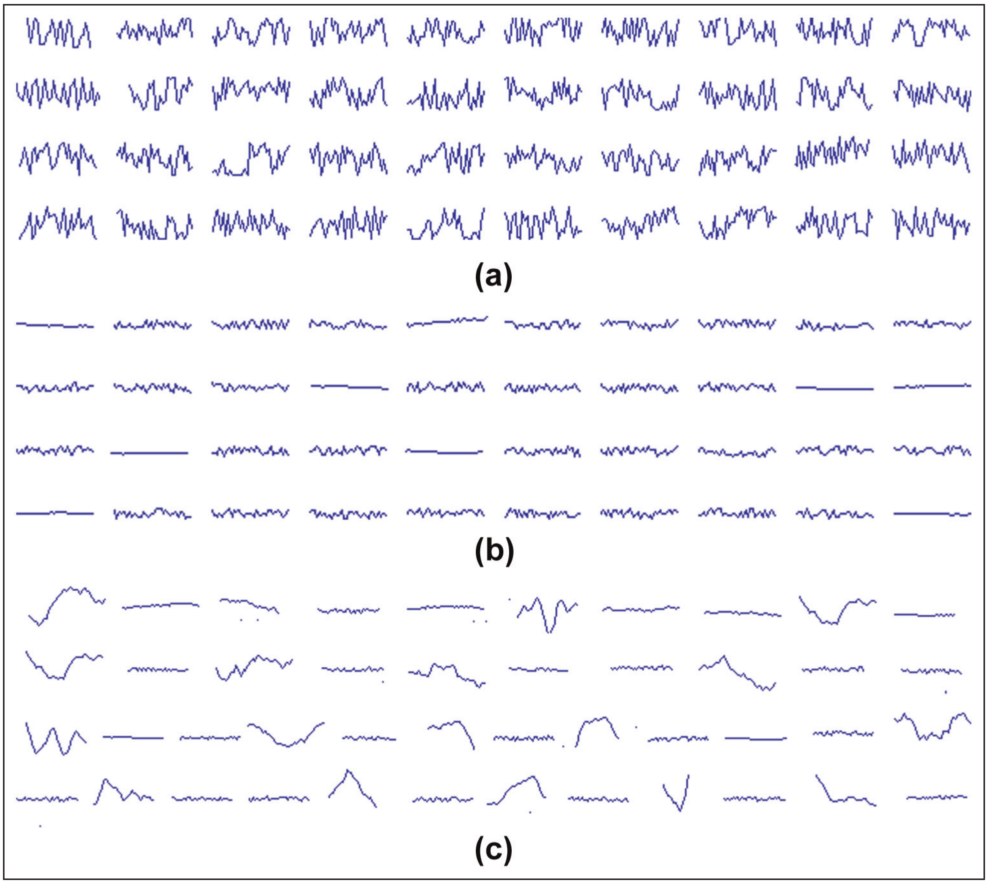


Figure 5. Some learned features with REM stage at the first hidden layers: (a) electroencephalogram (EEG) features; (b) electromyogram (EMG) features; (c) electro-oculogram (EOG) features.

of combination of multiple classifiers is used, the accuracy is only 72.68%. The reason maybe that the sleep data contains much noise and it is multimodal and non-stationary, which is not suitable for other existing feature extraction methods.

We also noticed that the multiple architectures and training method of SDBN lead to a high running time. The time complexity is the problem of deep models. As the deep learning aim is to solve increasingly challenging tasks, models of greater complexity are required. This in turn requires orders of magnitude more data to take advantage of these powerful models while avoiding overfitting (Mathieu et al., 2013). The complex architecture, more parameters and unsupervised training way of SDBN make the time complexity much higher than those of previous models. For comparison, the sixth experiment was performed. The training time for Feat-GOHMM and our method were approximately 0.83 and 5.3512 h, respectively, and the testing time for Feat-GOHMM and our method were approximately 1.1378 and 5.2703 min, respectively. However, with optimization, parallel computing and the use of cheap GPU, the time complexity will certainly decrease (Langkvist et al., 2012; Mathieu et al., 2013).

