Multimedia Tools and Applications (2025) 84:1985–2003
https://doi.org/10.1007/s11042-024-19118-7

EASM: An efficient AttnSleep model for sleep Apnea detection from EEG signals

Madan Singh[1] · Sujata Chauhan[2] · Anil Kumar Rajput[3] · Indu Verma[1] · Alok Kumar Tiwari[3]

Received: 8 August 2023 / Revised: 3 January 2024 / Accepted: 22 March 2024 /
Published online: 18 April 2024

Abstract
This paper addresses the crucial task of automatic sleep stage classification to assist sleep experts in diagnosing sleep disorders such as sleep apnea and insomnia. The proposed solution presents a novel attention-based deep learning model called, Efficient Attention-sleep Model (EASM), designed specifically for sleep apnea detection using EEG signals. EASM incorporates a streamlined architecture that includes a modified Muti-Resolution Convolutional Neural Network (MRCNN), Adaptive Feature Recalibration (AFR), and a simplified Temporal Context Encoder (TCE) module to reduce complexity. To mitigate overfitting, ridge regression is utilized, which incorporates a penalty term to enhance model generalization. Furthermore, the proposed EASM utilizes a class-balanced focal loss function to address data imbalance issues. The effectiveness of EASM is evaluated on two publicly available datasets, SLEEP EDF-20 and SLEEP EDF-78. Comparative analysis of EASM against state-of-the-art models demonstrates its superior performance in terms of accuracy, training time, and model complexity. Notably, the proposed model achieves a 50% reduction in training time and a 55.7% decrease in complexity compared to the Attnsleep model. The EASM achieves a classification accuracy of 85.8% with minimum loss when compared to the Attnsleep model.

Keywords Multi-head attention · CNN · Temporal context encoder · Focal loss

✉ Anil Kumar Rajput
rajputanilkumar@gmail.com

Madan Singh
madan.singh@christuniversity.in

Sujata Chauhan
sujatachauhan06@gmail.com

Indu Verma
indu.verma@christuniversity.in

Alok Kumar Tiwari
alok.tiwari243@gmail.com

[1] Christ (Deemed to be University), Delhi-NCR Campus, India

[2] Dronacharya Govt. College, Gurugram, India

[3] ABV-Indian Institute of Information Technology & Management, Gwalior, India

1 Introduction

Sleep is an essential part of human life. It is critical to human health since it affects so many aspects of daily life. Those who get adequate sleep, according to a study in [1], live a healthier lifestyle. Insufficient sleep, on the other hand, contributes to disorders such as insomnia and sleep apnea [2–4]. The ability to monitor people's sleeping habits significantly impacts medical practice and research. Sleep stages (i.e., deep and light sleep) are essential for metabolism, memory, and the immune system [5]. Consequently, it is essential to monitor sleep using the classification of sleep stages [6]. The electroencephalogram (EEG) signals are extracted and divided into short epochs then each epoch is analyzed by a specialist to classify sleep stages based on American Academy of Sleep Medicine (AASM) standards [7]. Automatically classification of sleep stages using a deep learning model is important as it helps sleep specialists who do it manually. Polysomnography (PSG), which comprises Electroencephalograms (EEG), Electrooculograms (EOG), Electromyograms (EMG), and Electrocardiograms (ECG) [8], is commonly used by sleep professional to identified sleep stages. Recent years have seen an increase in the use of EEG signals with only one channel due to their simplicity. EEG signals with a single channel are typically recorded in 30-second epochs.

Sleep stages : The sleep stage is divided into five major classes of sleep classification. They are wake, three normal classes are N1, N2 (the class with the most weightage), and N3 and Rapid Eye Movement (REM). Almost 75% of sleep is in the REM stage, and the majority of sleep is in the N2 stage. Each of the following phases of sleep—N1, N2, N3, N2, REM—is progressed through during the course of four to five sleep cycles. The average sleep cycle lasts around 90 to 110 minutes. Initial REM periods are brief; as the night passes, there are longer REM periods and a reduction in deep sleep [6].

1. **Wake** It is the first stage which depends on whether the eyes are closed or not. The EEG signal of this stage has the lowest amplitude and highest frequency, labeled as beta waves.
2. **N1 Light Sleep (5%)** The EEG records theta waves in this stage with low voltage. Skeletal muscle has muscular tone, and breathing occurs at a regular rate. This stage typically lasts 1 to 5 minutes and contributes about 5% of total sleep time.
3. **N2 Deep Sleep (45%)** As the body temperature and pulse rate decrease gradually, this stage symbolizes deeper slumber. This stage's EEG recordings include the K complex and sleep spindles. K-complexes are the longest and most apparent of all brain waves, lasting around one second. K-complexes have been shown to aid in the maintenance of sleep and memory consolidation. Every cycle, this stage lasts 25 minutes and contributes to 45 percent of total sleep.
4. **N3 deepest non-REM sleep** At this point, the EEG recordings are delta waves with the lowest frequency and maximum amplitude. Because of the low-frequency waves, this is also known as slow-wave sleep.
5. **Rapid Eye Movement (REM)** Except for the eyes and diaphragmatic breathing muscles, the skeletal muscles are atomic and immobile, even though the EEG resembles an awake person's. Dreaming is linked to REM sleep, which is not considered peaceful. The skeleton muscles, which are fixed, having eyes and breathing through the diaphragm as an exception which continue their operation even when the EEG signal resembles that a person is completely awake. However, the breathing pattern becomes more unpredictable. When we fall asleep, this stage generally begins within 90 minutes of falling asleep. Normal initial periods last 10 minutes, while second periods last for up to an hour. A penile/clitoral tumescence, nightmare, and dreaming condition is known as REM.

The manual procedure of sleep stage classification, in which sleep specialists obtain split samples of 30-second EEG signals and classify them, is sophisticated and time-consuming. Therefore, computerized methods for classifying sleep stages are needed by sleep specialists. Many methods based on machine learning and deep learning have been offered as potential solutions. There are some issues in the previous models that need to be addressed. Imbalance data, over-fitting, and complexity of CNN models are the main issues associated with classifying sleep stages in previous models.

