Computer Methods and Programs in Biomedicine 180 (2019) 105001

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Deep learning approaches for automatic detection of sleep apnea events from an electrocardiogram

Urtnasan Erdenebayar[a], Yoon Ji Kim[a], Jong-Uk Park[a], Eun Yeon Joo[b], Kyoung-Joung Lee[a,*]

[a] Department of Biomedical Engineering, College of Health Science, Yonsei University, Wonju 26493, Korea
[b] Department of Neurology, Samsung Medical Center, School of Medicine, Sungkyunkwan University, Korea

ARTICLE INFO

Article history:
Received 5 March 2019
Revised 20 July 2019
Accepted 29 July 2019

Keywords:
Sleep apnea
Deep learning
Convolutional neural network
Recurrent neural network
Long short-term memory
Gated-recurrent unit

ABSTRACT

Background and Objective: This study demonstrates deep learning approaches with an aim to find the optimal method to automatically detect sleep apnea (SA) events from an electrocardiogram (ECG) signal.
Methods: Six deep learning approaches were designed and implemented for automatic detection of SA events including deep neural network (DNN), one-dimensional (1D) convolutional neural networks (CNN), two-dimensional (2D) CNN, recurrent neural networks (RNN), long short-term memory, and gated-recurrent unit (GRU). Designed deep learning models were analyzed and compared in the performances. The ECG signal was pre-processed, normalized, and segmented into 10 s intervals. Subsequently, the signal was converted into a 2D form for analysis in the 2D CNN model. A dataset collected from 86 patients with SA was used. The training set comprised data from 69 of the patients, while the test set contained data from the remaining 17 patients.
Results: The accuracy of the best-performing model was 99.0%, and the 1D CNN and GRU models had 99.0% recall rates.
Conclusions: The designed deep learning approaches performed better than those developed and tested in previous studies in terms of detecting SA events, and they could distinguish between apnea and hypopnea events using an ECG signal. The deep learning approaches such as 1D CNN and GRU can be helpful tools to automatically detect SA in sleep apnea screening and related studies.

1. Introduction

Sleep apnea (SA) is an abnormal respiratory pattern occurring during sleep. It includes apnea and hypopnea and is caused by recurrent episodes of reduced or absent respiratory airflow caused by upper airway collapse or other airway obstruction [1]. The likelihood of developing SA is positively correlated with age [2]. SA itself may increase the risk of heart disease [3], diabetes [4], chronic kidney disease [5], stroke [6], depression [7], and cognitive impairment [8]. Generally, SA screening and diagnostic methods require that various physiological signals be recorded by polysomnography (PSG) during overnight sleep in sleep centers [9]. These methods are labor-intensive, time-consuming, come at high cost, and are inconvenient for patients. Moreover, it is also difficult to manually annotate PSG recordings.

To replace PSG, many simple and minimized methods have been proposed to detect SA. These methods are based on physiological recordings such as a single-lead ECG [10-16], SpO_2 [17-21], snoring signals [22-24], and respiratory signals [25-27]. The respiratory and snoring are directly effected by SA and upper airway obstruction. The ECG and SpO2 are indirectly effected by SA events because the responses to SA include both an increase in the symphathetic and the parasymphathetic tone to the heart rate and systolic blood pressure [28]. These physiological changes of the ECG signal during SA can be noticed by the time, frequency, and nonlinear domain analysis. Therefore, most alternative studies were focused on extracting the temporal, spectral and nonlinear features from the physiological signals, and various methods of feature selection including principal component analysis, statistical evaluation, and wrapper methods to reduce the dimension of the feature space. Many types of supervised learning methods have been employed in those extracted features to improve the performance of SA detection. In particular, the support vector machine [13,23] and neural networks [15,29,30] were used widely for SA detection, as well as the k-nearest neighbor [11,12], linear/quadratic discriminant analysis [18], AdaBoost [17], and fuzzy logic [31]. All these studies can be described within canonical supervised learning that composed data processing, feature extraction, feature selection, and classification. In supervised learning, the discrimination power of

* Corresponding author.
E-mail address: lkj5809@yonsei.ac.kr (K.-J. Lee).

https://doi.org/10.1016/j.cmpb.2019.105001
0169-2607/

feature is significant, but it can require domain knowledge, labor-intensive, and can be limited for complex data [32].

Recently, convolutional neural network (CNN) and recurrent neural network (RNN) systems have become increasingly popular. Although these represent supervised learning systems, they perform an excellent performance across a wide range of applications. CNN has become an invaluable technology in the field of image signal processing and computer vision [33], while RNN performs well in speech signal processing and speech recognition fields [34,35]. Few sleep studies have applied deep neural network (DNN), CNNs and RNNs to automatically detect SA [36-38]. They have achieved higher performances for the automatic detection of sleep-breathing disorder events, including apnea and hypopnea, than conventional machine learning methods. However, they did not eliminated processes in the complex signal analysis and the hand-crafted feature extraction. In addition, they did not deliver the appropriate function of the deep learning approach, which is automatic detection of SA events by using ECG.

In this study, we demonstrated the comprehensive analysis of the representative deep learning approaches that were optimally designed for automatic detection of SA using an ECG signal. Deep learning approach consists of the six different models including DNN, CNN, and RNN-based methods. The CNN-based model consists of the 1D CNN and 2D CNN which were designed and optimized by employing 1D and 2D convolutions, respectively. 1D CNN was designed for the time domain characteristics of the ECG signal whereas 2D CNN model was intended for the spectral components of the ECG signal during the SA events. Because of the SA events are occurring in the sequentially and repeatedly during sleep, long short-term memory (LSTM) and gated-recurrent unit (GRU) models were used to construct the RNN-based approach. Finally, we designed two basic models such as DNN and vanilla RNN model for comparison purposes. The clinical dataset of the SA patients was used to the training and testing each model, and to compare strengths and weaknesses for each constructed deep learning models.

2. Material and method

2.1. Subjects and datasets

For this study, nocturnal PSG recordings from 86 patients with SA were analyzed (Table 1). PSG recordings were measured by an Embla N7000 amplifier device (Embla System Inc., U.S.A.) at the Samsung Medical Center (Seoul, Korea). Accordance with the AASM guidelines [39] the PSG recording was labeled by an experienced sleep technician. The institutional review board of the Samsung Medical Center (IRB:2012-01063) authorized this study protocol. All patients that enrolled in this study provide written consent.

The SA datasets comprised normal breathing and SA events, including hypopnea (H) and apnea (A). Moreover, SA datasets composed of a balanced number of events were randomly selected from each subject group. The training set included 63,441 events from 69 subjects of the training group, while the test set contained 11,670 events from 17 subjects of the test group (Table 2).

Table 1
The demographics of the training group and the test group.

