An Attention-Based Deep Learning Approach for Sleep Stage Classification With Single-Channel EEG

Emadeldeen Eldele, Zhenghua Chen, *Senior Member, IEEE*, Chengyu Liu, *Senior Member, IEEE*, Min Wu, *Senior Member, IEEE*, Chee-Keong Kwoh, Xiaoli Li, *Senior Member, IEEE*, and Cuntai Guan, *Fellow, IEEE*

***Abstract*— Automatic sleep stage classification is of great importance to measure sleep quality. In this paper, we propose a novel attention-based deep learning architecture called AttnSleep to classify sleep stages using single channel EEG signals. This architecture starts with the feature extraction module based on multi-resolution convolutional neural network (MRCNN) and adaptive feature recalibration (AFR). The MRCNN can extract low and high frequency features and the AFR is able to improve the quality of the extracted features by modeling the inter-dependencies between the features. The second module is the temporal context encoder (TCE) that leverages a multi-head attention mechanism to capture the temporal dependencies among the extracted features. Particularly, the multi-head attention deploys causal convolutions to model the temporal relations in the input features. We evaluate the performance of our proposed AttnSleep model using three public datasets. The results show that our AttnSleep outperforms state-of-the-art techniques in terms of different evaluation metrics. Our source codes, experimental data, and supplementary materials are available at https://github.com/emadeldeen24/AttnSleep.**

***Index Terms*— Sleep stage classification, multi-resolution convolutional neural network, adaptive feature recalibration, temporal context encoder, multi-head attention.**

I. Introduction

SLEEP is a vital process for humans, as it affects all the aspects in their daily activities. Studies show that humans having good quality of sleep enjoy better health and brain functions [1]. On the other hand, interrupted sleep periods can cause some sleep disorders, such as insomnia or sleep apnea [2]. In particular, sleep stages (e.g., light sleep and deep sleep) are important for immune system, memory, metabolism, etc. [3]–[5]. Therefore, it is highly desired to measure sleep quality through sleep monitoring and sleep stage classification.

Sleep specialists usually determine the sleep stages based on the polysomnography (PSG), which consists of electroencephalogram (EEG), electrooculogram (EOG), anelectromyogram (EMG) and electrocardiogram (ECG) [6]. Single-channel EEG has recently become attractive for sleep monitoring due to its ease-of-use. In particular, PSG or single-channel EEG recordings are usually divided into 30-second segments and each segment is manually checked by sleep specialists and then classified into one of the six stages, *i.e.*, wake (W), rapid eye movement (REM) and four non-REM stages (N1, N2, N3 and N4) [7]. This manual process is very exhaustive, tedious, and time-consuming. As such, automatic sleep stage classification systems are required to assist sleep specialists.

Many studies have adopted conventional machine learning methods to classify EEG signals into corresponding sleep stages. These methods usually consist of two steps, namely, manual feature extraction and sleep stage classification. First, they design and extract various features from time and frequency domains. Feature selection algorithms are often applied to further select the most discriminative features. Second, the selected features are then fed into conventional machine learning models for sleep stage classification, such as Naive Bayes [8], support vector machines (SVM) [9], [10], random forest (RF) [7], [11], or even ensemble learning based classifiers [12]. However, these methods require domain knowledge to extract the best representative features.

Recently, deep learning has been employed in different areas and shown its superiority over conventional machine learning models without the need of domain knowledge. This motivates researchers to exploit deep learning techniques for automatic sleep stages classification. Several studies have designed convolutional neural networks (CNNs) [13]–[18] for this task. For example, successive convolution and pooling layers with fully-connected layers were used to perform this classification task in [13]. In [14], the authors used 12 convolution layers together with 2 fully connected layers. Additionally in [16], the authors used 2D convolution along with MaxPooling layers

Manuscript received November 10, 2020; revised March 3, 2021; accepted April 21, 2021. Date of publication April 28, 2021; date of current version May 4, 2021. The work of Emadeldeen Eldele was supported by A*STAR SINGA Scholarship. *(Corresponding author: Min Wu.)*

Emadeldeen Eldele, Chee-Keong Kwoh, and Cuntai Guan are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: emad0002@ntu.edu.sg; asckkwoh@ntu.edu.sg; ctguan@ntu.edu.sg).

Zhenghua Chen, Min Wu, and Xiaoli Li are with the Institute for Infocomm Research, A*STAR, Singapore 138632 (e-mail: chen0832@e.ntu.edu.sg; wumin@i2r.a-star.edu.sg; xlli@i2r.a-star.edu.sg).

Chengyu Liu is with the School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: chengyu@seu.edu.cn).

Digital Object Identifier 10.1109/TNSRE.2021.3076234

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

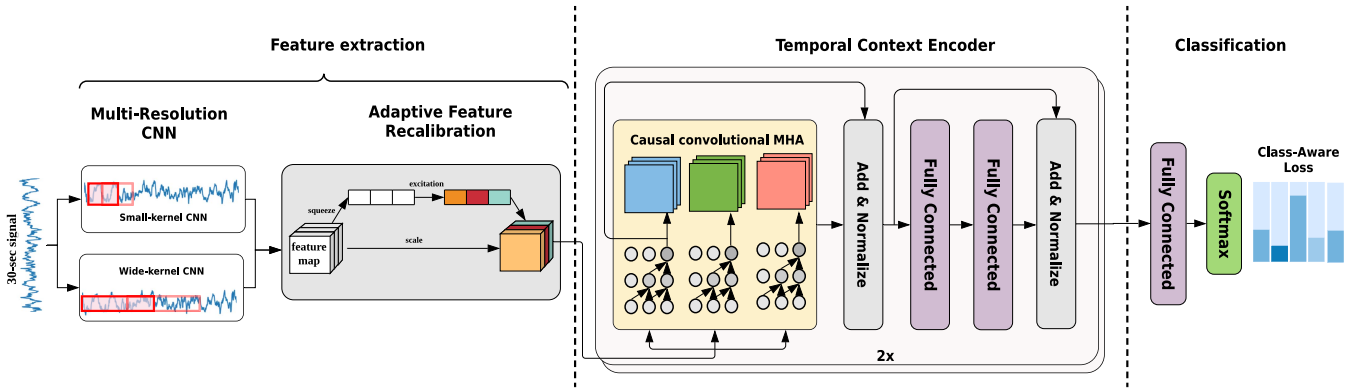

Fig. 1. Overall framework of the proposed AttnSleep model for automatic sleep stage classification.

to classify the raw data from three channels (*i.e.*, EEG, EOG, and EMG). In [15], the authors designed a relatively deep CNN architecture to show that a better performance can be achieved by relying on network depth. In [17], authors converted each raw signal into a log-power spectra and used a CNN to perform a joint classification and prediction task for identifying sleep stages. Generally, the CNN models above have achieved good performance for sleep stage classification. However, most of them are not able to effectively model the temporal dependencies among the EEG data.