To address the overfitting and data imbalance problems in the previous model, In this paper, an efficient Attnsleep model with low complexity is proposed. First, a modified multi-resolution CNN (MRCNN) is used to reduce the complexity of the model. To successfully capture the temporal links in the extracted features, our model employs a single temporal context encoder (TCE) with multi-head attention and causal convolutions. Furthermore, we derive a focal loss function to automatically address the disparity in data. The ride regression technique is incorporated into the proposed model to solve the overfitting problems. We conducted extensive experiments on two publicly available datasets, and the findings show that the proposed EASM performs better than the state-of-the-art sleep stages classification model.

The key contribution of this work is as follows:

- The Efficient AttnSleep Model (EASM) with low complexity is proposed for sleep apnea detection using EEG signals.
- The focus loss function is presented as a solution to address the imbalance problem within the dataset successfully.
- Ride regression technique is incorporated in the loss function to solve the over-fitting issue in the EASM.

2 Related work

Traditional machine-learning algorithms have been used to classify sleep stages with the help of EEG signals. Feature extraction followed by classification are two main tasks for sleep stage detection using machine learning algorithms. First, The frequency and temporal domains are used to extract features for EEG signals. The best features are selected using feature selection algorithms. Second, the extracted features are then given as input in Classifiers like Support Vector Machine (SVM) [9], Random Forest [10], and Naive Bayes [11] to classify the sleep stage. However, domain expertise is needed to extract the most informative features from these approaches.

Deep learning [12, 13] has lately gained popularity since it is more efficient than traditional machine algorithms because it does not require domain knowledge. Many research has employed CNN [14, 15] for sleep stage classification. A deep learning network, DeepSleepNet, has been suggested in [16] for automatically evaluating sleep stages from raw single-channel EEG data. Automatic sleep staging using convolutional neural networks (CNNs) is proposed in [17], along with a joint classification and prediction framework that uses a simple but effective CNN architecture. The previous CNN models performed well when categorizing sleep stages [18]. However, most of them cannot adequately characterize time relationships between EEG data.

RNNs (Recurrent Neural Networks) have been used in sleep phase classification to limit temporal dependence in time-varying EEG signals. Some researchers have combined convolutional neural networks (CNNs) and recurrent neural networks (RNNs), using CNNs for

feature extraction and RNNs for modeling time dependencies. For example, the long-term, short-term memory (LSTM) [19] is used in CNN architecture for feature extraction from raw data (EEG signal). In addition, a system for focusing on the most important information in the input sequence was developed using LSTM as an encoder-decoder for temporal dependencies [20]. However, RNNs' high model complexity and repetitive nature make simultaneous training challenging. In place of RNN, attention mechanisms have been utilized in several works. For example, [21] employed self-attention to learn inter-epoch and intra-epoch temporal aspects by segmenting the EEG epochs. Apart from this, in [4], a self-attention model with multi-convolution layers for feature extraction and a multi-head attention architecture for inter-dependencies in temporal features has been proposed. This model outperforms all previous models but can be improved.

Data imbalance is another important thing noticed after selecting a model for classifying sleep stages. In addition to choosing classification models, sleep stages also need to address the data imbalance issue because people spend different amounts of time in each phase. Oversampling is a common strategy to deal with this problem. Replicated minority classes used in the model training in [16]. The authors use the Synthetic Minority Over-sampling Technique (SMOTE) in [6] to over-sample the data to align it. However, Re-sampling methods increase the training data and lengthen the training period.

3 Architecture of efficient AttnSleep model

Figure 1 depicts the deep learning architecture of the Efficient AttnSleep model. The model mainly consists of a) Feature extraction, b) Temporal Context Encoder (TCE) Class balanced focal loss function, d) Ridge Regularization.

The Modified MRCNN module has two sets of CNN architecture (i.e., small kernel CNN and wide kernel CNN) for collecting features from the 30-second EEG recordings. Small kernel CNN and wide kernel CNN extract high and low-frequency signals for EEG, respectively. We specifically extract high-frequency characteristics by convolving a tiny kernel and low-frequency features by convolution with a broad kernel. Adaptive Feature ReCalibration (AFR) module is used to model inter-dependencies between extracted features from MRCNN. Furthermore, AFR may adaptively choose and highlight the most important elements, increasing classification performance. Next, a TCE module is created that captures the long-term dependencies in the input functions. TCE relies heavily on multi-headed attention, which is aided by causal convolution. In the third step, fully connected layers make the classification decision using softmax activation. This model's data imbalance problem is addressed through the introduction of a class-balanced focal loss function as a potential solution. Third, fully linked layers utilizing the softmax activation make the categorization choice. To overcome the overfitting issue, we have used ridge regularization that adds a

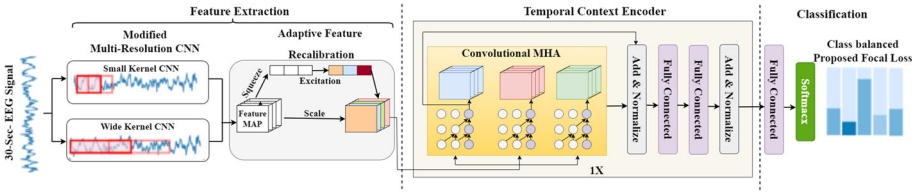

Fig. 1 The architecture of proposed Efficient AttnSleep model for automatic sleep stage detection

penalty term to reduce the overfitting problem in the proposed model. The next subsections will provide an in-depth introduction to each building block.