Measures	Training group	Test group	*p*-value	Total
Subjects (M: F)	69 (53: 16)	17 (12: 5)	=	86 (65: 21)
Age (years)	58.48 ± 10.74	56.88 ± 12.45	*NS*	58.18 ± 11.02
BMI (kg/m^2)	25.63 ± 3.05	25.33 ± 2.20	*NS*	25.57 ± 2.89
AHI (per h)	28.93 ± 19.00	19.28 ± 11.12	*NS*	27.02 ± 18.08
TRT (h)	7.46 ± 0.61	7.15 ± 1.04	*NS*	7.40 ± 0.72
TST (h)	5.86 ± 1.03	5.72 ± 1.07	*NS*	5.83 ± 1.04
SE (%)	78.40 ± 13.51	80.19 ± 11.64	*NS*	78.77 ± 13.10

Note: data are presented as mean ± SD; BMI: body mass index; AHI: apnea-hypopnea index; TRT: total recording time; TST: total sleep time; SE: sleep efficiency; *NS*: no significant difference between training and test set (p-value > 0.01).

Table 2
SA dataset specifications.

Events	Training set	Test set	Total
Normal	21,405	4561	25,966
Apnea	15,933	1623	17,556
Hypopnea	26,103	5486	31,589
Total	63,441	11,670	75,111

2.2. Data processing

The designed deep learning approaches were trained and validated on the SA dataset obtained from 86 patients diagnosed with SA. The single-lead ECG signals were recorded at 200 Hz over approximately 6 h during the PSG recording. ECG signals were filtered by an FIR bandpass (0.5–30 Hz) to remove the noise and baseline drift. Subsequently, the ECG signals were segmented to match event-based classification. Non-overlapping segmentations were applied to entire ECG signals, which were divided into 10-s intervals. ECG signals were converted into 2D spectrogram images to generate 2D input signals using the following short-time Fourier transformation:

$$x[n,k] = \sum_{m=0}^{L-1} w[m] \cdot x[n+m] \cdot e^{-jm(2\pi k/N)} \quad (1)$$

where n and k denote the time the signal and the signal frequency were received, respectively, and $w[m]$ is a window function where the window length was 128 points with a 127-point overlap. Each segment of the 1D and 2D approaches was formatted into (1×2000) and (129×1873) file format, respectively (Fig. 1). All abovementioned signal processing was performed by the signal processing toolbox of MATLAB software (Mathworks, U.S.A.).

2.3. CNN-based approaches

A CNN consists of three main parts: convolution layer, pooling layer, and classification layer. In the convolution layer, the feature map was extracted by applying a filter kernel to produce the convolution integral of the input data activation function, thereby enhancing discrimination. In the pooling layer, the feature map has reduced and restricted the dimensions of input data. Finally, the classification layer is performed the final discrimination of the input data by using the fully-connected network. At this stage, and the learning process is performed through feed-forward and back-propagation algorithms.

Two CNN models were designed and optimized for automatic detection of SA events, namely the 1D CNN and 2D CNN (Fig. 2). Primarily, the 1D CNN model can be used in biomedical engineering and speech recognition applications that use time-series signal as input. We compared the 1D and 2D CNN model to evaluate the differences and determine which is more appropriate in the application of physiological signal such as ECG. The 1D CNN model used a pre-processed ECG signal as input, which was batch-normalized before input to the CNN-based approach. Next, the deep learning model proceeds with the 1D convolution operation, which is explained in detail below, and 1D pooling at the convolution and pooling layers followed by the activation function and dropout. Finally, the 1D CNN model uses a fully-connected layer to discriminate SA events (Fig. 2A).

The 2D CNN model is a general method of deep learning applied to computer vision and the image recognition fields, where

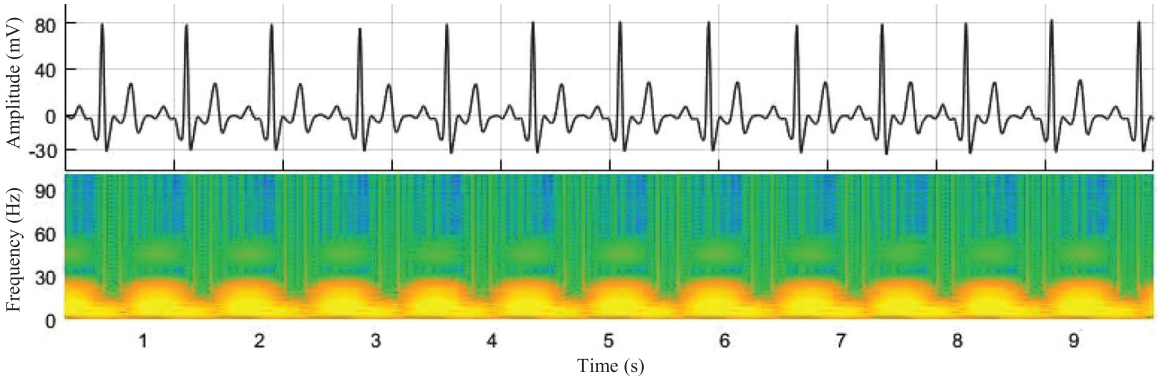

Fig. 1. Example of input signal for the designed deep learning approaches. Pre-processed ECG segment as 1D input (top), spectrogram of ECG segment as 2D input (bottom).

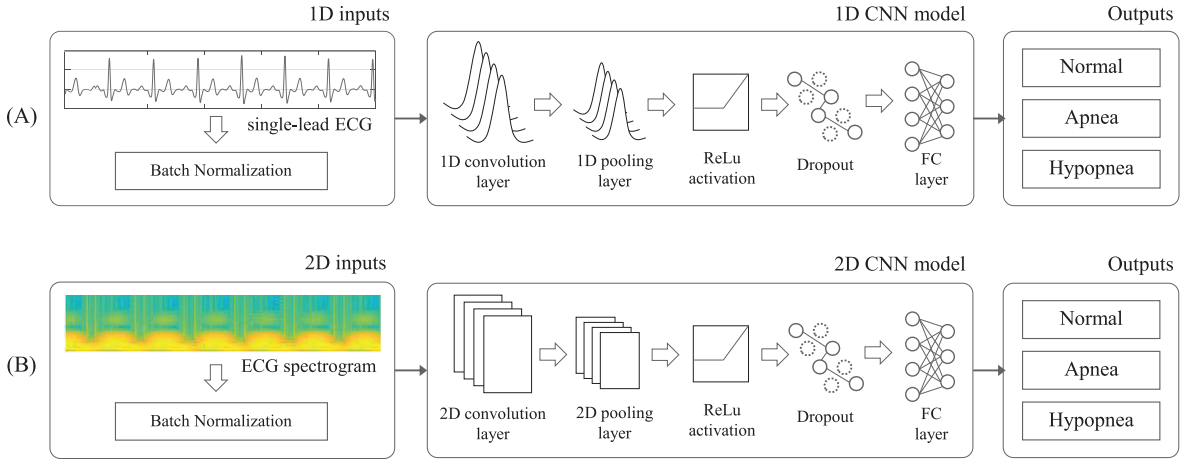

Fig. 2. CNN–based approaches for automatic detection of SA events. (A) Flowchart of 1D CNN model, six–layer convolution, three kernels with sizes of 50×1, 30×1, and 10×1, pooling size 1×2, dropout $p = 0.25$. (B) Flowchart of 2D CNN model, seven–layer convolution, three kernels with sizes of 50×2, 30×2, and 10×2, pooling size 2×2, dropout $p = 0.25$.