Recurrent Neural Networks (RNNs) were proposed to capture temporal dependencies in time-series EEG data. For example, [19] used a cascaded RNN architecture to classify sleep stages. Some researchers combined CNN with RNN by using CNN for features extraction and RNN for modelling the time dependencies [20]–[23]. For example, [20] used a CNN architecture to extract features from the raw data, and then exploited the long short-term memory (LSTM) to learn transition rules through sleep stages. Similarly, in [23], the authors also used a successive blocks of CNN followed by LSTM to classify raw EEG signals. In addition, [24] used an LSTM based encoder-decoder with an attention mechanism after the encoder to find out the most relevant parts in input sequences. However, RNNs also have limitations due to their recurrent nature, *i.e.*, they usually have high model complexity and it is thus difficult to train them in parallel. Some works used attention mechanism completely instead of using RNN. For example, [25] segmented the EEG epochs and used self-attention to learn both intra-epoch features, and inter-epoch temporal features.

Aside from selecting classification models, sleep staging also needs to address the data imbalance problem, since humans spend different time periods in each stage. Oversampling is a common strategy to address this issue. For example, the studies in [13], [17], [20] replicated the minority classes for model training. The authors in [24] applied Synthetic Minority Over-sampling TEchnique (SMOTE) [26] for oversampling to balance the data. However, oversampling techniques expand the training data and thus increase the training time.

To address the above issues, we propose a novel architecture called AttnSleep for automatic sleep stages classification. First, we propose a novel feature extraction module based on multi-resolution CNN (MRCNN) and adaptive feature recalibration (AFR). The MRCNN extracts features corresponding to low and high frequencies from different frequency bands, and the AFR models the features inter-dependencies to enhance the feature learning. Second, we propose a novel temporal context encoder (TCE) that deploys a multi-head attention with causal convolutions to efficiently capture the temporal dependencies in the extracted features. We also design a class-aware loss function to effectively address the data imbalance issue without additional computations. We perform extensive experiments on three public datasets and experimental results demonstrate that our AttnSleep model outperform the state-of-the-arts for sleep stage classification.

Overall, the main contributions of our proposed model can be summarized as follows.

1) We propose a novel feature extraction technique, *i.e.*, a multi-resolution CNN module, to extract features corresponding to low and high frequencies from different frequency bands, and an adaptive feature recalibration to learn the features interdependencies and enhance the representation capability of the extracted features.
2) We propose a novel temporal context encoder that deploys a multi-head self-attention with causal convolutions to efficiently capture the temporal dependencies in the extracted features.
3) We design a class-aware loss function to efficiently handle the class imbalance without introducing additional computations.
4) These novel components are supported by extensive experiments over three public datasets. The results demonstrate that our proposed model outperforms state-of-the-arts in sleep stage classification.

The rest of the paper is organized as follows: Section II illustrates the details of the proposed model. In Section III, we introduce the datasets, evaluation metrics, experimental setup and the baseline methods. We then present the comparison results against the baseline methods and the ablation study of our AttnSleep model, as well as a sensitivity analysis of the choice of the number of heads in the MHA. Finally, the conclusion of the paper is presented in Section IV.

II. Proposed Method

In this section, we introduce our proposed AttnSleep model for sleep stage classification from single-channel EEG data.

A. Overview of AttnSleep Model

Fig. 1 illustrates the overall framework of our AttnSleep model. It consists of three main blocks, namely, 1) feature extraction, 2) temporal context encoder and 3) classification.

First, the MRCNN with two-branch CNN architectures is exploited to extract the features from a 30-second EEG signal. In particular, it extracts high-frequency features by the small kernel convolutions and low-frequency features by the wide kernel convolutions. Following MRCNN, we propose an AFR module to model the inter-dependencies among the features extracted by MRCNN. Moreover, AFR can adaptively select and highlight the most important features, which helps to enhance the classification performance. Second, we develop a TCE module to capture the long-term dependencies in the input features. The core component of TCE is the multi-head attention supported by causal convolutions. Third, the classification decision is done by a fully connected layer with a softmax activation function. We also leverage a class-aware cost-sensitive loss function to handle the data imbalance issue. In the following subsections, we will introduce each block in details.

B. Feature Extraction

Fig. 2 shows the MRCNN and AFR modules for feature extraction from raw single-channel EEG signals.

1) Multi-Resolution CNN: To extract different types of features, we develop a multi-resolution CNN architecture as shown in Fig. 2. We implement two branches of convolutional layers with different kernel sizes, where the choice of the kernel sizes is related to the sampling rate of the EEG signals and aims to explore different frequency bands. This is inspired by some previous works that used multiple CNN kernel sizes to extract different frequency features (*i.e.*, low- and high-frequencies) such as [27], [28]. Additionally, different sleep stages are characterized by different frequency ranges [7], and thus, it is becoming important to address different frequency bands to improve the extracted features. Therefore, we use different kernel sizes to capture different ranges of timesteps, and hence address features from different sleep-related frequency bands. To further explain this, we consider a dataset with a sampling rate of 100 Hz (100 timesteps are sampled in one second) to justify the selection of the kernel sizes of the two branches. First, the wide kernel (with a kernel of 400) captures timesteps with 4-second windows, and thus captures a whole cycle of sinusoidal signal down to ∼0.25 Hz ($T = 1/F$). This range corresponds to delta band. Second, for the smaller kernel (with a kernel of 50), each convolution window captures 50 samples (0.5 second), thus it will be able to capture a whole cycle of sinusoidal signal down to ∼2 Hz, which means that data corresponds to alpha and theta bands.

On the other hand, such a combination of features is important for the non-stationary characteristic of EEG signals that requires exploring different kinds of features. As shown in Fig. 2, each branch consists of three convolutional layers and two max-pooling layers, where each convolution layer includes a batch normalization layer [29] and uses a Gaussian Error Linear Unit (GELU) as the activation function. In particular,

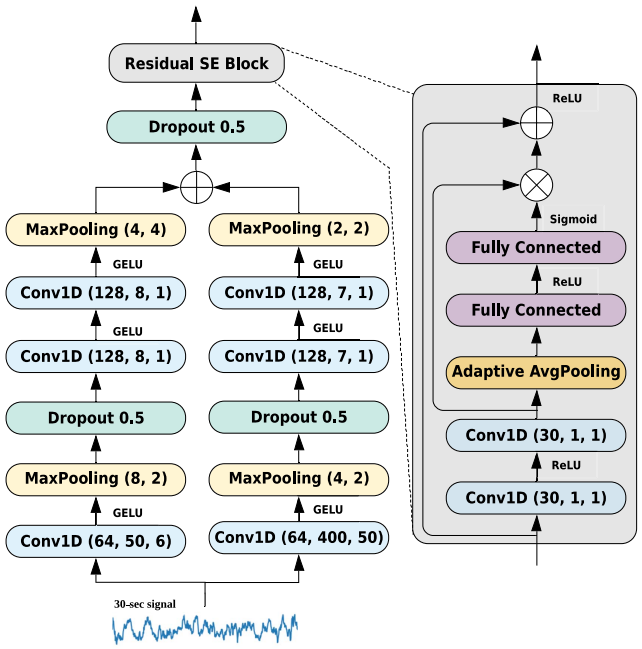

Fig. 2. The MRCNN and AFR modules for feature extraction. Each convolution block is followed by a Batch Normalization.