3.1 Multi resolution convolution neural network

We design a modified version of a multi-resolution CNN architecture, as depicted in Fig. 2, in order to extract many different sets of features. We use the convolution layer's two branches with varying kernel sizes to examine distinct frequency bands. It is motivated by previous work in which numerous CNN kernel sizes were used to extract features with varying frequencies (low and high frequencies). Additionally, the distinct stages of sleep are characterized by different frequency ranges [22] . As a result, dealing with numerous frequency bands in order to improve the features that have been recovered is becoming an increasingly significant practice. As a consequence of this, numerous kernel widths are utilized in order to record different time-step ranges in order to address various sleep-related frequency band features. To justify the selection of two different branch kernel widths, let's take into consideration data collection with a sample rate of 100 Hz (100-time steps recorded in seconds). First, the wide kernel (kernel size of 300) records time steps in a 3-second frame, recording entire sine wave cycles down to 0.3 Hz (T=1/F), Which corresponds to the delta band range. Second, each convolution window catches 25 samples (0.25 seconds) for the

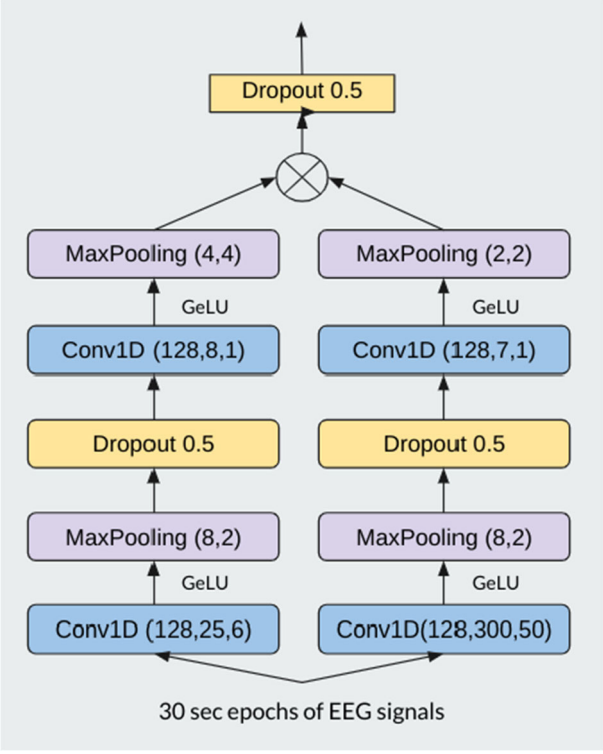

Fig. 2 Modified Multi-Resolution CNN for feature extraction from EEG epochs of 30 sec

smaller kernels (25 kernels size), allowing us to capture the whole cycle of a sinusoidal signal down to 4 Hz, which means this data corresponds in the delta and theta bands.

On the other hand, such feature combinations are required for the non-stationary nature of the EEG signal, and many features must be examined. Each branch, as depicted in Fig. 2, is made up of two convolution layers and two max pooling layers. A batch normalizing layer with a Gaussian error of linear unit (GELU) is included in each convolution layer. The use of a 1D convolution layer with 128 filters, a kernel size of 40, and a step size of 6 is indicated by the notation Conv1D(128, 40, 6). On the other hand, a max-pooling layer with a kernel size of 8 and a stride of 2 is referred to as a max-pooling layer with the notation "max-pooling (8, 2)." To prevent over-fitting, we additionally use dropout after initial max-pooling on both branches and after concatenating both branches, as illustrated in Fig. 2. As shown in Fig. 2, a number of parameters to be trained depends directly on the kernel size and filter size used in convolution. To reduce the complexity, we are using two layers of CNN in the branch, but this will lead to less number of features extracted to counter this we are using double the number of filters in the initial layer of the MRCNN module in both branches.

3.2 Adaptive feature recalibration

Adaptive Feature Recalibration (AFR) intends to fine-tune the features learned through MRCNN to improve its performance. In particular, the AFR uses a residual squeeze and excitation (residual SE) block to model the interdependencies between the features and

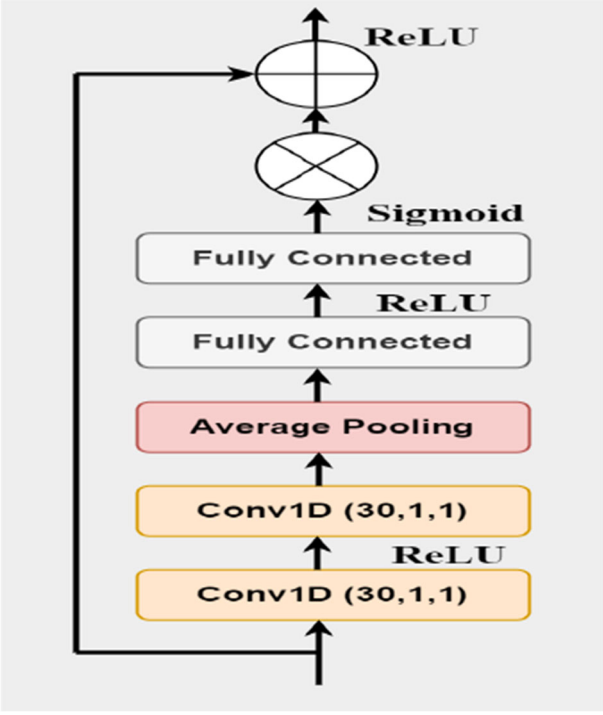

Fig. 3 Adaptive Feature Recalibration module

adaptively pick the most discriminative features [23]. The SE block offers a context-aware method that assists the network's lower levels in making better use of contextual information from beyond their immediate receptive field. This information might come from anywhere in the network. Two Conv1D(30,1,1) operations with kernel and stride sizes of 1 and ReLU as the activation function are used in the last SE block (see Fig. 3). After that Average pooling leads to a reduction in dimensionality as it averages all the values in the kernel to a single element in the output matrix. After that, two fully connected (FC) layers are applied in order to take advantage of the aggregated information. After the first layer is completed, a ReLU activation function is applied in order to decrease the dimension of the problem. After the second layer, a smoothing sigmoidal activation function is applied in order to limit the increase in dimensionality.