the image is used as input. In this study, the 2D CNN model received a whole spectrogram of the pre-processed ECG segment as the input image. The 2D convolution and 2D pooling were performed with ReLu activation and dropout, to avoid overfitting and divergence. This routine was repeated seven times, upon which the CNN model generated the feature maps that were used at the fully-connected layer for final classification (Fig. 2B).

2.3.1. 1D and 2D convolution

CNN-based approaches were performed to a 1D and 2D convolution operation, according to the inputs. Each convolutional operation was calculated at the convolutional layer of CNN model. First, the 1D convolution operation of convolutional layer was computed according to the following expression:

$$x = f\left(b + \sum_{i=1}^{N} *(w, y)\right) \tag{2}$$

where x is the total feature map, b is the bias, y is a feature map, w is the convolutional kernel, N is the total number of features, $(*)$ and $f(.)$ are vector convolution and the activation function, respectively.

The spectrogram, or 2D conversion, of the ECG signals for the 2D CNN approach had a straightforward application to the 2D convolutional kernels. Input data I had the dimensions (M_i, N_i) and the filter function $f(.)$ had the dimensions (M_f, N_f), such that the 2D convolution kernels which calculated the full output size could be expressed as:

$$C(j,k) = \sum_{m=0}^{M_i-1} \sum_{n=0}^{N_i-1} I(m,n) * f(j-m, k-n) \tag{3}$$

where $0 \le j \le M_i + M_f - 1$ and $0 \le k \le N_i + N_f - 1$.

2.3.2. CNN architecture

The CNN-based approaches used almost same architecture for automatic detection of SA events. In Table 3, we presented the architecture of the 1D CNN and 2D CNN models. The 1D CNN model used three kernels with sizes of 50×1, 30×1, and 10×1. Each pooling layer in the 1D CNN model computed 1×2 max-pooling regions. The 2D CNN model employed kernels with sizes of 50×2, 30×2 and 10×2 and 2×2 max-pooling regions. It used one more convolutional layer than the 1D-CNN model.

2.4. RNN-based approaches

A recurrent neural network (RNN) is regarded as most common type of the conventional neural network that can accept variable and sequence input [40]. RNN is ideally suited to sequential data, which proves robust for time series, as they have a memory as well. Consequently, the current input data and previous state affect the output of the next state.

RNN-based approaches consist of three basic sections: input blocks, the deep learning model, and output blocks (Fig. 3). LSTM and GRU models were used for RNN-based approaches, and the same architectures were applied to the corresponding datasets

Table 3
Architectures of the 1D CNN and 2D CNN models.

Layers	1D CNN			2D CNN		
	Filter size	Output shape	Parameters	Filter size	Output shape	Parameters
batchnorm_1	=	2000 × 1	4	=	129 × 1873	4
conv_1	16@50 × 1	1985 × 16	272	32@50 × 2	128 × 1858 × 32	1056
maxpool_1	2 × 1	992 × 16		2 × 2	64 × 929 × 32	
conv_2	16@50 × 1	977 × 64	16,448	64@50 × 2	63 × 914 × 64	65,600
conv_3	64@30 × 1	962 × 64	65,600	64@30 × 2	62 × 899 × 64	131,136
maxpool_2	2 × 1	481 × 64		2 × 2	31 × 449 × 64	
batchnorm_2	=	481 × 64	256	=	31 × 449 × 64	256
dense_1	32@30 × 1	481 × 32	2080	32@30 × 2	31 × 449 × 32	2080
maxpool_3	2 × 1	240 × 32		2 × 2	15 × 224 × 32	
conv_4	24@10 × 1	233 × 32	8224	24@10 × 2	14 × 217 × 32	16,416
maxpool_4	2 × 1	116 × 32		2 × 2	7 × 108 × 32	
conv_5	16@10 × 1	109 × 16	4112	12@10 × 2	6 × 101 × 16	8208
maxpool_5	2 × 1	54 × 16		2 × 2	3 × 50 × 16	
flatten_1	2	864 × 2	1730	2	2400 × 2	4802
dense_1						
Total	168 filters		98,726			229,558

Note: ReLu activation was used for all deep learning models.

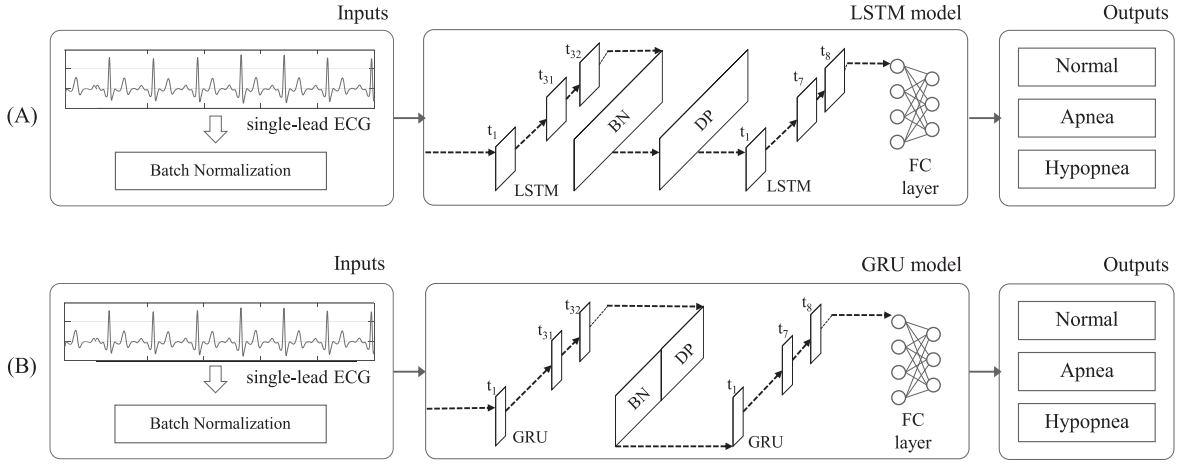

Fig. 3. RNN–based approaches for automatic detection of SA events. (A) LSTM model, (B) GRU model. Each model has the same architecture of three–layer RNN with 60, 80, and 120 memory cells.

to compare their performance. Detailed explanations of LSTM (Fig. 3A) and GRU (Fig. 3B) models follow.