Conv1D (64, 50, 6) in Fig. 2 refers to using 1D convolution layer with 64 filters, a kernel size of 50 and a stride of 6. Similarly, MaxPooling (8, 2) refers to a maxpooling layer with a kernel size of 8 and a stride of 2. To reduce overfitting, we also apply dropout after the first maxpooling in both branches and after the concatenation of both branches as shown in Fig. 2.

2) Adaptive Feature Recalibration (AFR): AFR aims to recalibrate the features learned by MRCNN for improving its performance. In particular, the AFR models the inter-dependencies between the features and adaptively selects the most discriminative features through a residual squeeze and excitation (residual SE) block [30]. The SE block helps the lower layers of the network to exploit more contextual information outside its local receptive field by a context aware mechanism. In residual SE block, we implement two convolutions Conv1D (30,1,1) with both the kernel and stride size as 1 and ReLU as the activation function. Given a feature map $I \in \mathbb{R}^{L\times d}$ learned by MRCNN, we apply two convolution operations to I such that $F = Conv2(Conv1(I))$, where $F = \{F_1, \ldots, F_N\} \in \mathbb{R}^{N\times d}$, N is the total number of features, d is the length of F_i ($1 \le i \le N$), and $Conv1$ and $Conv2$ are the two convolution operations in AFR module.

Next, the global spatial information is squeezed by using adaptive average pooling that shrinks $F \in \mathbb{R}^{N\times d}$ to $\mathbf{s} = \{s_1, \ldots, s_N\}$, where s_i is the average of the d data points in $F_i \in \mathbb{R}^d$, $1 \le i \le N$. Two fully connected (FC) layers are then applied to make use of the aggregated information. In particular, the first layer is followed by a ReLU activation function to perform dimensionality reduction, and the second layer is followed by a smoothing sigmoid activation function to perform dimensionality increasing as shown in Equation 1.

$$\mathbf{e} = \sigma\left(\mathcal{W}_2\left(\delta(\mathcal{W}_1(\mathbf{s}))\right)\right) \quad \in \mathbb{R}^{N\times d}, \tag{1}$$

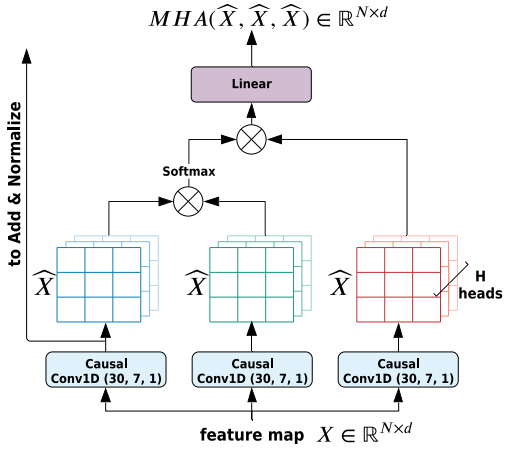

Fig. 3. Structure of proposed multi-head attention.

where σ and δ refer to sigmoid and ReLU activation functions respectively, and $\mathcal{W}_1$ and $\mathcal{W}_2$ represent the two FC layers in AFR. Then, the feature map F is scaled by $\mathbf{e}$ as follows:

$$\mathbf{O} = F \otimes \mathbf{e} \quad \in \mathbb{R}^{N\times d}, \tag{2}$$

where $\otimes$ refers to the point-wise multiplication between F and $\mathbf{e}$. We also add a shortcut connection to combine the original input I with the enhanced selected features learned from the residual SE block. The final output of the AFR module is:

$$X = I + \mathbf{O} \quad \in \mathbb{R}^{N\times d}. \tag{3}$$

Note that we use GELU activation function in the MRCNN module as it allows some negative weights of the input to pass through. These negative weights might be important for the following AFR module, leading to different decisions. Compared to ReLU, GELU should perform better, as ReLU suppresses all the negative weights to zeros, and thus the AFR module will not be able to make use of them. However, we use ReLU in the AFR module itself as ReLU aims to avoid exploding/vanishing gradient besides making the computations faster and easier to converge [31]. On the other hand, GELU can be a better choice over some other activation functions that also pass negative values, such as Leaky-ReLU and PReLU. The reason is that these activation functions allow strong negative activations to generate undesirable impact on the sum of activations feeding the next layers, which generates undesirable effects. Differently, GELU shows more control to bound the effect of these negative activations. These conclusions are supported with experiments in Table S.1 in the supplementary materials.

C. Temporal Context Encoder (TCE)

The TCE layer aims to capture temporal dependencies in the input features. As shown in Fig. 1, TCE layer consists of a multi-head attention (MHA) layer, a normalization layer and two FC layers. Moreover, TCE stacks two identical structures to generate the final features. As attention mechanism is a key part in the TCE module, we first introduce the self-attention mechanism, and then we introduce each component in the TCE layer.

1) Self-Attention: We use self-attention to quantify the inter-dependence within input features, at which higher weights are assigned to the regions of interest according to each position in the input, while lower weights are assigned for less interesting regions. In particular, given an input $Z = \{z_1, \ldots, z_N\} \in \mathbb{R}^{N\times d}$ where N is the total number of features, and d is the length of x_i, $1 \le i \le N$, this input is transformed into another space using a transforming function $\phi(\cdot)$. In our AttnSleep model, $\phi(\cdot)$ is a causal convolution function.

Next, we calculate a score α_{ij} that indicates the weight at which i-th position is attending to j-th position, as follows:

$$\alpha_{ij} = \frac{\exp\left(s_{ij}\right)}{\sum_{k=1}^{d}\exp\left(s_{ik}\right)}, \tag{4}$$

$$s_{ij} = \phi(z_i)\phi(z_j)^{\mathsf{T}}. \tag{5}$$

Each attention output element a_i is computed as weighted sum of the transformed input elements:

$$a_i = \sum_{j=1}^{d} \alpha_{ij}\phi(x_j). \tag{6}$$

The output of the attention layer is $A = (a_0, a_1, \ldots, a_d) \in \mathbb{R}^{N\times d}$.

2) Multi-Head Attention (MHA): MHA is inspired by the Transformer model [32], which shows great success in machine translation applications due to its ability to learn long range relationships in sentences [33]. MHA improves the self-attention in two main aspects. First, it expands the model's capability to focus on different positions, as the encoding of each head knows about the encodings of the other heads as well. This improves the model ability to learn temporal dependencies. Second, splitting the input features into different partitions increases the representation subspaces. Therefore, the generated attention weights for each subspace are more representative to the importance of each partition, and concatenating these representations produces better overall representation, which enhances the classification accuracy. In our AttnSleep model, MHA leverages the causal convolutions to encode the positional information of input features and capture their temporal relations. The causal convolutions have an advantage of fast and parallel processing, which significantly reduces the model training time compared with RNNs. Next we illustrate how MHA works in our AttnSleep model.