In order to demonstrate the usefulness of our system, we implemented seven experiments, which were also based on leave-one-out cross-validation of the 10 recordings. We compared the seven different results that were based on different sets of handcrafted features with our proposed method. The performance comparison was summarized in Table 9. From Table 9, we can draw the conclusions as follows:

1) Because there are many features and there exists no effective method to measure the importance of each feature, different sets of features were chosen to identify sleep stages, and then the results were also different. Although one category feature-energy of alpha, beta, theta, delta and spindle was used, the accuracy of classification was 84.8% (Hsu et al., 2013). When we used the methods of Agarwal and Gotman (2001) and Khalighi et al. (2012), which contained multi-category features, the accuracies of classification were 75.16% and 72.7%, respectively, which also demonstrate the blindness of handcrafted feature selection.
2) By using a combination of multiple classifiers, the first seven experimental results were improved 6.15%, 8.95–10.92%, 1.73%, 0.53%, 2.3–4%, 9.28% and 8.37%, respectively. The results also indicated that our proposed principle of voting is an effective method. The combination of multiple classifiers increased range also varied with respect to different sets of features.

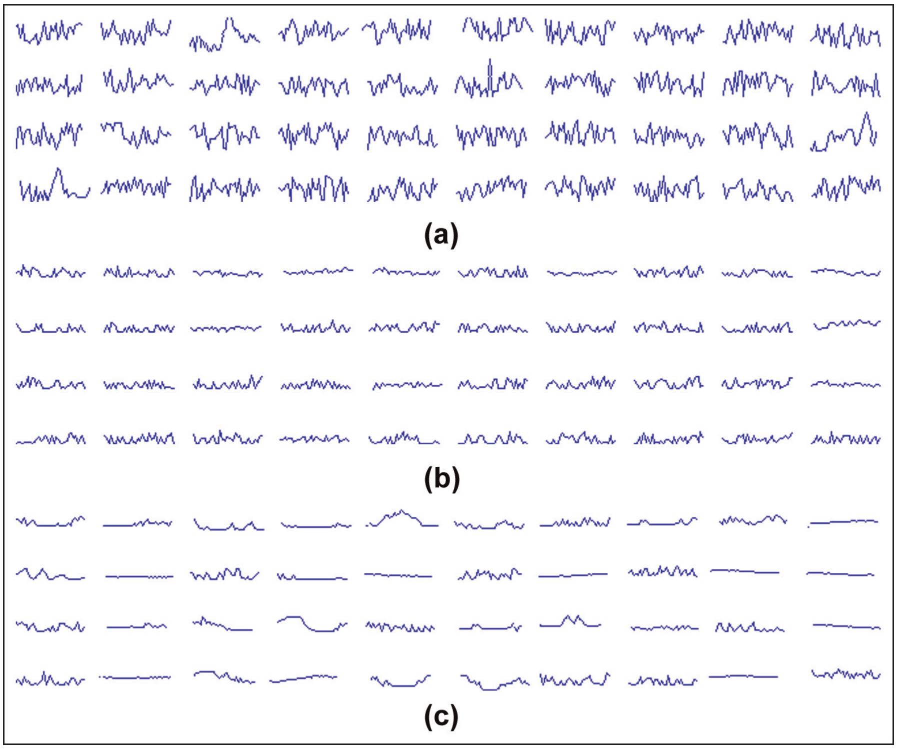


Figure 6. Some learned features with S1 stage at the first hidden layers: (a) electroencephalogram (EEG) features; (b) electromyogram (EMG) features; (c) electro-oculogram (EOG) features.

3) The accuracy of our proposed method was 91.33%, which is larger than other experiment results. The main point is that we do not care which feature should be selected and which feature is more important.
4) From the seventh experiment, we can see that the performance of our proposed method is much better than that of Langkvist et al. (2012). The reasons are that DBN tends to learn distributed, non-sparse representations. However, SDBN enables deep belief nets to learn sparse representations, and DBN might seem unrelated to the selectivity of a unit, but SDBN can guarantee the selectivity.