2.4.1. Long short-term memory

Long short-term memory (LSTM) is an updated version of a basic RNN with memory cells that facilitates the learning of temporal correlations of data over time. The concept of LSTM is based on a memory cell that handles the read, write, and reset functions of its internal state through an input gate (i_t), an output gate (o_t), and a forget gate (f_t), respectively. Each gate works to remember when and to what extent the weights in the memory should be updated. The input and output gates controls the flow of input and output of the memory cell activations. The forget gate deals the internal state of the cell, therefore adaptively forgetting or resetting the cell's memory (Fig. 4A). The following expressions depict the functions of these gates.

$$i_t = \sigma\left(W^{xi}x_t + W^{hi}h_{t-1} + b_i\right) \tag{4}$$

$$f_t = \sigma\left(W^{xf}x_t + W^{hf}h_{t-1} + b_f\right) \tag{5}$$

$$o_t = \sigma\left(W^{xo}x_t + W^{ho}h_{t-1} + b_o\right) \tag{6}$$

$$g_t = \sigma\left(W^{xc}x_t + W^{hc}h_{t-1} + b_c\right) \tag{7}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{8}$$

$$h_t = o_t \cdot \varphi(c_t) \tag{9}$$

where c is the cell activation vector. Terms σ and τ are the nonlinear hyperbolic and tangent functions. x_t is the input to the memory cell layer at time t. W is weight matrices, b_i, b_f, b_c, and b_o are bias vectors.

2.4.2. Gated-recurrent memory

A gated-recurrent unit (GRU) is a regarded as a simplified version of the LSTM that contains two gates, namely the update (z_t) and reset gate (r_t), and controls the flow of information similarly to LSTM without the memory unit. Despite these simplifications, GRUs have shown similar performance to those of LSTMs [41]. In addition, GRU can reduce the number of calculations per training phase (Fig. 4B). These calculations are can be represented as:

$$z_t = \sigma\left(W^{xz}x_t + W^{hz}h_{t-1} + b_z\right) \tag{10}$$

$$r_t = \sigma\left(W^{xr}x_t + W^{hr}h_{t-1} + b_r\right) \tag{11}$$

$$\tilde{h}_t = \tanh\left(Wx_t + Wh_{t-1} \cdot r_t\right) \tag{12}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \tag{13}$$

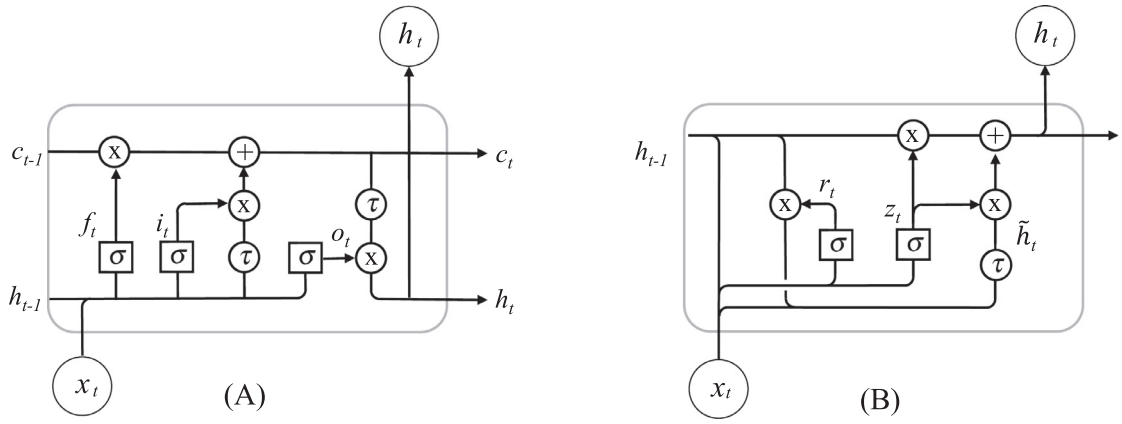

Fig. 4. Unit structure of (A) LSTM and (B) GRU. (A) f_t, i_t, and o_t are the forget, input, and output gates, respectively. c_{t-1} and c_t denote the memory cell and the new memory cell content. (B) r_t and z_t are the reset and update gates, and h_t and $\tilde{h}_t$ depict the activation and the candidate activation.

Table 4
Architectures of the LSTM and GRU models.

Layers	LSTM			GRU		
	Memory cells	Activation	Parameters	Memory cells	Activation	Parameters
batchnorm_0	=	=	4	=	=	4
rnn_1	120	sigmoid	58,560	120	sigmoid	43,920
batchnorm_1			480			480
rnn_2	100	sigmoid	88,400	100	sigmoid	66,300
batchnorm_2			400			400
rnn_3	80		57,920	80	sigmoid	43,440
batchnorm_3		sigmoid	320			320
dense_1			6480			6480
rnn_4	60	sigmoid	33,840	60	sigmoid	25,380
batchnorm_4			240			240
rnn_5	40	sigmoid	16,160	40	sigmoid	12,120
batchnorm_5			160			160
rnn_6	20		4880	20		3660
batchnorm_6		sigmoid	80		sigmoid	80
dense_2			420			420
dense_3	2	softmax	42	2	softmax	42
Total			268,386			229,558

2.4.3. RNN architecture

RNN-based approaches were designed for this study as shown in Table 4. The architecture consisted of three layers of RNNs, each of which had either 60, 80, or 120 memory cells. After the RNN layers, output feature maps conducted batch normalization and dropout to avoid overfitting and divergence. Then the optimal architecture for automatic detection of SA events was determined empirically and used in LSTM and GRU models to compare their performance.

2.5. Implementation

The deep learning approaches were implemented using Python 2.7, the Keras library, and a TensorFlow background [42]. Keras is a deep learning library that used to build and evaluate the designed deep learning approaches. Each approach training and testing was conducted on a hardware specification with a GTX1080 Ti (3584 CUDA cores) in the Win10 environment. Training of the deep learning approaches was entirely supervised by back-propagating algorithm. The model parameters were optimized by minimizing cross-entropy loss functions based on the Adam update rule [43]. Data were segmented into mini-batches of 256 data segments to optimize the training and testing processes. An accumulated gradient was computed using this configuration for the parameters after being trained on every mini-batch.