The output of the AFR module, denoted as $X = \{x_1, \ldots, x_N\} \in \mathbb{R}^{N\times d}$, serves as the input of MHA as shown in Fig. 3. Here, N is the total number of features, and d is the length of x_i, $1 \le i \le N$. More specifically, MHA takes three duplicates of X as inputs. First, the causal convolutions generate $\widehat{X}$ from X, *i.e.*, $\widehat{X} = \phi(X)$. Second, we pass the three matrices of $\widehat{X}$ to calculate the attention in Equation 7 according to [32].

$$ATT(\widehat{X},\widehat{X},\widehat{X}) = Softmax(\frac{\widehat{X}\cdot\widehat{X}^{T}}{\sqrt{d}})\cdot\widehat{X}, \tag{7}$$

where $(\cdot)$ is the multiplication operation.

We further expand the attention over H heads for each of the three matrices. In particular, each matrix $\widehat{X}$ is split into H subspaces, *i.e.*, $\widehat{X} = \{X^1, \cdots, X^H\}$, $\widehat{X}^h \in \mathbb{R}^{N\times\frac{d}{H}}$, $1 \le h \le H$.

In each subspace h, we calculate the attention A^h similarly in Equation 8.

$$A^h = ATT(\widehat{X}^h, \widehat{X}^h, \widehat{X}^h) \in \mathbb{R}^{N \times \frac{d}{H}}. \quad (8)$$

Finally, all the H representations are concatenated together to produce the final output as follows:

$$MHA(\widehat{X}, \widehat{X}, \widehat{X}) = Concat(A^1, \cdots, A^H) \in \mathbb{R}^{N \times d}. \quad (9)$$

3) Add and Normalize Layer: The TCE has two *Add & Normalize* layers, which add the output of the previous layer to the input of that layer through a residual connection, and then normalize the sum. This operation can be expressed as $LayerNorm(x + SubLayer(x))$, where $LayerNorm$ refers to applying layer normalization [34], $SubLayer$ refer to either the MHA or the two FC layers as shown in Fig. 1, and x is the input of the $SubLayer$. Using the residual connections helps the model to utilize the lower-layer features by propagating them to the higher layers if they are useful. Additionally, the normalization operation helps to speed up the training process.

4) Feed-Forward Layer: The outputs of the MHA layer are fed into a feed-forward neural network, which is a combination of two FC layers. This layer employs ReLU activation function to break the non-linearity in the model and consider the interactions among latent dimensions. This operation can be modeled as $F_{out} = \mathcal{W}_4(\delta(\mathcal{W}_3(x)))$, where $\mathcal{W}_3$ and $\mathcal{W}_4$ refer to the the two FC layers in the TCE module, as shown in Fig. 1.

D. Class-Aware Loss Function and Optimization

Basically, we can apply the standard multi-class cross-entropy in Equation 10 as the loss function for our model.

$$\mathcal{L} = -\frac{1}{M} \sum_{k=1}^{K} \sum_{i=1}^{M} y_i^k \, log(\widehat{y}_i^k), \quad (10)$$

where y_i^k is the actual label for i-th sample and $\widehat{y}_i^k$ is the predicted probability of i-th sample for the class k, M is the total number of samples and K is the number of classes. Note that various sleep datasets are imbalanced, *i.e.*, the amount of data for each class varies a lot. The loss function in Equation 10 equally penalizes the miss-classification of all the classes, and thus the trained model may be biased towards the majority classes.

We propose a class-aware loss function to address the above issue, which uses a weighted cross-entropy loss as follows:

$$\mathcal{L} = -\frac{1}{M} \sum_{k=1}^{K} \sum_{i=1}^{M} w_k \, y_i^k \, log(\widehat{y}_i^k), \quad (11)$$

$$w_k = \mu_k \cdot max(1, log(\mu_k \, M/M_k)), \quad (12)$$

where w_k represents the weight assigned to the class k, μ_k is a tunable parameter, and M_k is the number of samples in class k.

The choice of the class weight w_k relies on *two factors i.e.* the number of samples of this class (controlled by M/M_k), and the distinctness of this class (controlled by μ_k). With analyzing the public sleep data, we can reach out to two conclusions. First, class N2 has a large number of samples, while classes N1 and N3 have much fewer number of samples (*i.e.*, N1 and N3 are minority classes). Second, we observe that signals of N3 have significantly higher magnitude than other classes as shown in Fig. 4. Hence, the model can easily make correct predictions for N3 samples. Meanwhile, N1 samples are not distinguishable from those in classes N2 and REM as shown in Fig. 4. Therefore, we assign the highest μ_k to N1, the lowest to N3 and assign similar values to the other three classes W, N2 and REM as follows.

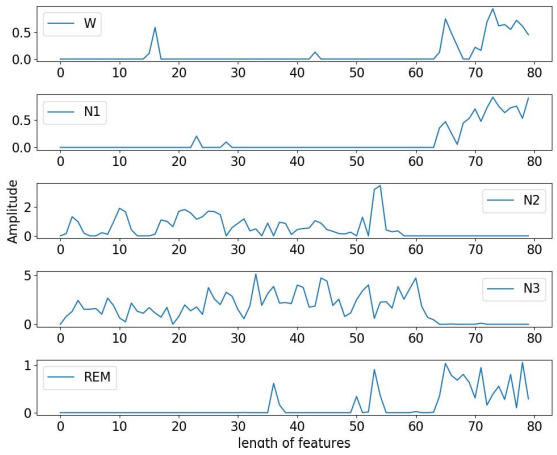

Fig. 4. The amplitude of the extracted features differs among 5 classes. This snapshot is from Sleep-EDF-20 dataset [36].

$$\mu_k = \begin{cases} a/K & k = N3 \\ b/K & k = W, N2, REM \\ c/K & k = N1 \end{cases}$$

where a, b, c are hyperparameters that change for each dataset. To fulfill the above recommendation, we chose $a < b < c$. Note that we use μ_k to scale down the M/M_k value so we keep the values of μ_k less than 1 by dividing them by K. As such, the values of a, b, c are better to be kept less than K to scale down the weights. Additional experiments on the effect of the different variants of a, b, and c values are provided in Section S.II the supplementary materials.

Finally, we use Adam [35] as the optimizer to minimize our class-aware loss in Equation 11 and learn model parameters.

III. Experimental Results

In this section, we first introduce the experimental setup. Then, we demonstrate the evaluation results of our proposed AttnSleep.

A. Datasets and Evaluation Metrics

In our experiments, we used three public datasets, namely, Sleep-EDF-20, Sleep-EDF-78 and Sleep Heart Health Study (SHHS) as shown in Table I. For each dataset, we used a single EEG channel for various models in our experiments.