3.3 Temporal context encoder

The purpose of Temporal Context Encode (TCE) layers is to capture the temporal dependence of input features. The TCE layer is made up of Multi-Head Attention (MHA) layers, a normalization layer, and two FC levels, as shown in Fig. 1. Two identical structures are layered in TCE to get the final product. Given the importance of the Attention Mechanism to the TCE layer, we will start with the Self-Attention Mechanism and work our way through the rest of the module.

3.3.1 Multi head attention

The transformer model inspires Multi-Head Attention (MHA) [24] and has seen a tremendous boom in machine learning applications due to its capacity to learn long-range correlations in phrases. MHA encourages self-attention in two different ways.To begin, it enables the model to focus on a range of positions since the encodings of each head are aware of the encodings of the other heads. This makes it feasible for the model to concentrate on a number of positions. As a consequence of this, the capability of the model to learn temporal dependencies is enhanced. After that, the representation subspaces are also enlarged when the input features are partitioned. Because of this, the attention weights that were calculated for each subspace ended up more precisely reflecting the significance of each division, and the combined representations ended up being more accurate than the initial ones. This improves the accuracy with which the subspaces are classified. As a consequence of this, the attention weights that were ultimately determined for each subspace more correctly reflect the significance of each division, and the combined representations are more accurate than the initial ones (originally). This improves the accuracy with which the subspaces are classified. By employing causal convolutions, our Efficient AtnSleep sleep model is able to encapsulate the spatial relationships between input characteristics and detect temporal dependencies. Compared to RNNs, causal convolutions can handle data quickly and in parallel, which makes model training much faster.

As can be seen in Fig. 1, the output X of the AFR module is used as the input for the MHA module. To be more exact, MHA requires that each instance of X be inputted three times as shown in Fig. 4. To begin, the causal convolutions produce the value $\hat{X}$from the value X, which can be written as $\hat{X} = \phi(X)$. Second, in accordance with the [24], we use the three matrices that represent $\hat{X}$ to compute the attention in (1).

$$ATT(\hat{X}, \hat{X}, \hat{X}) = Softmax(\frac{\hat{X}\hat{X}^T}{\sqrt{d}}).X \tag{1}$$

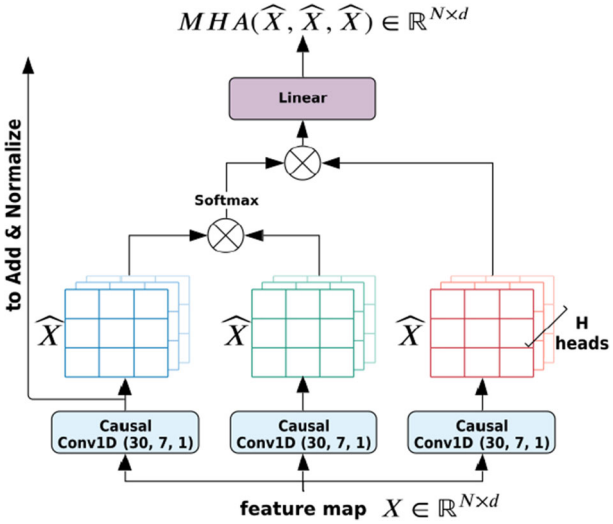

Fig. 4 Multi head attention module for capturing temporal dependencies

In the final step, all of the H representations are stringed together using concatenation to produce the following final output as given in (2):

$$MHA(\hat{X}, \hat{X}, \hat{X}) = Concat(A^1 .....A^h) \tag{2}$$

3.3.2 Add and normalise layer

The TCE has two layers of summing and normalization, with the previous layer's output being added to the input of this layer via the residual link and the sum being normalized. Layer Norm(x + SubLayer (x)) can be used to express this operation. Layer Norm refers to the layer normalization application, and SubLayer refers to either the MHA or the two FC layers as shown in Fig. 1, where x represents the input in SubLayer. Residual linkages enable models to exploit lower-layer features and transfer them to higher layers when appropriate. Furthermore, the normalization procedure speeds up the training process.

3.3.3 Feed forward Layer

The MHA layer's output is supplied into a feedforward neural network, which is made up of two FC layers. This layer decomposes model nonlinearities using the ReLU activation function, allowing interactions between latent dimensions. This behaviour can be modelled as $F_{out} = W4(\delta(W3(x))$ where W3 and W4 are shown in Fig. 1, refer to two FC layers within the TCE module

3.4 Class balance focal loss

It operates on the basis of computing an effective number of samples for each class, which is defined by (5):

$$E_n = \frac{1-\beta^n}{1-\beta} \tag{3}$$

Thus, the loss function is defined as:

$$CB(p,y) = \frac{1}{E_{n_y}} £(p,y) = (\frac{1-\beta}{1-\beta^n}) £(p,y) \tag{4}$$

As shown in Fig. 5 that N2 class consists of 42.1% while some minority class like N1 consists of just 6.6% this creates an imbalance, so we adopt the class balance focal loss function to reduce the loss due to data imbalance in our dataset as shown in Fig. 5. We use the cross-entropy loss function, which is enhanced by using the focal loss function as a loss now is inversely proportional to the effective number of samples in each class.

3.4.1 Ridge regularization

We found evidence of overfitting in the model, which means that although the model performed well on the testing samples, it did not generalize well at all. In order to resolve this matter, we will be utilizing Ridge regularisation, which will add a penalty term to the loss function. The L_2 Norm is utilized as a penalty term in ridge regularisation. We will use a value of 100 for the lambda parameter.

$$Loss = Error(Y - \widehat{Y}) + \lambda \sum_{1}^{n} w_i^2 \tag{5}$$

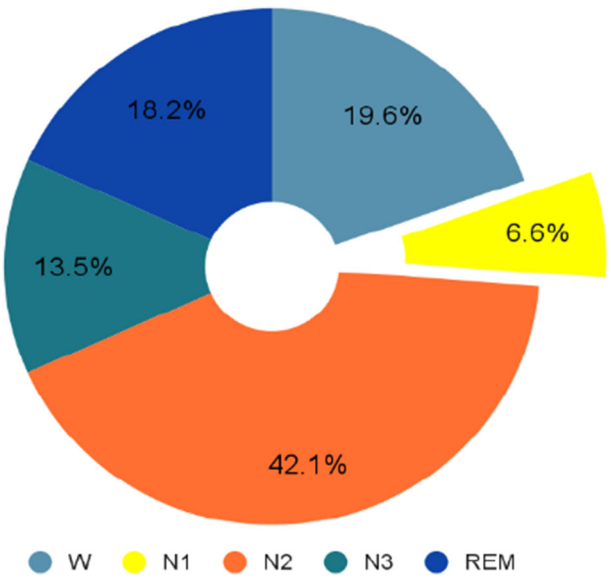

Fig. 5 Sleep stages classes: There is a data imbalance problem in the dataset

Where w is the weight matrix. This penalty term is added to the loss function. For implementation, it is added as an argument in Adam optimizer as a parameter.