The results show that our method may replace traditional methods for sleep stages. It is essential to state that our approach, which learned features from raw sleep data, is a real automatic sleep stage system. Almost all the other methods are based on handcrafted features, which requires designers to have domain knowledge and sometimes they are not sure which are to be used among those handcrafted features. Our method does not need any domain knowledge and extracts useful features automatically. Additionally, the SDBN not only enables the discovery of new useful feature representations that a human expert might not be aware of, but also presents a way of exploiting massive amounts of unlabeled data, which is very helpful for home sleep monitoring.

Discussion

In this paper, SDBN was the first time to be applied to sleep data for unsupervised feature extraction and a new voting principle based on classification entropy was developed. From Table 8, we can see that the total accuracies of SVM, KNN and HMM were 83.98%, 82.05% and 85.55%, respectively, which indicated that some appropriate features for sleep stages were extracted by SDBN. Although the total accuracy of individual classifier was higher than 80%, the accuracy for each sleep stage was not all larger than 80%.

There are two likely reasons for the low accuracy of S1. The first is that S1 is a transition from wakefulness to other sleep stages, especially S2, and the transition between two successive stage phases may occur during an epoch or may last longer than the 20-s period during which it is difficult for the expert to make a decision (Hsu et al, 2013; Zoubek et al., 2007). At the same time Figures 7 and 8, we show that some features of EMG and EOG with S1 resemble those of S2, which makes distinguishing completely between S1 and S2 difficult. For example, Table 4 shows that 21% of the S1 is classified as S2; Table 5 indicates that 9.56% of the S1 is classified as S2; Table 6 shows that 5.29% of the S1 is classified as S2.

From the viewpoint of machine learning, different classifiers may have different results in the same features. From Tables 4–6, we can see that the classification errors of S1 using SVM, KNN and HMM were 37.91%, 34.48% and 30.24%, respectively. Considering the fact that the misclassified

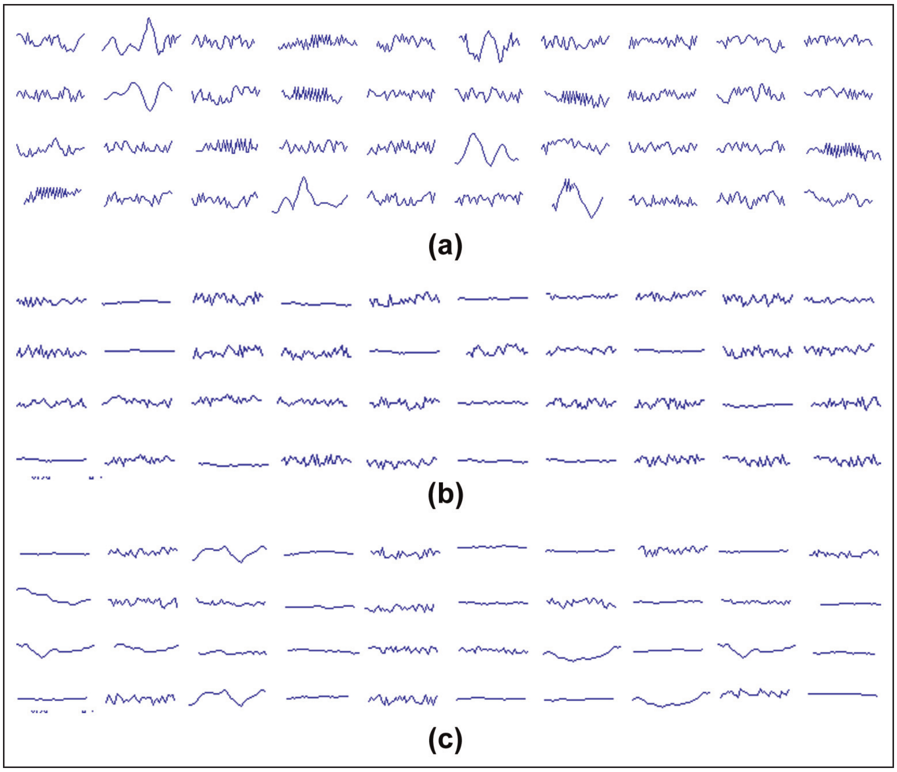


Figure 7. Some learned features with S2 stage at the first hidden layers: (a) electroencephalogram (EEG) features; (b) electromyogram (EMG) features; (c) electro-oculogram (EOG) features.