Sleep-EDF-20 and Sleep-EDF-78 were obtained from the PhysioBank [36]. Sleep-EDF-20 contains data files for

TABLE I
DETAILS OF THREE DATASETS USED IN OUR EXPERIMENTS (EACH SAMPLE IS A 30-SECOND EPOCH)

Datasets	#Subjects	EEG Channel	Sampling Rate	W	N1	N2	N3	REM	#Total Samples
Sleep-EDF-20	20	Fpz-Cz	100 Hz	8285 *19.6%*	2804 *6.6%*	17799 *42.1%*	5703 *13.5%*	7717 *18.2%*	42308
Sleep-EDF-78	78	Fpz-Cz	100 Hz	65951 *33.7%*	21522 *11.0%*	69132 *35.4%*	13039 *6.7%*	25835 *13.2%*	195479
SHHS	329	C4-A1	125 Hz	46319 *14.3%*	10304 *3.2%*	142125 *43.7%*	60153 *18.5%*	65953 *20.3%*	324854

20 subjects, while Sleep-EDF-78 is an expanded version with 78 subjects. The participants were involved in two studies. The first is Sleep Cassette (SC* files), which studies age effects on sleep and it was conducted on healthy participants aged from 25 to 101 years. The second is Sleep Telemetry (ST* files), which addressed the temazepam effects on sleep in 22 Caucasian males and females without having any other medication. For these two datasets, each PSG file contains two EEG channels (Fpz-Cz, Pz-Oz) with a sampling rate of 100 Hz, one EOG channel and one chin EMG channel. Following previous studies [13], [15], [17], [18], [37], we adopted the data from Sleep Cassette study and used the single Fpz-Cz channel as the input for various models in our experiments.

SHHS [38], [39] is a multi-center cohort study of the cardiovascular and other consequences of sleep-disordered breathing. The subjects suffer from various diseases including lung diseases, cardiovascular diseases and coronary diseases. To minimize the impact of these diseases, we followed the study in [40] to select subjects, who are considered to have a regular sleep (e.g., Apnea Hypopnea Index or AHI less than 5). Eventually, 329 out of 6,441 subjects were selected for our experiments. Notably, we selected the C4-A1 channel with a sampling rate of 125 Hz. Further details about the datasets can be found in Section S.III in our supplementary materials. For the three datasets, we applied the following preprocessing steps. First, we excluded any UNKNOWN stages that don't belong to any of the sleep stages. Second, we merged stages N3 and N4 into one stage (N3) according AASM standard. Third, we include only 30 minutes of wake periods before and after the sleep periods to add more focus on the sleep stages [20].

We adopted four metrics to evaluate the performance of various models for sleep stage classification, namely, the accuracy (ACC), macro-averaged F1-score (MF1), Cohen Kappa (κ) [41], and the macro-averaged G-mean (MGm). Both MF1 and MGm are common metrics to evaluate the performance of the models on imbalanced datasets [42]. Given the True Positives (TP_i), False Positives (FP_i), True Negatives (TN_i) and False Negatives (FN_i) for the i-th class, the overall accuracy ACC, MF1 and MGm are defined as follows.

$$ACC = \frac{\sum_{i=1}^{K} TP_i}{M}, \tag{13}$$

$$MF1 = \frac{1}{K}\sum_{i=1}^{K} \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}, \tag{14}$$

TABLE II
CONFUSION MATRIX OF PROPOSED MODEL APPLIED ON FPZ-CZ CHANNEL FROM EDF-20 DATASET

	Predicted					Per-class metrics			
	W	N1	N2	N3	REM	PR	RE	F1	GM
W	7432	437	109	24	283	89.6	89.7	89.7	93.5
N1	358	1097	593	6	750	47.1	39.1	42.8	61.6
N2	288	308	15769	493	941	89.1	88.6	88.8	90.3
N3	33	1	535	5119	15	90.7	89.8	90.2	94.1
REM	184	485	702	4	6342	76.1	82.2	79.0	88.0

TABLE III
CONFUSION MATRIX OF PROPOSED MODEL APPLIED ON FPZ-CZ CHANNEL FROM EDF-78 DATASET

	Predicted					Per-class metrics			
	W	N1	N2	N3	REM	PR	RE	F1	GM
W	60545	3607	565	57	1177	92.3	91.8	92.0	93.9
N1	3481	8443	6559	102	2937	45.3	39.2	42.1	60.8
N2	565	3712	59800	2027	3028	83.5	86.5	85.0	88.5
N3	31	12	2276	10686	34	82.3	82.0	82.1	90.0
REM	984	2849	2440	106	19456	73.1	75.3	74.2	85.0

TABLE IV
CONFUSION MATRIX OF PROPOSED MODEL APPLIED ON C4-A1 CHANNEL FROM SHHS DATASET

	Predicted					Per-class metrics			
	W	N1	N2	N3	REM	PR	RE	F1	GM
W	38604	2798	2188	304	2425	90.3	83.3	86.7	90.6
N1	856	3747	2385	46	3270	30.6	36.7	33.2	59.5
N2	1675	2823	123242	6891	7494	87.4	86.7	87.1	88.5
N3	214	19	7051	52695	174	87.5	87.6	87.5	92.3
REM	1405	2858	6143	308	55239	80.5	83.8	82.1	89.1

$$MGm = \frac{1}{K}\sum_{i=1}^{K} \sqrt{Specificity_i \times Recall_i}, \tag{15}$$

where $Precision_i = \frac{TP_i}{TP_i+FP_i}$, $Recall_i = \frac{TP_i}{TP_i+FN_i}$ and $Specificity_i = \frac{TN_i}{TN_i+FP_i}$. M is the total number of samples and K is the number of classes or sleep stages.

We also used per-class precision (PR), per-class recall (RE), per-class F1-score (F1), and per-class G-mean (GM) to evaluate each our model. They are calculated as in binary classification by considering one class as the positive class and the other four classes as the negative class.

B. Scoring Performance of AttnSleep

Tables II, III and IV show the confusion matrices of the proposed model applied on the Fpz-Cz channel in both

TABLE V
COMPARISON AMONG ATTNSLEEP AND STATE-OF-THE-ART MODELS. THE BEST VALUES ON EACH DATASET ARE HIGHLIGHTED IN BOLD

Dataset	Method	Per-Class F1-score					Overall Metrics				Avg Training time / fold
		W	N1	N2	N3	REM	Accuracy	MF1	κ	MGm	
Sleep-EDF-20	DeepSleepNet [20]	86.7	**45.5**	85.1	83.3	**82.6**	81.9	76.6	0.76	**86.9**	2.5 hrs
	SleepEEGNet [24]	89.4	44.4	84.7	84.6	79.6	81.5	76.6	0.75	85.3	1.5 hrs
	ResnetLSTM [43]	86.5	28.4	87.7	89.8	76.2	82.5	73.7	0.76	81.8	1.2 hrs
	MultitaskCNN [17]	87.9	33.5	87.5	85.8	80.3	83.1	75.0	0.77	83.1	2.6 hrs
	AttnSleep *(ours)*	**89.7**	42.6	**88.8**	**90.2**	79.0	**84.4**	**78.1**	**0.79**	85.5	**21 mins**
Sleep-EDF-78	DeepSleepNet [20]	90.9	**45.0**	79.2	72.7	71.1	77.8	71.8	0.70	81.6	7.2 hrs
	SleepEEGNet [24]	89.8	42.1	75.2	70.4	70.6	74.2	69.6	0.66	82.3	4.6 hrs
	ResnetLSTM [43]	90.7	34.7	83.6	80.9	67.0	78.9	71.4	0.71	80.8	3.4 hrs
	MultitaskCNN [17]	90.9	39.7	83.2	76.6	73.5	79.6	72.8	0.72	82.5	5.3 hrs
	AttnSleep *(ours)*	**92.0**	42.0	**85.0**	**82.1**	**74.2**	**81.3**	**75.1**	**0.74**	**83.6**	**1.7 hrs**
SHHS	DeepSleepNet [20]	85.4	**40.5**	82.5	79.3	81.9	81.0	73.9	0.73	82.6	14.4 hrs
	SleepEEGNet [24]	81.3	34.4	73.4	75.9	77.0	73.9	68.4	0.65	82.7	6.4 hrs
	ResnetLSTM [43]	85.1	9.4	86.3	87.0	79.1	83.3	69.4	0.76	76.4	5.2 hrs
	MultitaskCNN [17]	82.2	25.7	83.9	83.3	81.1	81.4	71.2	0.74	80.4	6.2 hrs
	AttnSleep *(ours)*	**86.7**	33.2	**87.1**	**87.1**	**82.1**	**84.2**	**75.3**	**0.78**	**84.0**	**2.1 hrs**