2.6. Performance evaluation

The accuracy, sensitivity, and specificity were calculated as the evaluation measure to the performance of the designed deep learning approaches. In addition, Cohen's kappa coefficient (k) was calculated for comparison purposes. The evaluation measures are expressed as below:

$$accuracy = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \tag{14}$$

$$sensitivity = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{15}$$

$$specificity = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{16}$$

$$k = (accuracy - P_e)/(1 - P_e) \tag{17}$$

where true positive (TP), the number of normal events is classified as normal, true negative (TN), the number of events abnormal counted as abnormal, false positive (FP), the number of events abnormal detected as normal, and false negative (FN), the number of events normal presented as abnormal. P_e is the hypothetical probability of agreement by chance.

3. Results

The sensitivity, specificity, and accuracy of the designed deep learning approaches was evaluated in the DNN, 1D CNN, 2D CNN, RNN, LSTM, and GRU models for the automatic detection of SA events. In the test set, the 1D CNN model exhibited an accuracy of 98.5%, 96.4%, and 96.3% for apnea, hypopnea, and A+H events, respectively. The 2D CNN model performance on the test set showed an accuracy of 95.9%, a sensitivity of 96.0%, and a specificity of 96.0% for apnea, 95.8%, 96.0%, and 96.0%, for hypopnea, and 91.2%, 92.0%, and 91.0% for combined A+H events, respectively. DNN model showed a lower accuracy of 93.1%, 82.3%, and 85.3% for apnea, hypopnea, and A+H events, even though it has the same architecture of the 1D CNN model (Table 5).

Furthermore, the LSTM model obtained an accuracy of 98.0%, 97.0%, and 96.0% for apnea, hypopnea, and A+H events, respectively. The GRU model had the accuracy, sensitivity, and specificity with 99.0%, 99.0%, and 99.0% for apnea events, 97.0%, 97.0%, and 97.0%, for hypopnea events, and 95.0%, 95.0%, and 96.0% for combined A+H events, respectively (Table 6). In contrast, simple RNN model achieved the accuracy, sensitivity, and specificity with 85.4%, 97.0%, and 87.0% for apnea events, 80.7%, 95.0%, and 79.0% for hypopnea events, and 83.2%, 96.0%, and 82.0% for A+H events, respectively.

The accuracy of the designed deep learning approaches (1D CNN, 2D CNN, LSTM, and GRU) is compared in Fig. 5. A more significant difference was found in the performance between the 1D and 2D CNN models than between the RNN-based approaches in all experiments (Fig. 5). The 1D CNN model performed best in detecting combined A+H events. LSTM and GRU models were not shown a significant difference in the terms of accuracy. However, they did require at least 20 iterations to achieve optimal performance and showed some spikes and fluctuations thereafter.

4. Discussion

The deep learning approaches were designed and found the optimal method for automatically detecting SA events based on an ECG signal. Six deep learning models, based on DNN, CNN and RNN, were designed and evaluated for their effectiveness. Finally, we obtained very high performances, with accuracies of 98.5%, 95.9%, 98.0%, and 99.0% for 1D CNN, 2D CNN, LSTM, and GRU, respectively.

CNNs are popular models used in image recognition, bioinformatics, and medical imaging applications. They are capable of representing the low to high-level features of input data through linear and nonlinear data abstraction. CNNs perform a morphology-based recognition, such that they can accept one-, two-, or three-dimensional input data. The 1D CNN and 2D CNN models were compared in the context of automatic detection of SA events. The designed 1D CNN model exhibited higher performances, and it is the lightest and simplest among the designed deep learning

Table 5
Performance evaluation of the CNN-based approaches and DNN.

Event	Approach	Accuracy (%)	Sensitivity (%)	Specificity (%)	Kappa
Apnea	DNN	93.1	93.0	94.0	0.85
	1D CNN	98.5	99.0	99.0	0.98
	2D CNN	95.9	96.0	96.0	0.92
Hypopnea	DNN	82.3	85.0	83.0	0.67
	1D CNN	96.4	96.0	96.0	0.92
	2D CNN	95.8	96.0	96.0	0.92
A+H	DNN	85.3	88.0	85.0	0.74
	1D CNN	96.3	96.0	96.0	0.92
	2D CNN	91.2	92.0	91.0	0.83

Table 6
Performance evaluation of the RNN-based approaches.

Event	Approach	Accuracy (%)	Sensitivity (%)	Specificity (%)	Kappa
Apnea	RNN	85.4	97.0	87.0	0.81
	LSTM	98.0	98.0	98.0	0.96
	GRU	99.0	99.0	99.0	0.98
Hypopnea	RNN	80.7	95.0	79.0	0.75
	LSTM	97.0	97.0	97.0	0.94
	GRU	97.0	97.0	97.0	0.94
A+H	RNN	83.2	96.0	82.0	0.78
	LSTM	96.0	96.0	96.0	0.92
	GRU	95.0	95.0	96.0	0.91

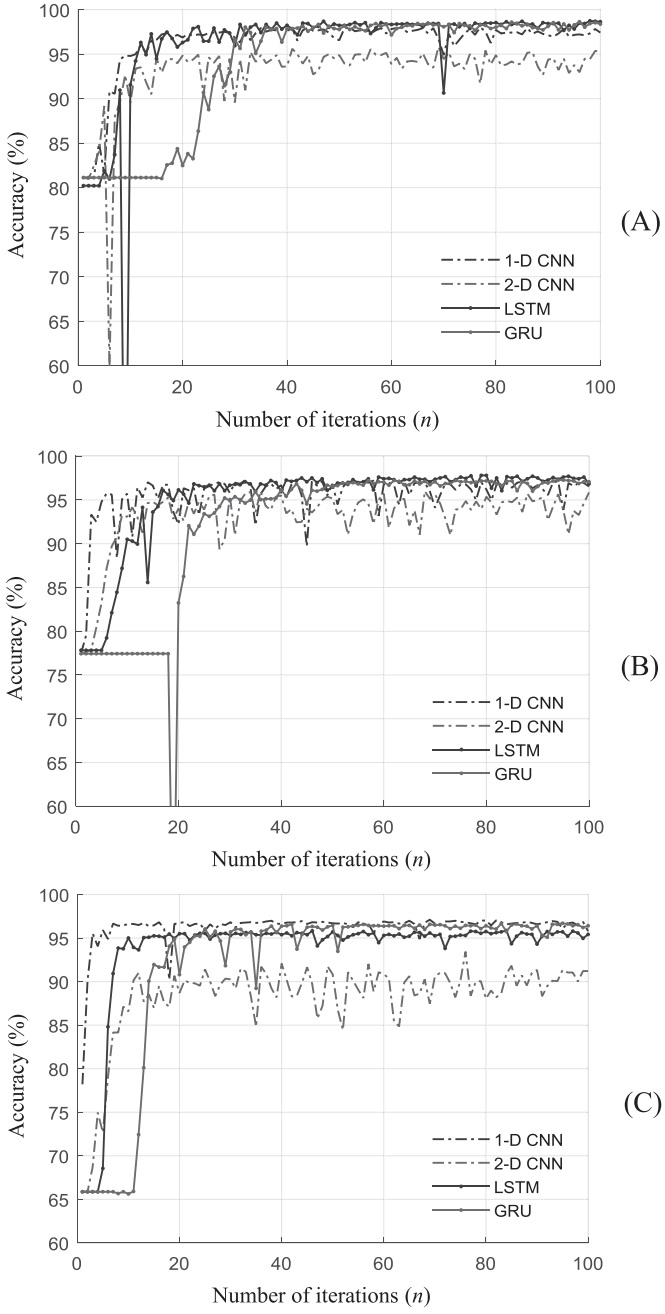

Fig. 5. Comparison of the accuracy of the deep learning models. (A) Graph of accuracy for apnea events. (B) Graph of accuracy for hypopnea events. (C) Accuracy for combined A+H events.