4 Experimental results

In the section that follows, we will begin by describing the setup that will be used in the experiment. After that, we will present the findings of the examination of our proposed EASM.

4.1 Experimental setup

DeepSleepNet [6], MultitaskCNN [25], and Attnsleep model [4] are used to benchmark the proposed model. PyTorch version 2.3 was used in the construction of our model, and it was trained on a Nvidia RTX-5000 GPU. We used a batch size of 128, and the Adam optimizer. The learning rate began at 1e-3, then it decreased to 1e-4 after 10 iterations of training. A Gaussian distribution with zero mean and 0.02 standard deviations was used to initialize all the convolutional layers. We trained on the Sleep-EDF dataset with 5 MHA heads and 80 feature dimensions (d) for the TCE module.

4.2 Datasets and pre-processing

We used two public datasets in our tests, Sleep-EDF-20 and Sleep-EDF-78, as described in Table 1. Our research used one EEG channel for each dataset to test different models.

We have obtained Sleep-EDF-20 and Sleep-EDF-78 from PhysioBank [26]. There are 20 data files of subjects of Sleep-EDF-20 where as we have an extended version of, i.e., Sleep-EDF-78 containing 78 data files. There were two studies with many people involved. Sleep Cassette(SC* files) was one of the earliest studies done, and it documented the changes in sleep quality that occur between the ages of 25 and 101. In the second study, which involved the use of sleep telemetry (ST*files), the researchers were interested in determining how temazepam affected the quality of sleep experienced by 22 Kasian participants who were not already taking any other medications. Each PSG file for these two datasets includes one

Table 1 Statistics of dataset on which experiments are performed [16]

Dataset	Sleep-EDF-20	Sleep-EDF-78
Subject	20	78
EEG Channel	Fpz-Cz	Fpz-Cz
Sampling Frequency	100 Hz	100 Hz
W	8285	65951
N1	2804	21522
N2	17799	69132
N3	5703	13039
REM	7717	25835
Length of EEG Singnal	80	80
Total Samles	42308	195479

EOG channel, one chin EMG channel, and two EEGs channels (Fpz-Cz, Pz-Oz) sampled at 100 Hz. As a follow-up to earlier research, we used information from a sleep cassette study to run various models with Fpz-Cz as the sole input channel.

4.3 SLEEP EDF-20 dataset

Sleep Cassette (SC), which explores the influence of age on sleep, and Sleep Telemetry (ST), which studies the effect of temazepam on sleep, are included in this study on healthy people. For this dataset, each PSG file has two EEG channels (Fpz-Cz,Pz-Oz) which have a sampling frequency of 100. Sleep was assessed on 30-second epochs, and hypnograms were manually rated under the Rechtschaffen and Kales protocol. The epochs are then labeled as N1, N2, N3, N4, Wake, REM, and UNKNOWN class as shown in Fig. 6.

4.4 SLEEP EDF-78 dataset

It contains data from 78 subjects, this dataset is an extended version of the EDF-20 dataset. Both datasets contain the same characteristics. This dataset contains around 2 lakh samples, of which the majority are of N2 class because our sleep cycle mostly consists of the N2 state. This creates a data imbalance problem which is solved in this model.

The following is an example of a common and basic preprocessing technique that we utilize for the datasets:

1. Since M and UNKNOWN stages don't fit into any sleep stages, we exclude both.
2. In accordance with AASM standards, we combine stages N3 and N4 into one stage N3.

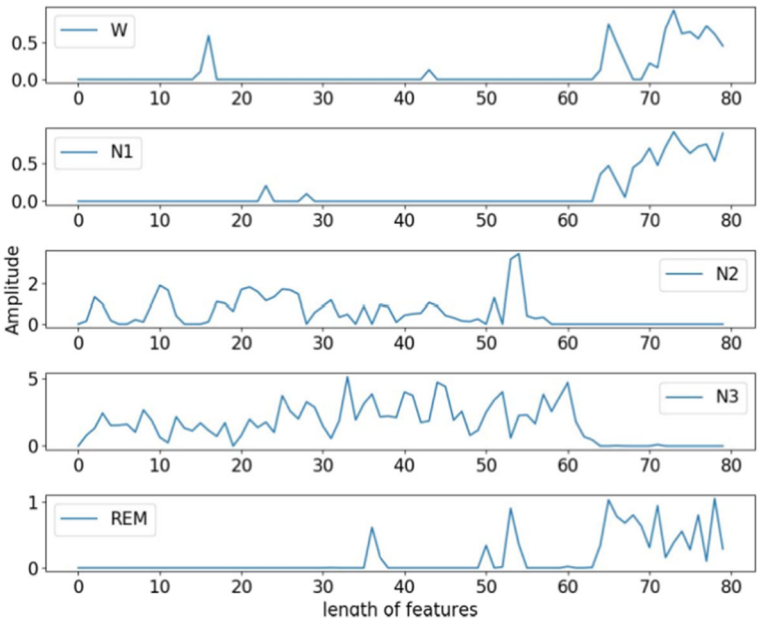

Fig. 6 Five different classes with unique amplitude for the extracted features [26]

3. To emphasize the sleep stages, We have included only half-hour of wake intervals before and after sleep periods.

4.5 Evaluation metrics

We employed these four measures to evaluate the efficacy of various sleep stage classification models: precision, accuracy (ACC), F1-score, and Recall.