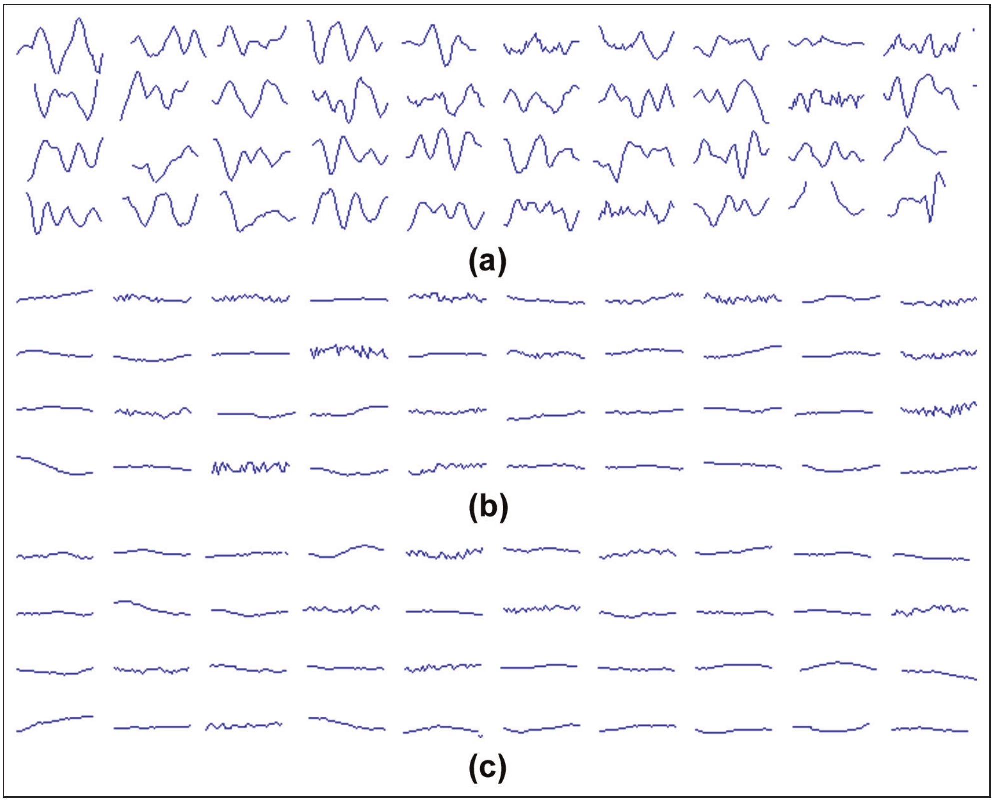


Figure 8. Some learned features with SWS stage at the first hidden layers: (a) electroencephalogram (EEG) features; (b) electromyogram (EMG) features; (c) electro-oculogram (EOG) features.

Table 9. Classification performance comparison of our system with six existing sets of features.