Table 7
Performance comparison with previous studies.

Study	Subject	Method	Accuracy (%)	Sensitivity (%)	Specificity (%)
Jafari [14]	35	SVM	94.8	95.4	94.1
Chen [15]	90	SVM	97.4	98.9	92.9
Hassan [44]	35	RUSBoost	88.8	87.5	91.4
Sharma [45]	35	LS-SVM	90.1	90.8	88.8
Nishad [46]	35	RF	92.7	93.9	90.9
Viswabhargav [47]	35	SVM	=	85.4	92.6
Pathinarupothi [36]	35	LSTM	98.0	=	=
Cheng [48]	35	LSTM	97.8	=	=
Dey [49]	35	CNN	98.9	97.8	99.2
Choi [38]	179	CNN	96.6	81.1	98.5
Our method	86	DNN	93.1	93.0	94.0
		1D CNN	98.5	99.0	99.0
		2D CNN	95.9	96.0	96.0
		RNN	85.4	97.0	87.0
		LSTM	98.0	98.0	98.0
		GRU	99.0	99.0	99.0

approaches. In addition, it is straightforward to apply and enables facile analysis of physiological information, such as the ECG signal. Meanwhile, the 2D CNN model showed the loweer performance in comparison to the 1D CNN model. The 2D CNN model has the bigger number of parameters and requires a high computational cost. Moreover, the conversion or domain transformation process is needed to apply the time series into the 2D CNN model. Hence, the 1D CNN outperformed the 2D CNN model in every respect. In the case where the input signal is the physiological signal as ECG or time series, we recommend the 1D CNN model for any application in deep learning and artificial intelligence algorithm.

RNNs are familiar models used with sequence data such as speech signal, physiological signal, and gene sequences. They have memory gates that store the previous sequences and use those to predict the next sequence. RNNs conduct gate-based classification using the input, update, and forgot gates. With respect to the designed LSTM model, not only was the computational cost the highest, but also the amount of training/learning time, since it contains the biggest number of parameters. No significant differences were found between the performances of the LSTM and GRU models, though the LSTM model performed slightly worse at apnea and hypopnea event detections than the GRU model. Furthermore, there is a significant difference between the computational cost and training time of the LSTM and GRU models in terms of the automatic detection of SA events. This is because the LSTM model has one additional memory cell in comparison to the GRU model. Meanwhile, the GRU model exhibited robust performance and relatively lower computational cost for the training and test phases. However, GRU model demonstrated comparable and higher performances in comparison to the 1D CNN model, for all SA events. For this reason, the GRU model is deemed to be appropriate for automatic detection of SA events using an ECG signal.

For automatic detection of SA events, conventional studies performed well and obtained high scores. In those studies, they follow the canonical procedure of supervised learning, which consists of data processing, feature extraction, feature selection, and classification. They extract a number of features to classify SA by using several signal processing techniques that analyze ECG signal at the time, frequency, and non-linear domain [10-13,17-19]. Then they obtain many well-known and powerful features to automatically detect SA events from an ECG signal. However, all processes are hand-crafted, require heavy computation, and require domain knowledge. Deep learning approaches, such as CNN and RNN models, do not require a separate hand-crafted feature extraction process. They are capable of learning which features are significant directly from raw data, using the convolution processes and memory cells. Another important point of deep learning is the ability to detect specific events like hypopnea that cannot be discriminated using conventional machine learning methods. Hypopnea events have consistently been the most difficult to detect from an ECG signal. They are associated with reduced airflow and respiratory effort, such that the ECG signals they produce may appear similar to those produced in normal sleep. For this reason, AASM guidelines recommend the use of additional physiological signals such as airflow, SpO_2, and CO_2 measurements in order to reliably determine whether hypopnea events have occurred [39]. However, the deep learning approaches designed and implemented in this study proved able to effectively detect hypopnea events using ECG signal.

The designed deep learning approaches were compared with previous studies, which used RNN models listed in the lower rows of Table 7. Pathinarupothi et al. [36] developed an LSTM algorithm that was 98.0% accurate; however, the pulse-to-pulse interval features had to be extracted manually before it could be trained. Cheng et al. [48] likewise applied an LSTM algorithm but used an ECG and an SpO_2 signal from which and instantaneous heart rate data was extracted to train the algorithm. Despite involving a larger amount of data from the two-channel signal, this model performed worse than the designed deep learning approaches in this study. Hence, the designed deep learning models using an ECG signal without any feature extraction demonstrated better performance in comparison to other methods and were able to distinguish between apnea and hypopnea events.

In this study, our designed deep learning models have some advantages in automatically detecting SA events. First, we did not use any hand-crafted or hand-extracted features for the automatic detection of SA events. Therefore, there is no need of the preprocessing such as feature extraction and selection. Second, our study population consists of the diverse and equally sized SA patients including mild, moderate and severe groups. Third, we have designed and evaluated six different deep learning models by the same datasets to demonstrate the differences between each model. Fourth, the designed deep learning models have greater ability to classify or detect events in comparison with conventional classification methods listed in Table 7. Finally, designed deep learning models can more accurately detect SA and better distinguish events including apnea, hypopnea, and normal breath using an ECG signal. Nevertheless, this study has some limitations, which are described as follows. Primarily, this study does not cover some types of SA, including central and mixed apnea events. In addition, the ECG signal was contaminated by position changing, loudy snoring, and coughs during sleep. Finally, a relatively small number of patients and SA dataset was used for training and evaluation of this study. In further research, we should be conducted the study that overcomes these limitations of the automatic detection of SA

events by using multi-modal and multi-class classification based on deep reinforcement learning.