4.5.1 Precision

Precision is a metric that measures the proportion of times a class is properly predicted relative to the total number of samples in that class. Precision is determined by using (6).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{6}$$

4.5.2 Recall

The recall score is the proportion of times a class was properly predicted out of the total of times it was predicted. The formula for determining recall is given in (7).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negatives} \tag{7}$$

4.5.3 F1 score

The F1 score is often calculated by taking the average of the Recall and Precision scores, although these scores are also weighted. The F1 score is computed with the help of (8).

$$F1\ score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{8}$$

4.5.4 Accuracy

The classifier's accuracy is measured by how often it makes accurate predictions. It is calculated by taking the number of correct guesses and dividing it by the total number of predictions. The accuracy can be determined by using (9).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

4.6 Classification of sleep stages

A classification report is generated after the model is trained completely. Class-wise metrics are given including the precision, F1 score, and recall. Support is the total number of samples of the class in the training set. The complete classification report with per-class metrics is shown in Table 2.

A confusion matrix is provided for a more comprehensive understanding of the categorization. It indicates the percentage of occasions when the expected class was correctly predicted. The categories are displayed in rows and columns. Our model's confusion matrix for identifying sleep stages is depicted as a heat map in Fig. 7.

Table 2 Classification report of Efficient AttnSleep model (proposed model)

Sleep classes	Precision	Recall	F1-score	Support
W	0.95890	0.92875	0.94358	2035
N1	0.59537	0.51841	0.50806	402
N2	0.89292	0.89632	0.88446	3395
N3	0.91292	0.90153	0.90719	1442
REM	0.80979	0.79353	0.80158	1792
accuracy			0.85848	9066
Macro avg	0.77058	0.76771	0.76898	9066
Weighted avg	0.86063	0.85848	0.85941	9066

4.7 Training and validation accuracy

This section compares the proposed ESAM's training and testing accuracy and loss with the previous Attnsleep model [4]. Figure 8 shows the training validation accuracy of the Attenslepp model there is a large deviation approximately 6% deviation in training and validation accuracy due to overfitting of the model. To resolve the overfitting problem, the ridge regularization method is used in the proposed EASM. Therefore, the proposed model has a smaller deviation of 0.7% in training and validation accuracy, as shown in Fig. 9.

4.8 Training and validation loss

Figure 10 shows the training and validation loss of the Attnsleep model. It is observed from Fig. 10, the attention sleep model has a larger loss between training and validation due to the presence of data imbalance problems. A class-balanced focal loss function is used in the proposed ESAM to solve the data imbalance problem. It can be observed from Fig. 11 that loss between training and validation is reduced when compared to the previous attention sleep model.

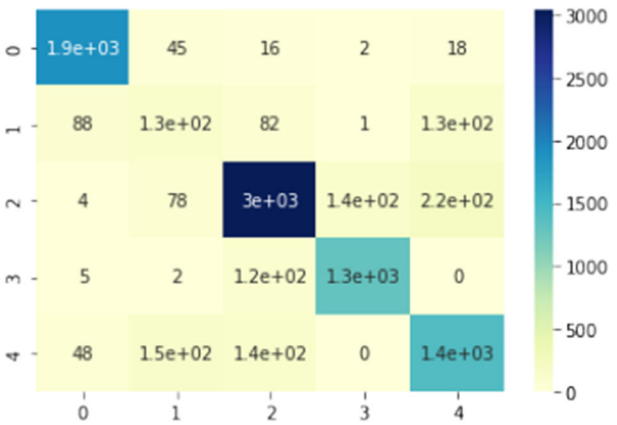

Fig. 7 Confusion matrix to get the overall idea of classification of sleep classes

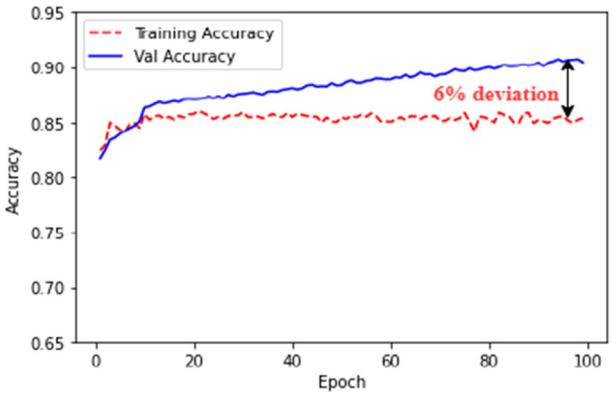

Fig. 8 Training and validation accuracy versus the number of epochs of AttnSleep model [4]

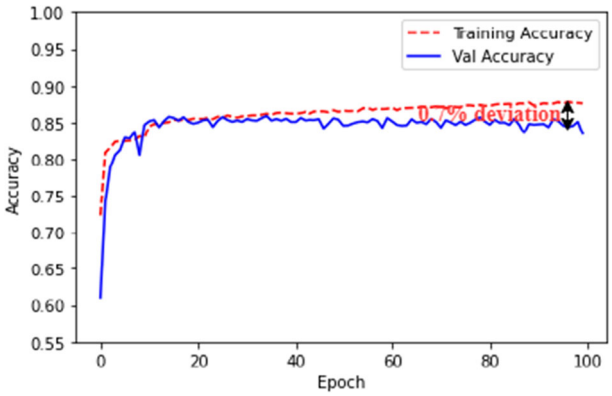

Fig. 9 Training and validation accuracy versus the number of epochs of proposed Efficient AttnSleep model