Inputs	Features	Classifiers	Total accuracy
EEG, EOG, EMG (Agarwal and Gotman, 2001).	Amplitude, dominant frequency, spindles, alpha-slow-wave index, theta-slow-wave index, eye movements	Clustering Multiple classifiers combination	75.16% 81.31%
EEG, EOG, EMG (Khalighi et al., 2012)	Energy, percentage of energy, mean, standard deviation, relative spectral power, harmonic parameters, percentile 25, 50, 75, skewness	Kernel logic regression SVM Multiple classifiers combination	72.7% 74.67% 83.62%
EEG, EOG, MG (Álvarez-Estévez et al., 2013)	Amplitude (EOG and EMG signals), frequencies of alpha, beta and theta signals, Short time Fourier transform method, power spectral density, sleep spindles, K-complexes	Fuzzy logic Multiple classifiers combination	89.55% 91.28%
EEG (Hsu et al., 2013)	The energy of alpha, beta, theta, delta, spindle	Recurrent neural netwok Multiple classifiers combination	84.8% 85.33%
EEG (Gudmundsson et al., 2005)	Hjorth parameters, power spectrum, relative power, median frequency, spectral entropy, amplitude and frequency distribution	SVM K-NN Multiple classifiers combination	80% 78.3% 82.3%
EEG, EOG, EMG	28 handcrafted features[a] (Langkvist et al., 2012)	HMM Multiple classifiers combination	63.4% 72.68%
EEG, EOG, EMG	Unsupervised feature learning based on DBN (Langkvist et al., 2012)	HMM Multiple classifiers combination	66% 74.37%
EEG, EOG, EMG	Unsupervised feature learning based on SDBN	Multiple classifiers combination	91.33%

[a] The 28 features listed by Langkvist et al. (2012).
For abbreviations, see text.

instances of individual classifiers do not necessarily overlap, different classifiers may perform differently in decision making. Hence, making use of the complementary information by combination of multiple classifiers could further improve the total performance of sleep stages. From Table 7, it can be seen that the classification errors of S1 using classifier combination was only 19.95% and 8.03% of the S1 was misclassified as S2. Although the classification error of S1 with classifier combination is less than that of individual classifier, it is still greater than the classification error of other sleep phases using classifier combination. The review of relevant research shows that the reasons include: 1) the EEG features of REM are similar to those of S1, which also could be seen from Figures 5(a)–6(a); 2) with the progress of S1, the slow-rolling eye movement will decrease or disappear; 3) the REM phase consists of tonic REM sleep and phasic REM sleep, and rapid eye movement could be seen during the phasic REM sleep, while the eye movement will disappear during tonic REM sleep; 4) because the activity of EMG is highly individual during S1, it may highly decrease or maintain the level of the Wake phase; 5) the activity of EMG during REM is too low or may even disappear gradually, but during muscle twitching during tonic REM sleep or phasic REM sleep, the patient's EMG activity will be elevated. Hence, when the features EEG, EOG and EMG are quite similar to those between adjacent phases, it is difficult for machine to discriminate S1 from REM. From Tables 4–7, we can see that the classification errors of S1 as REM were 12.24%, 18.03%, 16.79% and 9.54%, respectively.

However, the accuracies of S2, REM, SWS and Wake with individual classifier were larger than 76.31%, 91.2%, 94.58% and 96.25%, respectively. The performance of classifier combination with S2, REM, SWS and Wake were 91.2%, 95.31%, 98.22% and 98.49%, respectively. One reason for the excellent classification of S2 by classifier combination was that the misclassified instances of individual classifier do not overlap, and different classifiers may perform differently in decision making. As the accuracies of REM, SWS and Wake with single classifier are larger than 91%, the classifier combination improves the percentage accuracies of these three phases by a few digits only. There are two reasons for this situation. The first reason is that the appropriate features were extracted by SDBN based on EEG, EOG and EMG, which is in agreement with the R&K rules and can improve the classification of sleep stages. The second reason is that S3 and S4 were combined into SWS, which was able to decrease misclassification caused by transitions between S3 and S4.

Conclusions

In this paper, SDBN and combination of multiple classifiers were applied to analyse the multimodal sleep data for the first time. The results showed that SDBN was capable of learning useful features in an unsupervised fashion and classifier combination can obtain excellent classification performance comparable with individual classifier. The advantage of our approach was efficient and fully automated, and the method can be easily adapted to other physiological signal analysis and prediction problems. In future, we will improve the discrimination of S1. Indeed, the classification accuracy of S1 is lower compared with those of the other sleep stages. Moreover, real-time sleep stages detection will also be our future work.

Declaration of conflicting interest

The authors declare that there is no conflict of interest.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Notes

1. The performance of a classifier about different classes.
2. The performance of a classifier about the same class.

References

Acharya UR, Chua ECP, Chua KC, et al. (2010) Analysis and automatic identification of sleep stages using higher order spectra. *International Journal of Neural Systems* 20: 509–521.