5. Conclusions

In this study, the comprehensive analysis of the representative deep learning models was performed for automatic detection of SA events using an ECG signal. We designed six different deep learning models (DNN, 1D CNN, 2D CNN, RNN, LSTM, and GRU) for detecting apnea, hypopnea and normal event from ECG signal. Designed deep learning models did not use any hand-extracted features, and they were trained and tested using a single-lead ECG signal of the clinical PSG study. We obtained robust performance for all SA events than conventional studies that used the ECG signal. Also, 1D CNN and GRU model were more appropriate to use for automatic detection of SA events using a single-lead ECG signal than other models that we have designed. We can recommend 1D CNN and GRU for automatic detection based on the time series signals such as ECG and other physiological signals.

Funding

This work was supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the National Innovation Cluster R&D program [P0006697]; and supported by the Yonsei University Research Fund of 2019.

Declaration of Competing Interest

None.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2019.105001.

References

[1] E.J. Olson, W.R. Moore, T.I. Morgenthaler, P.C. Gay, B.A. Staats, Obstructive sleep apnea-hypopnea syndrome, Mayo Clin. Proc. 78 (2003) 1545–1552 https://doi.org/10.4065/78.12.1545.
[2] N.M. Punjabi, The epidemiology of adult obstructive sleep apnea, Proc. Am. Thorac. Soc. 5 (2008) 136–143 https://doi.org/10.1513/pats.200709-155MG.
[3] D.J. Gottlieb, G. Yenokyan, A.B. Newman, G.T. O'Connor, N.M. Punjabi, S.F. Quan, S. Redline, H.E. Resnick, E.K. Tong, M. Diener-West, E. Shahar, Prospective study of obstructive sleep apnea and incident coronary heart disease and heart failure: the sleep heart health study, Circulation 122 (2010) 352–360, doi:10.1161/CIRCULATIONAHA.109.901801.
[4] N. Botros, J. Concato, V. Mohsenin, B. Selim, K. Doctor, H.K. Yaggi, Obstructive sleep apnea as a risk factor for type 2 diabetes, Am. J. Med. 122 (2009) 1122–1127 https://doi.org/10.1016/j.amjmed.2009.04.026.
[5] Y.T. Chou, P.H. Lee, C.T. Yang, C.L. Lin, S. Veasey, L.P. Chuang, S.W. Lin, Y.S. Lin, N.H. Chen, Obstructive sleep apnea: a stand-alone risk factor for chronic kidney disease, Nephrol. Dial. Transplant. 26 (2011) 2244–2250 https://doi.org/10.1093/ndt/gfq821.
[6] H. Yaggi, V. Mohsenin, Sleep-disordered breathing and stroke, Clin. Chest Med. 24 (2003) 223–237 https://doi.org/10.1016/s0272-5231(03)00027-3.
[7] P.E. Peppard, M. Szklo-Coxe, K.M. Hla, T. Young, Longitudinal association of sleep-related breathing disorder and depression, Arch. Intern. Med. 166 (2006) 1709–1715 https://doi.org/10.1001/archinte.166.16.1709.
[8] S.J. Kim, J.H. Lee, D.Y. Lee, J.H. Jhoo, J.I. Woo, Neurocognitive dysfunction associated with sleep quality and sleep apnea in patients with mild cognitive impairment, Am. J. Geriatr. Psychiatry 19 (2011) 374–381 https://doi.org/10.1097/JGP.0b013e3181e9b976.
[9] V. Kapur, K. Strohl, S. Redline, C. Iber, G. O'Connor, J. Nieto, Underdiagnosis of sleep apnea syndrome in U.S. communities, Sleep Breath 6 (2002) 49–54 https://doi.org/10.1007/s11325-002-0049-5.
[10] A. Khandoker, C. Karmakar, M. Palaniswami, Automated recognition of patients with obstructive sleep apnoea using wavelet-based features of electrocardiogram recordings, Comput. Biol. Med. 39 (2009) 88–96 https://doi.org/10.1109/TITB.2012.2185809.
[11] M. Mendez, J. Corthout, S. Van Huffel, M. Matteucci, T. Penzel, S. Cerutti, A. Bianchi, Automatic screening of obstructive sleep apnea from the ECG based on empirical mode decomposition and wavelet analysis, Physiol. Meas. 31 (2010) 273–289 https://doi.org/10.1088/0967-3334/31/3/001.
[12] A. Yildiz, M. Akın, M. Poyraz, An expert system for automated recognition of patients with obstructive sleep apnea using electrocardiogram recordings, Expert Syst. Appl. 38 (2011) 12880–12890 https://doi.org/10.1016/j.eswa.2011.04.080.
[13] H. Al-Angari, A. Sahakian, Automated recognition of obstructive sleep apnea syndromeusing support vector machine classifier, IEEE Trans. Inf. Technol. Biomed. 16 (2012) 463–468 https://doi.org/10.1109/TITB.2012.2185809.
[14] A. Jafari, Sleep apnoea detection from ECG using features extracted from reconstructed phase space and frequency domain, Biomed. Signal Process. Control 8 (2013) 551–558 https://doi.org/10.1016/j.bspc.2013.05.007.
[15] L. Chen, X. Zhang, C. Song, An automatic screening approach for obstructive sleep apnea diagnosis based on single-lead electrocardiogram, IEEE T. Autom. Sci. Eng. 12 (2015) 106–115 https://doi.org/10.1109/TASE.2014.2345667.
[16] R.K. Tripathy, Application of intrinsic band function technique for automated detection of sleep apnea using HRV and EDR signals, Biocybern. Biomed. Eng. 38 (2018) 136–144 https://doi.org/10.1016/j.bbe.2017.11.003.
[17] J.V. Marcos, R. Hornero, D. Álvarez, F. Del Campo, C. Zamarrón, M. López, Utility of multilayer perceptron neural network classifiers in the diagnosis of the obstructive sleep apnoea syndrome from nocturnal oximetry, Comput. Methods Programs Biomed. 92 (2008) 79–89 https://doi.org/10.1016/j.cmpb.2008.05.006.
[18] D. Álvarez, R. Hornero, J. Marcos, F. del Campo, Feature selection from nocturnal oximetry using genetic algorithms to assist in obstructive sleep apnoea diagnosis, Med. Eng. Phys. 34 (2012) 1049–1057 https://doi.org/10.1016/j.medengphy.2011.11.009.
[19] B. Xie, Hlaing Minn, Real-Time sleep apnea detection by classifier combination, IEEE Trans. Inf. Technol. Biomed. 16 (2012) 469–477 https://doi.org/10.1109/TITB.2012.2188299.
[20] J. Marcos, R. Hornero, D. Álvarez, F. Del Campo, M. Aboy, Automated detection of obstructive sleep apnoea syndrome from oxygen saturation recordings using linear discriminant analysis, Med. Biol. Eng. Comput. 48 (2010) 895–902 https://doi.org/10.1007/s11517-010-0646-6.
[21] J. Marcos, R. Hornero, D. Álvarez, M. Aboy, F. Del Campo, Automated prediction of the apnea-hypopnea index from nocturnal oximetry recordings, IEEE Trans. Biomed. Eng. 59 (2012) 141–149 https://doi.org/10.1109/TBME.2011.2167971.
[22] J. Solà-Soler, J. Fiz, J. Morera, R. Jané, Multiclass classification of subjects with sleep apnoea–hypopnoea syndrome through snoring analysis, Med. Eng. Phys. 34 (2012) 1213–1220 https://doi.org/10.1016/j.medengphy.2011.12.008.
[23] A.M. Benavides, R.F. Pozo, D.T. Toledano, J.L.B. Murillo, E.L. Gonzalo, L.H. Gómez, Analysis of voice features related to obstructive sleep apnoea and their application in diagnosis support, Comput. Speech Lang. 28 (2014) 434–452 https://doi.org/10.1016/j.csl.2013.08.002.
[24] J. Solé-Casals, C. Munteanu, O. Martín, F. Barbé, C. Queipo, J. Amilibia, J. Durán-Cantolla, Detection of severe obstructive sleep apnea through voice analysis, Appl. Soft Comput. 23 (2014) 346–354 https://doi.org/10.1016/j.asoc.2014.06.017.
[25] B. Koley, D. Dey, Automatic detection of sleep apnea and hypopnea events from single channel measurement of respiration signal employing ensemble binary SVM classifiers, Measurement 46 (2013) 2082–2092 https://doi.org/10.1016/j.measurement.2013.03.016.
[26] H. Lee, J. Park, H. Kim, K. Lee, New rule-based algorithm for real-time detecting sleep apnea and hypopnea events using a nasal pressure signal, J. Med. Syst. 40 (2016) 282 https://doi.org/10.1007/s10916-016-0637-8.
[27] S. Grover, S. Pittman, Automated detection of sleep disordered breathing using a nasal pressure monitoring device, Sleep Breath 12 (2008) 339–345 https://doi.org/10.1007/s11325-008-0181-y.
[28] V.A. Rossi, J.R. Stradling, M. Kohler, Effects of obstructive sleep apnoea on heart rhythm, Eur. Resp. J. 41 (2013) 1439–1451 https://doi.org/10.1183/09031936.00128412.
[29] M.E. Tagluk, M. Akin, N. Sezgin, Classification of sleep apnea by using wavelet transform and artificial neural networks, Expert Syst. Appl. 37 (2010) 1600–1607 https://doi.org/10.1016/j.eswa.2009.06.049.
[30] L. Almazaydeh, M. Faezipour, K. Elleithy, A neural network system for detection of obstructive sleep apnea through SpO2 signal features, Int. J. Adv. Comput. Sci. Appl 3 (2012) https://doi.org/10.14569/ijacsa.2012.030502.
[31] D. Álvarez-Estévez, V. Moret-Bonillo, Fuzzy reasoning used to detect apneic events in the sleep apnea-hypopnea syndrome, Expert Syst. Appl. 36 (2009) 7778–7785 https://doi.org/10.1016/j.eswa.2008.11.043.
[32] C. Angermueller, H.J. Lee, W. Reik, O. Stegle, DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning, Genome Biol 18 (2017) 67 https://doi.org/10.1186/s13059-017-1189-z.
[33] D. Ravı, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, G.Z. Yang, Deep learning for health informatics, IEEE J. Biomed. Health Inform. 21 (2017) 4–21 https://doi.org/10.1109/JBHI.2016.2636665.
[34] A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947.
[35] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, Recurrent neural network based language model, In INTERSPEECH-2010 1045–1048.
[36] R.K. Pathinarupothi, E.S. Rangan, E.A. Gopalakrishnan, R. Vinaykumar, K.P. Soman, Single sensor techniques for sleep apnea diagnosis using deep learning, IEEE J. Biomed. Health Inform (2017 August) 524–529 https://doi.org/10.1109/ICHI.2017.37.
[37] R.K. Tripathy, U.R. Acharya, Use of features from RR-time series and EEG signals for automated classification of sleep stages in deep neural network framework, Biocybern. Biomed. Eng. 38 (2018) 890–902 https://doi.org/10.1016/j.bbe.2018.05.005.