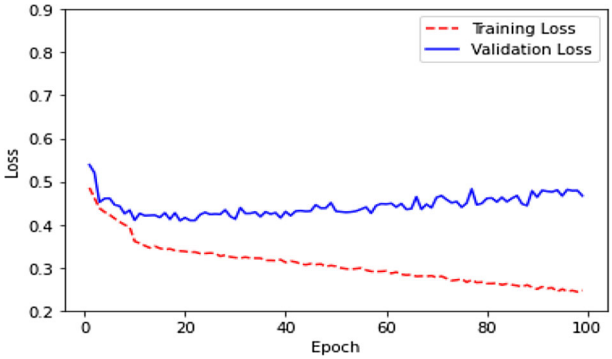

Fig. 10 Training and testing loss versus a number of epochs of the AttnSleep model. [4]

Table 3 Comparison of Efficient AttnSleep model with other State-of-the-art models

Dataset	Network	W	N1	N2	N3	REM	Accuracy	MF1	Trainable Parameters	Training time
	DeepSleepNet [6]	86.7	45.5	85.1	83.3	82.6	81.9	76.6	15,72,325	2.5 hrs
	MultitaskCNN [25]	87.9	33.5	87.5	85.8	80.3	83.1	75.1	11,35,780	2.6 hrs
	AttnSleep [4]	89.7	42.6	88.8	90.2	79	84.4	78.1	4,54,005	21 mins
Sleep-EDF-20	Efficient AttnSleep	95.89	59.5	89.29	91.27	80.9	**85.8**	**80.8**	**2,00,693**	**11 mins**
	DeepSleepNet [6]	90.9	45.0	79.2	72.7	71.1	77.8	71.8	15,72,325	7.2 hrs
	MultitaskCNN [25]	90.9	39.7	83.2	76.6	73.5	79.6	72.8	11,35,780	5.3 hrs
	AttnSleep [4]	92.0	42.0	85.0	82.1	74.2	81.3	75.1	4,54,005	1.7 hrs
Sleep-EDF-78	Efficient AttnSleep	93.82	57.1	87.15	90.28	81.6	**81.7**	**78.1**	**2,00,693**	**1.1 hrs**

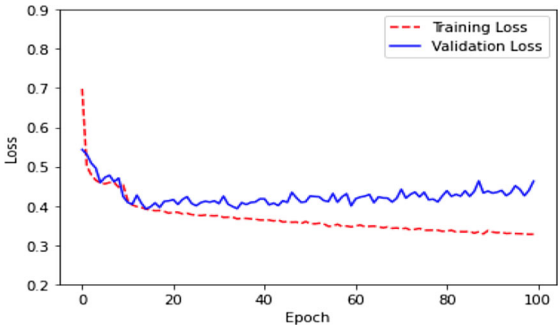

Fig. 11 Training and testing loss versus the number of epochs of Efficient AttnSleep model

4.9 Comparison with other models

The performance of the Efficient AttnSleep model is evaluated against various state-of-the-art models. The overall accuracy, F1-score, training parameters, and average training time matrices are used to compare the performance of different models on two datasets (i.e. EDF-20 and EDF-78). Table 3 shows the comparison of the proposed EASM with DeepSleepNet [6], MultitaskCNN [25], and AttnSleep model [4]. Due to its improved feature extraction module, single TCE with attention mechanism, and advanced focus loss function with ride regulation mechanisms, we find that the Efficient AttnSleep Model (EASM) provides superior classification performance than the other models. This is something that we have observed. In particular, EASM achieves better accuracy on Sleep-EDF20 and Sleep-EDF78 datasets.

The complexity of the proposed model is reduced by reducing the number of convolutional layers in the feature extraction module (i.e., MRCNN) and single TCE block. Therefore, the proposed model's training parameter and training time are much less than other methods as shown in Table 3. The accuracy of the proposed model is improved by 1.7% and runtime is reduced to 60% when compared to the previous Attnsleep model [4].

5 Ablation study

In order to assess the efficacy of individual components within our EASM, we conducted an ablation study using the Sleep-EDF-20 dataset. We formulated three distinct model variants for this purpose. Notably, the initial two variants were designed without the integration of the class-aware loss function. From the ablation study presented in Table 4, we infer three key findings. Initially, the Adaptive Feature Refinement (AFR) module boosts classification efficacy, underscoring the importance of modeling feature interdependencies. Secondly, when contrasting MRCNN with MRCNN+TCE, it's evident that incorporating Temporal Contextual Encoding (TCE) significantly enhances the model's ability to classify sleep stages by capturing temporal dependencies. Lastly, AttnSleep markedly outperforms the other four variants in terms of MF1 and MGm scores. This indicates that our proposed class-aware cost-sensitive loss function effectively mitigates data imbalance issues without increasing computational demands.

Table 4 Ablation study on Sleep-EDF-20 dataset

	Accuracy	Macro F1 Score	Macro G-mean	Kappa
MRCNN	83	75.2	83.2	76.6
MRCNN + TCE	83.7	76.6	84.3	77.8
EASM	85.8	80.8	85.7	0.80

5.1 Analysis of MHA

Given the pivotal role of Multi-Head Attention (MHA) in our model, examining its impact on performance is crucial, specifically focusing on the variation in the number of heads while keeping other parameters constant. For our experimentation, we varied the number of heads in MHA, ensuring compatibility with the feature-length d, which is 80 for the Sleep-EDF-20 dataset. Table 5 shows the model performance on the Sleep-EDF-20 dataset in terms of accuracy and MF1 score. Overall, the model performance is quite stable when we use different numbers of heads. Accordingly, we tested the model with 1, 2, 4, 5, 8, and 10 heads. The performance outcomes reflected in accuracy and the MF1 score on the Sleep-EDF-20 dataset, are detailed in Table 5. Results indicate a relatively stable performance across different head counts. Incremental improvements were observed as the number of heads increased from 1 to 5, attributed to enhanced feature and interaction detections. However, further increasing the heads to 8 and 10 resulted in diminishing returns due to the reduced feature length per head, slightly affecting the performance. Ultimately, we established the optimal number of heads as 5 for this dataset in our tests.