Agarwal R and Gotman J (2001) Computer-assisted sleep staging. *IEEE Transactions on Biomedical Engineering* 48: 1412–1423.

Álvarez-Estévez D, Fernández-Pastoriza JM, Hernández-Pereira E, et al. (2013) A method for the automatic analysis of the sleep macrostructure in continuum. *Expert Systems with Applications* 40: 1796–1803.

Bengio Y, Lamvlin P, Popovici D, et al. (2006) Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems (NIPS 2006)*, pp. 153–160.

Chang C-C and Lin C-J (2011) LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2: 1–27.

Charbonnier S, Zoubek L, Lesecq S, et al. (2011) Self-evaluated automatic classifier as a decision-support tool for sleep/wake staging. *Computers in Biology and Medicine* 41: 380–389.

Crisler S, Morrissey MJ, Anch AM, et al. (2008) Sleep-stage scoring in the rat using a support vector machine. *Journal of Neuroscience Methods* 16: 524–534.

Gelfand AE (2000) Gibbs sampling. *Journal of the American Statistical Association* 95: 1300–1304.

Gudmundsson S, Runarsson TP and Sigurdsson S (2005) Automatic sleep staging using support vector machines with posterior probability estimates. In: *International Conference on Computational Intelligence for Modelling, Control and Automation, 2005, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, IEEE 2: 366–372.

Guyon I, Weston J, Barnhill S, et al. (2002) Gene selection for cancer classification using support vector machines. *Machine Learning* 46: 389–422.

Hamida S and Ahmed B (2013) Computer based sleep staging: challenges for the future. *7th IEEE GCC Conference and Exhibition*, pp. 280–285.

Hinton GE, Osindero S and Teh YW (2006) A fast learning algorithm for deep belief nets. *Neural Computation* 18: 1527–1554.

Hinton GE (2002) Training products of experts by minimizing contrastive divergence. *Neural Computation* 14: 1771–1800.

Hsu C, Chang C and Lin C (2003) *A practical guide to support vector classification*. Technical report, Department of Computer Science, National Taiwan University. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Hsu YL, Yang YT, Wang JS, et al. (2013) Automatic sleep stage recurrent neural classifier using energy features of EEG signals. *Neurocomputing* 104: 105–114.

Hubel DH and Wiesel TN (1959) Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology (London)* 148: 574–591.

Hubel DH and Wiesel TN (1968) Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)* 195: 215–243.

Ito M and Komatsu H (2004) Representation of angles embedded within contour stimuli in area v2 of macaque monkeys. *The Journal of Neuroscience* 24: 3313–3324.

Jha SK, Imahashi M, Hayashi K, et al. (2014) Data fusion approach for human body odor discrimination using GC-MS spectra. *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 1–6.

Kannathal N, Acharya UR, Lim CM, et al. (2005) Characterization of EEG – a comparative study. *Computer Methods and Programs in Biomedicine* 80: 17–23.

Kanoje BK and Shingare AS (2014) Automatic sleep stage detection of an EEG signal using an ensemble method. *International Journal of Advanced Research in Computer Engineering & Technology* 8: 2717–2724.

Khalighi S, Sousa T and Nunes U (2012) Adaptive automatic sleep stage classification under covariate shift. *2012 Annual International Conference of the IEEE*, pp. 2259–2262.

Kobatake E and Tanaka K (1994) Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. *Journal of Neurophysiology* 71: 856–867.

Koley B and Dey D (2012) An ensemble system for automatic sleep stage classification using single channel EEG signal. *Computers in Biology and Medicine* 42: 1186–1195.

Krakovská A and Mezeiová K (2011) Automatic sleep scoring: a search for an optimal combination of measures. *Artificial Intelligence in Medicine* 53: 25–33.

Kurihara Y and Watanabe K (2012) Sleep-stage decision algorithm by using heartbeat and body-movement signals. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 42: 1450–1459.

Langkvist M, Karlsson L and Loutfi A (2012) Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems* 2012: 5.