Sleep-EDF datasets and on C4-A1 channel in SHHS dataset. The confusion matrix is calculated by adding up all the scoring values of the testing data through the 20 folds. Each row represents the number of samples classified by experts, while each column represents the number of epochs predicted by our model. The tables also show the per-class precision, recall, F1 score and G-mean value for each class.

Notably, stage N1 achieves the lowest performance with the F1 less than 50%, where it is often misclassified to classes W, REM and N2. In counterpart, class N3 achieves the best performance for Sleep-EDF-20 and SHHS datasets, but it decreases for Sleep-EDF-78 as it is the minority class on this dataset. Most of the misclassifications in the different datasets are with class N2 as it is the majority class.

C. Baselines and Experimental Setup

In our experiments, we compared our model with five baselines, namely, DeepSleepNet [20], SleepEEGNet [24], ResnetLSTM [43], MultitaskCNN [17] and SeqSleepNet [37]. Brief descriptions for each baseline are as follows.

- **DeepSleepNet** [20] exploits a custom CNN architecture followed by an LSTM with a residual connection for sleep stage classification.
- **SleepEEGNet** [24] employs the same CNN architecture as DeepSleepNet [20] followed by an encoder-decoder with attention mechanism.
- **ResnetLSTM** [43] implements a ResNet architecture for feature extraction, followed by an LSTM to classify EEG signals into different sleep stages.
- **MultitaskCNN** [17] starts by converting the raw EEG signals into power spectrum images, and then applies a joint classification and prediction technique using a multi-task CNN architecture for identifying sleep stages.
- **SeqSleepNet** [37] also converts the raw EEG signal into power spectrum images and then uses a hierarchical RNN structure to classify multiple epochs at once.

In particular, we used the published codes for DeepSleepNet [20], SleepEEGNet [24], MultitaskCNN [17] and SeqSleepNet [37], and re-implemented ResnetLSTM [43]. To evaluate the performance of various models, we adopted a subject-wise 20-fold cross-validation by dividing the subjects in each dataset into 20 groups. For example, subject-wise 20-fold cross-validation on Sleep-EDF-20 dataset with 20 subjects is thus leave-one-subject-out (LOSO) cross-validation. For each round, we selected one group of subjects as testing data and the remaining 19 groups as training data. Eventually we combined the predicted sleep stages for the testing samples from all the 20 rounds to calculate various performance metrics. We chose the neural network hyperparameters based on the performance across these folds. In addition, we noticed that AttnSleep performance stabilizes before reaching 100 epochs, so we trained all the models for 100 epochs in each round to fairly compare their average training time. Fig. 5 shows the performance graph of our model showing both the loss and accuracy during AttnSleep training. Our model shows a stable performance during training, and we notice that it converges quickly. Additionally, it can be seen that the validation loss stabilizes even with the continual decrement in the training loss, which reflects the robustness of our model against overfitting.

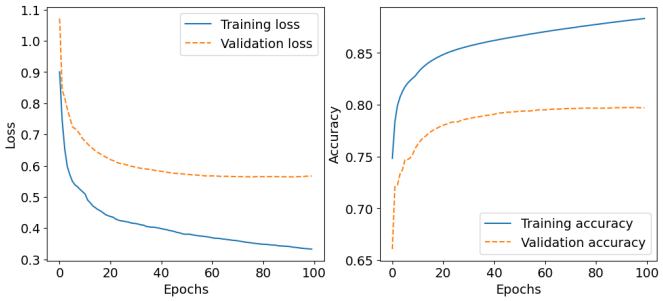

Fig. 5. Training and testing accuracy and loss comparison for a random fold (*i.e.* fold 10 on subject 16) in Sleep-EDF-20 dataset.

We built our model using PyTorch 1.4 and trained it on a Tesla K40 GPU. We applied a batch size of 128, and the Adam optimizer with the learning rate starting with 1e-3 then

reducing to 1e-4 after 10 epochs. The weight decay of Adam was set to 1e-3, the betas ($b1$, $b2$) were used as (0.9, 0.999) respectively, the epsilon value was 1e-08 and the amsgrad was set to true. All the convolutional layers were initialized using a Gaussian distribution with a mean of 0 and a variance of 0.02. For the TCE module, we used 5 heads in MHA and the dimension of each feature d was 80 for Sleep-EDF dataset, and 100 for SHHS dataset because of its higher sampling rate, and hence its longer signal length. For the two fully connected layers, the input dimension was d and the output dimension was set to 120, and vice versa for the second fully connected layer. A detailed description can also be found in Table S.3 in our supplementary materials. Our source codes and supplementary materials are publicly available at https://github.com/emadeldeen24/AttnSleep.

D. Comparison With State-of-the-Art Approaches

We evaluated the performance of our AttnSleep model against various state-of-the-art approaches. We compared their performance in terms of overall accuracy, macro F1-score, cohen kappa, macro G-mean and the average training time on three datasets.

Table V shows the comparison among DeepSleepNet [20], SleepEEGNet [24], ResnetLSTM [43], MultitaskCNN [17] and our AttnSleep. We observe that our AttnSleep achieves better classification performance than the other four approaches, due to its powerful feature extraction module as well as the TCE with attention mechanism. In particular, our AttnSleep achieves better MF1 and MGm on Sleep-EDF-78 and SHHS, indicating that the designed cost-sensitive loss function is helpful to handle imbalanced data. In addition, we can observe that our AttnSleep achieves lower performance for class N1 than [20], [24]. As shown in Fig. 4, W, REM and N1 have similar features in our framework. Therefore, our AttnSleep tends to misclassify N1 as other classes including W and REM, which is also demonstrated in the confusion matrices in Tables II, III and IV.