6 Conclusion

A novel self-attention model for the classification of the sleep stage is proposed. The model accomplishes the task of feature extraction using the MRCNN and AFR models. We have used only two CNN layers per branch to reduce complexity and extract higher-frequency features. The TCE module maps the temporal dependencies using Multi-head attention. To solve the data imbalance problem, we have introduced a class-balanced focal loss which calculates an effective number of samples per class. At last, the model's over-fitting problem is solved using ridge regularization. The model has outperformed other models with a classification accuracy of 85.5% and a reduction in training time to just 11 minutes.

Data Availability The Data used in this work is available in the public domain.

Table 5 The performance of EASM on Sleep-EDF-20 dataset with different number of heads in MHA

Number of heads	Accuracy	F1- Score
2	84	77.8
4	85.0	78
6	85.8	80
8	84	77.8
10	84.2	78.2

Declarations

Conflicts of Interest The authors declare that they have no known conflict of interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Luyster FS, Strollo PJ Jr, Zee PC, Walsh JK (2012) Sleep: a health imperative. Sleep 35(6):727–734
2. Zhao X, Wang X, Yang T, Ji S, Wang H, Wang J, Wang Y, Wu Q (2021) Classification of sleep apnea based on eeg sub-band signal characteristics. Sci Rep 11(1):5824
3. Zywietz C, Von Einem V, Widiger B, Joseph G (2004) Ecg analysis for sleep apnea detection. Methods Inf Med 43(01):56–59
4. Eldele E, Chen Z, Liu C, Wu M, Kwoh C-K, Li X, Guan C (2021) An attention-based deep learning approach for sleep stage classification with single-channel eeg. IEEE Trans Neural Syst Rehabil Eng 29:809–818. https://doi.org/10.1109/TNSRE.2021.3076234
5. Rauchs G, Desgranges B, Foret J, Eustache F (2005) The relationships between memory systems and sleep stages. J Sleep Res 14(2):123–140
6. Supratak A, Guo Y (2020) Tinysleepnet: an efficient deep learning model for sleep stage scoring based on raw single-channel eeg. In: 2020 42nd Annual international conference of the IEEE engineering in medicine & biology society (EMBC), pp 641–644. https://doi.org/10.1109/EMBC44109.2020.9176741
7. Seo H, Back S, Lee S, Park D, Kim T, Lee K (2020) Intra- and inter-epoch temporal context network (iitnet) using sub-epoch features for automatic sleep scoring on raw single-channel eeg. Biomed Signal Process Control 61:102037. https://doi.org/10.1016/j.bspc.2020.102037
8. Pant H, Dhanda HK, Taran S (2022) Sleep apnea detection using electrocardiogram signal input to fawt and optimize ensemble classifier. Measurement 189:110485. https://doi.org/10.1016/j.measurement.2021.110485
9. Zhu G, Li Y, Wen P (2014) Analysis and classification of sleep stages based on difference visibility graphs from a single-channel eeg signal. IEEE J Biomed Health Inform 18(6):1813–1821
10. Memar P, Faradji F (2017) A novel multi-class eeg-based sleep stage classification system. IEEE Trans Neural Syst Rehabil Eng 26(1):84–95
11. Dimitriadis SI, Salis C, Linden D (2018) A novel, fast and efficient single-sensor automatic sleep-stage classification based on complementary cross-frequency coupling estimates. Clin Neurophysiol 129(4):815–828
12. Dash S, Acharya BR, Mittal M, Abraham A, Kelemen A (2020) Deep learning techniques for biomedical and health informatics. Springer, ???
13. Mohanty C, Mahapatra S, Acharya B, Kokkoras F, Gerogiannis VC, Karamitsos I, Kanavos A (2023) Using deep learning architectures for detection and classification of diabetic retinopathy. Sensors 23(12):5726
14. Tsinalis O, Matthews PM, Guo Y, Zafeiriou S (2016) Automatic sleep stage scoring with single-channel eeg using convolutional neural networks. arXiv:1610.01683
15. Sokolovsky M, Guerrero F, Paisarnsrisomsuk S, Ruiz C, Alvarez SA (2019) Deep learning for automated feature discovery and classification of sleep stages. IEEE/ACM Trans Comput Biol Bioinf 17(6):1835–1845
16. Supratak A, Dong H, Wu C, Guo Y (2017) Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg. IEEE Trans Neural Syst Rehabil Eng 25(11):1998–2008. https://doi.org/10.1109/TNSRE.2017.2721116
17. Phan H, Andreotti F, Cooray N, Chén OY, De Vos M (2018) Joint classification and prediction cnn framework for automatic sleep stage classification. IEEE Trans Biomed Eng 66(5):1285–1296
18. Li F, Yan R, Mahini R, Wei L, Wang Z, Mathiak K, Liu R, Cong F (2021) End-to-end sleep staging using convolutional neural network in raw single-channel eeg. Biomed Signal Process Control 63:102203
19. Tsinalis O, Matthews PM, Guo Y, Zafeiriou S (2016) Automatic sleep stage scoring with single-channel eeg using convolutional neural networks. arXiv:1610.01683
20. Chen Z, Wu M, Cui W, Liu C, Li X (2020) An attention based cnn-lstm approach for sleep-wake detection with heterogeneous sensors. IEEE J Biomed Health Inform 25(9):3270–3277
21. Zhu T, Luo W, Yu F (2020) Convolution-and attention-based neural network for automated sleep stage classification. Int J Environ Res Public Health 17(11):4152
22. Memar P, Faradji F (2018) A novel multi-class eeg-based sleep stage classification system. IEEE Trans Neural Syst Rehabil Eng 26(1):84–95. https://doi.org/10.1109/TNSRE.2017.2776149

23. Cheng G, Si Y, Hong H, Yao X, Guo L (2020) Cross-scale feature fusion for object detection in optical remote sensing images. IEEE Geosci Remote Sens Lett 18(3):431–435
24. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30
25. Sors A, Bonnet S, Mirek S, Vercueil L, Payen J-F (2018) A convolutional neural network for sleep stage scoring from raw single-channel eeg. Biomed Signal Process Control 42:107–114
26. Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE (2000) Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. Circulation 101(23):215–220