Lecun Y, Bottou L, Bengio Y, et al. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86: 2278–2324.

Lee H, Ekanadham C and Ng AY (2007) Sparse deep belief net model for visual area V2. *Advances in Neural Information Processing Systems (NIPS 2007)* 7: 873–880.

Lee JE and Yoo SK (2013) Electroencephalography analysis using neural network and support vector machine during sleep. *Engineering* 5: 88.

Liang SF, Kuo CE, Hu YH, et al. (2012) Automatic stage scoring of single-channel sleep EEG by using multiscale entropy and autoregressive models. *IEEE Transactions on Instrumentation and Measurement* 61: 1649–1657.

Marijnissen M (2011) *Sparse Restricted Boltzmann Machines as a model of the Mirror Neuron System*. Nijmegen, Netherlands: Radboud University.

Mathieu M, Henaff M and LeCun Y (2013) Fast training of convolutional networks through FFTs. *CoRR*, abs/1312.5851.

Melgani F and Bruzzone L (2004) Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing* 42: 1778–1790.

Ngiam J, Chen Z, Bhaskar SA, et al. (2011) Sparse filtering. *Advances in Neural Information Processing Systems* (*NIPS* 2011) 1125–1133.

Penzel T and Conradt R (2000) Computer based sleep recording and analysis. *Sleep Medicine Reviews* 4: 131–148.

Rabiner L (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE* 77: 257–286.

Rabiner L and Juang B (1986) An introduction of hidden Markov models. *IEEE ASSP Magazine* 3: 4–16.

Ranzato M, Poultney C, Chopra S, et al. (2007) Efficient learning of sparse representations with an energy-based model. In: Platt J, et al, *Efficient Learning of Sparse Representations with an Energy-Based Model, Advances in Neural Information Processing Systems* (*NIPS* 2006).

Rechtschaffen A and Kales A (1968) *A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects*. Washington, DC: Government Printing Office, Public Health Service, pp. 3–7.

Rizzolatti G and Craighero L (2004) The mirror-neuron system. *Annual Review of Neuroscience* 27:169–192.

Ronzhina M, Janoušek O, Kolářová J, et al. (2012) Sleep scoring using artificial neural networks. *Sleep Medicine Reviews* 16: 251–263.

Stanus E, Lacroix B, Kerkhofs M, et al. (1987) Automated sleep scoring: a comparative reliability study of two algorithms. *Electroencephalography and Clinical Neurophysiology* 66: 448–456.

Schulz H (2008) Rethinking sleep analysis: comment on the AASM manual for the scoring of sleep and associated events. *Journal of Clinical Sleep Medicine* 4: 99.

van Hateren JH and van der Schaaf A (1998) Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 265: 359–366.

Wang D and Shang Y (2013) Modeling physiological data with deep belief networks. *International Journal of Information and Education Technology*, 3.

Wang Z, Lyu S, Schalk G, et al. (2013) Deep feature learning using target priors with applications in ECoG signal decoding for BCI. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, pp. 1785–1791.

Wulsin D, Blanco J, Mani R, et al. (2010) Semi-supervised anomaly detection for EEG waveforms using deep belief nets. *International Conference on Machine Learning and Applications*. IEEE, pp. 436–441.

Wulsin DF, Gupta JR, Mani R, et al. (2011) Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement. *Journal of Neural Engineering* 8: 53–57.

Xie B and Minn H (2012) Real-time sleep apnea detection by classifier combination. *IEEE Transactions on Information Technology in Biomedicine* 16: 469–477.

Yιlmaz B, Asyalι M H, Arιkan E, et al. (2010) Sleep stage and obstructive apneaic epoch classification using single-lead ECG. *Biomedical Engineering Online* 9: 39.

Yu D and Deng L (2011) Deep learning and its applications to signal and information processing. *IEEE Signal Processing Magazine* 28: 145–154.

Zoubek L, Charbonnier S, Lesecq S, et al. (2007) Feature selection for sleep/wake stages classification using data driven methods. *Biomedical Signal Processing and Control* 2: 171–179.