Note that all the five methods in Table V use a single epoch (*i.e.*, 30-second EEG signal) as the model input. Differently, SeqSleepNet [37] takes 3 epochs as input and then predicts the label for the middle epoch. For fair comparison, we compare our AttnSleep with SeqSleepNet in Table VI by using 3 epochs as input. As shown in Table VI, our AttnSleep outperforms SeqSleepNet in terms of all the four metrics (ACC, MF1, κ and MGm). By comparing Tables V and VI, we also observe that using more epochs as input includes more temporal relations and helps our AttnSleep model to achieve better performance.

In addition, the training time of our method is much less than other methods as shown in Tables V and VI. First, DeepSleepNet [20], SleepEEGNet [24] and SeqSleepNet [37] all exploit LSTMs which slow down the training due to the recurrent processing in LSTM. Second, MultitaskCNN [17] and SeqSleepNet [37] require additional computation to pre-train a DNN-based filter bank before training the main model. Differently, our AttnSleep model captures the temporal dependency among EEG data using TCE instead of LSTM, and can thus benefit from parallel computation to achieve the reduced training complexity.

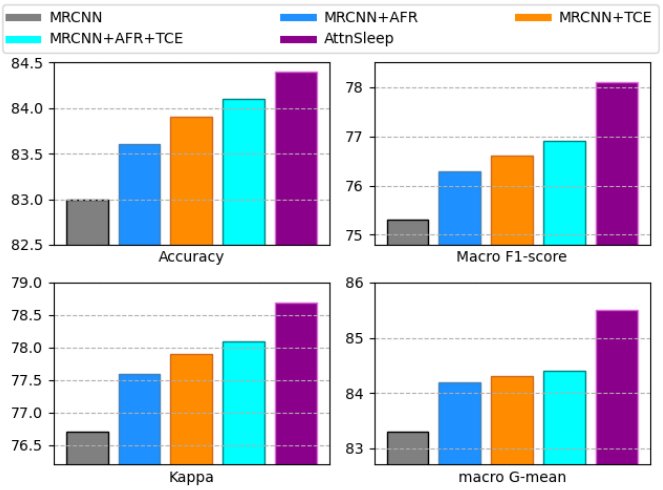

Fig. 6. Ablation study conducted on Sleep-EDF-20 dataset.

E. Ablation Study

Note that our AttnSleep consists of MRCNN, AFR and TCE modules together with the class-aware loss function. To analyze the effectiveness of each module in our AttnSleep, we present an ablation study conducted on Sleep-EDF-20 dataset as shown in Fig. 6. Specifically, we derive five model variants as follows and the first four variants do not use the class-aware loss function.

1) **MRCNN**: MRCNN module only.
2) **MRCNN+AFR**: MRCNN and AFR without TCE.
3) **MRCNN+TCE**: MRCNN and TCE without AFR.
4) **MRCNN+AFR+TCE**: training MRCNN, AFR and TCE together, without the class-aware loss function.
5) **AttnSleep**: training MRCNN, AFR and TCE together, with the class-aware loss function.

We can draw the following three conclusions based on the ablation study as shown in Fig. 6. First, AFR can enhance the classification performance, which demonstrates the necessity of modeling the feature inter-dependencies. This is further demonstrated by comparing the third and fourth variants (*i.e.*, MRCNN+TCE vs. MRCNN+AFR+TCE). Second, by comparing MRCNN and MRCNN+TCE (similarly MRCNN+AFR vs. MRCNN+AFR+TCE), we conclude that capturing the temporal dependencies with TCE is important for sleep stage classification. Moreover, TCE is even more important than AFR as MRCNN+TCE outperforms MRCNN+AFR. Third, AttnSleep achieves significantly better MF1 and MGm than other four variants, indicating that the proposed class-aware cost-sensitive loss function can effectively address the data imbalance issue without any added computation overhead. We also conduct the ablation study for Sleep-EDF-78 and SHHS dataset, which can be found in Fig. S.3 and S.4 in our supplementary materials.

F. Sensitivity Analysis for the Number of Heads in MHA

As MHA is one key component of our model, it is important to study how the number of heads affects the model performance. In particular, we fix the other parameters and

TABLE VI
COMPARISON OF THE PERFORMANCE OF ATTNSLEEP AGAINST SEQSLEEPNET WITH 3 EPOCHS AS INPUT

Dataset	Method	Per-Class F1-score					Overall Metrics				Avg Training time / fold
		W	N1	N2	N3	REM	Accuracy	MF1	κ	MGm	
Sleep-EDF-20	SeqSleepNet [37]	87.7	43.8	88.2	86.5	84.0	84.6	78.0	0.79	85.3	2.5 hrs
	AttnSleep *(ours)*	**90.3**	**47.9**	**89.8**	**89.0**	**85.0**	**85.6**	**80.9**	**0.80**	**88.2**	**31 mins**
Sleep-EDF-78	SeqSleepNet [37]	91.8	46.0	85.0	77.5	81.0	82.6	76.3	0.76	84.3	7.3 hrs
	AttnSleep *(ours)*	**92.6**	**47.4**	**85.5**	**83.7**	**81.5**	**82.9**	**78.1**	**0.77**	**85.6**	**1.9 hrs**
SHHS	SeqSleepNet [37]	84.2	**47.3**	87.2	85.4	**88.6**	85.6	78.5	0.80	85.4	15.2 hrs
	AttnSleep *(ours)*	**88.3**	46.3	**88.7**	**87.6**	87.4	**86.6**	**79.7**	**0.81**	**87.9**	**2.9 hrs**

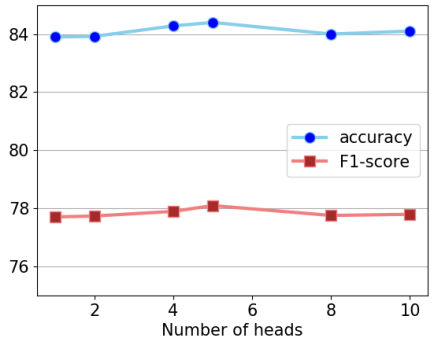

Fig. 7. The performance of our AttnSleep model on Sleep-EDF-20 dataset by using different number of heads in MHA.

test different number of heads in MHA. Note that the number of heads should be dividable by the length of features d. As d is 80 for Sleep-EDF-20 dataset, we run our model using 1, 2, 4, 5, 8 and 10 heads. Fig. 7 shows the model performance on Sleep-EDF-20 dataset in terms of accuracy and MF1 score. Overall, the model performance is quite stable when we use different number of heads. With increasing the number of heads from 1, 2 to 4 and 5, we can observe a slight improvement on the performance, since using more heads allows the model to find more meaningful features and feature interactions. Meanwhile, when the number of heads further increases (H equals to 8 and 10), *i.e.*, the length of features in each head becomes smaller, which leads to a slight performance decrease. In our experiments, we eventually set H as 5 on Sleep-EDF-20 dataset. For other two datasets, we also set H as 5, and the detailed sensitivity analysis can be found in Fig. S.5 and S.6 in our supplementary materials.