[38] S.H. Choi, H. Yoon, H.S. Kim, H.B. Kim, H.B. Kwon, S.M. Oh, K.S. Park, Real-time apnea-hypopnea event detection during sleep by convolutional neural networks, Comput. Biol. Med. 100 (2018) 123–131 https://doi.org/10.1016/j.compbiomed.2018.06.028.
[39] R.B. Berry, R. Brooks, C.E. Gamaldo, S.M. Harding, R.M. Lloyd, C.L. Marcus, B.V. Vaughn, The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications: Version 2.3, American Academy of Sleep Medicine, 2016.
[40] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, Recurrent neural network based language model, INTERSPEECH, 2010.
[41] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: encoder-decoder approaches, *arXiv preprint* 2014 (Accessed 12 Jan 2015).
[42] F. Chollet, Keras (2015) http://keras.io/.
[43] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint (2014).
[44] A.R. Hassan, M.A. Haque, An expert system for automated identification of obstructive sleep apnea from single-lead ECG using random under sampling boosting, Neurocomputing 235 (2017) 122–130 https://doi.org/10.1016/j.neucom.2016.12.062.
[45] M. Sharma, S. Agarwal, U.R. Acharya, Application of an optimal class of antisymmetric wavelet filter banks for obstructive sleep apnea diagnosis using ECG signals, Comput. Biol. Med. 100 (2018) 100–113 https://doi.org/10.1016/j.compbiomed.2018.06.011.
[46] A. Nishad, R.B. Pachori, U.R. Acharya, Application of TQWT based filter-bank for sleep apnea screening using ECG signals, J. Ambient Intell. Humaniz. Comput. (2018) 1–12 https://doi.org/10.1007/s12652-018-0867-3.
[47] C.S. Viswabhargav, R.K. Tripathy, U.R. Acharya, Automated detection of sleep apnea using sparse residual entropy features with various dictionaries extracted from heart rate and EDR signals, Comput. Biol. Med. 108 (2019) 20–30 https://doi.org/10.1016/j.compbiomed.2019.03.016.
[48] M. Cheng, W.J. Sori, F. Jiang, A. Khan, S. Liu, Recurrent neural network based classification of ECG signal features for obstruction of sleep apnea detection, in: Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2, 2017, pp. 199–202. https://doi.org/10.1109/CSE-EUC.2017.220.
[49] D. Dey, S. Chaudhuri, S. Munshi, Obstructive sleep apnoea detection using convolutional neural network based deep learning framework, Biomed. Eng. Lett. 8 (2018) 95–100 https://doi.org/10.1007/s13534-017-0055-y.