IV. CONCLUSION

We proposed a novel architecture for sleep stage classification from single channel raw EEG signals called AttnSleep. The AttnSleep relies on extracting the features from EEG signals using two modules: the multi-resolution convolutional neural network (MRCNN) and the adaptive feature recalibration (AFR). These two modules are followed by the temporal context encoder (TCE) module, which captures the temporal dependencies among the extracted features by using a multi-head attention (MHA) mechanism. We also proposed a class-aware cost-sensitive loss function to handle the issue of data imbalance. The experimental results on three public datasets demonstrated that our model outperforms state-of-the-art methods under various evaluation matrices. Besides, an ablation study was performed to show the effectiveness of each module in the proposed method. Finally, we conducted a sensitivity analysis to demonstrate the impact of the number of heads in MHA. The results indicated that our method is quite stable with different number of heads. For future directions, we will consider transfer learning and domain adaptation techniques, which adapt the model trained on labeled dataset to classify the unlabeled sleep data in other datasets.

REFERENCES

[1] F. S. Luyster, P. J. Strollo, P. C. Zee, and J. K. Walsh, "Sleep: A health imperative," *Sleep*, vol. 35, no. 6, pp. 727–734, Jun. 2012.
[2] P. H. Finan *et al.*, "Partial sleep deprivation attenuates the positive affective system: Effects across multiple measurement modalities," *Sleep*, vol. 40, no. 1, pp. 1–9, Jan. 2017.
[3] G. Rauchs, B. Desgranges, J. Foret, and F. Eustache, "The relationship between memory systems and sleep stages," *J. Sleep Res.*, vol. 14, no. 2, pp. 123–140, 2005.
[4] S. Sharma and M. Kavuru, "Sleep and metabolism: An overview," *Int. J. Endocrinol.*, vol. 2010, pp. 1–12, Oct. 2010.
[5] J. Tank *et al.*, "Relationship between blood pressure, sleep K-complexes, and muscle sympathetic nerve activity in humans," *Amer. J. Physiol.-Regulatory, Integrative Comparative Physiol.*, vol. 285, no. 1, pp. R208–R214, Jul. 2003.
[6] A. S. Keenan, "An overview of polysomnography," in *Handbook of Clinical Neurophysiology*, vol. 6. Amsterdam, The Netherlands: Elsevier, 2005, ch. 3, pp. 33–50.
[7] P. Memar and F. Faradji, "A novel multi-class EEG-based sleep stage classification system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 1, pp. 84–95, Jan. 2018.
[8] S. I. Dimitriadis, C. Salis, and D. Linden, "A novel, fast and efficient single-sensor automatic sleep-stage classification based on complementary cross-frequency coupling estimates," *Clin. Neurophysiol.*, vol. 129, no. 4, pp. 815–828, Apr. 2018.
[9] G. Zhu, Y. Li, and P. Wen, "Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 6, pp. 1813–1821, Nov. 2014.
[10] S. Seifpour, H. Niknazar, M. Mikaeili, and A. M. Nasrabadi, "A new automatic sleep staging system based on statistical behavior of local extrema using single channel EEG signal," *Expert Syst. Appl.*, vol. 104, pp. 277–293, Aug. 2018.
[11] X. Li, L. Cui, S. Tao, J. Chen, X. Zhang, and G.-Q. Zhang, "HyCLASSS: A hybrid classifier for automatic sleep stage scoring," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 2, pp. 375–385, Mar. 2018.
[12] A. R. Hassan and M. I. H. Bhuiyan, "Computer-aided sleep staging using complete ensemble empirical mode decomposition with adaptive noise and bootstrap aggregating," *Biomed. Signal Process. Control*, vol. 24, pp. 1–10, Feb. 2016.
[13] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," 2016, *arXiv:1610.01683*. [Online]. Available: http://arxiv.org/abs/1610.01683

[14] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, and J.-F. Payen, "A convolutional neural network for sleep stage scoring from raw single-channel EEG," *Biomed. Signal Process. Control*, vol. 42, pp. 107–114, Apr. 2018.
[15] M. Sokolovsky, F. Guerrero, S. Paisarnsrisomsuk, C. Ruiz, and S. A. Alvarez, "Deep learning for automated feature discovery and classification of sleep stages," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 6, pp. 1835–1845, Nov. 2020.
[16] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 758–769, Apr. 2018.
[17] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1285–1296, May 2019.
[18] F. Li *et al.*, "End-to-end sleep staging using convolutional neural network in raw single-channel EEG," *Biomed. Signal Process. Control*, vol. 63, Jan. 2021, Art. no. 102203.
[19] N. Michielli, U. R. Acharya, and F. Molinari, "Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals," *Comput. Biol. Med.*, vol. 106, pp. 71–81, Mar. 2019.
[20] A. Supratak, H. Dong, C. Wu, and Y. Guo, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 11, pp. 1998–2008, Nov. 2017.
[21] Z. Chen, M. Wu, W. Cui, C. Liu, and X. Li, "An attention based CNN-LSTM approach for sleep-wake detection with heterogeneous sensors," *IEEE J. Biomed. Health Informat.*, early access, Jun. 30. 2020, doi: 10.1109/JBHI.2020.3006145.
[22] Z. Chen *et al.*, "A novel ensemble deep learning approach for sleep-wake detection using heart rate variability and acceleration," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, Jun. 8, 2020, doi: 10.1109/TETCI.2020.2996943.
[23] A. Malafeev *et al.*, "Automatic human sleep stage scoring using deep neural networks," *Frontiers Neurosci.*, vol. 12, p. 781, Nov. 2018.
[24] S. Mousavi, F. Afghah, and U. R. Acharya, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLoS ONE*, vol. 14, no. 5, May 2019, Art. no. e0216456.
[25] T. Zhu, W. Luo, and F. Yu, "Convolution- and attention-based neural network for automated sleep stage classification," *Int. J. Environ. Res. Public Health*, vol. 17, no. 11, p. 4152, Jun. 2020.
[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[27] W. Huang, J. Cheng, Y. Yang, and G. Guo, "An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis," *Neurocomputing*, vol. 359, pp. 77–92, Sep. 2019.
[28] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 421–425.
[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.
[31] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853
[32] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[33] J. Devlin, M.-W. Chang, K. Lee, and N. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 4171–4186.
[34] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[36] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.
[37] H. Phan, F. Andreotti, N. Cooray, O. Y. Chen, and M. De Vos, "SeqSleepNet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 3, pp. 400–410, Mar. 2019.
[38] G.-Q. Zhang *et al.*, "The national sleep research resource: Towards a sleep data commons," *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 10, pp. 1351–1358, Oct. 2018.
[39] S. F. Quan *et al.*, "The sleep heart health study: Design, rationale, and methods," *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.
[40] P. Fonseca, N. den Teuling, X. Long, and R. M. Aarts, "Cardiorespiratory sleep stage detection using conditional random fields," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 4, pp. 956–966, Jul. 2017.
[41] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.
[42] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 39, no. 1, pp. 281–288, Feb. 2009.
[43] Y. Sun, B. Wang, J. Jin, and X. Wang, "Deep convolutional network method for automatic sleep stage classification based on neurophysiological signals," in *Proc. 11th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2018, pp. 1–5.