NAMRTNet: Automatic Classification of Sleep Stages Based on Improved ResNet-TCN Network and Attention Mechanism

Xuebin Xu 1,2, Chen Chen 1,2,*, Kan Meng 1,2, Longbin Lu 1,2, Xiaorui Cheng 1,2 and Haichao Fan 1,2

1 School of Computer Science and Technology, Xi’an University of Posts & Telecommunications, Xi’an 710121, China; ccpp9999@126.com (X.X.); mengkan@stu.xupt.edu.cn (K.M.); lulongbin@xupt.edu.cn (L.L.); cxxxxxr@stu.xupt.edu.cn (X.C.); fhc7911@stu.xupt.edu.cn (H.F.)
2 Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi’an University of Posts & Telecommunications, Xi’an 710121, China

* Correspondence: 2103210107@stu.xupt.edu.cn

Abstract: Sleep, as the basis for regular body functioning, can affect human health. Poor sleep conditions can lead to various physical ailments, such as poor immunity, memory loss, slow cognitive development, and cardiovascular diseases. Along the increasing stress in society comes with a growing surge in conditions associated with sleep disorders. Studies have shown that sleep stages are essential for the body’s memory, immune system, and brain functioning. Therefore, automatic sleep stage classification is of great medical practice importance as a basis for monitoring sleep conditions. Although previous research into the classification of sleep stages has been promising, several challenges remain to be addressed: (1) The EEG signal is a non-smooth signal with harrowing feature extraction and high requirements for model accuracy. (2) Some existing network models suffer from overfitting and gradient descent. (3) Correlation between long time sequences is challenging to capture. This paper proposes NAMRTNet, a deep model architecture based on the original single-channel EEG signal to address these challenges. The model uses a modified ResNet network to extract features from sub-epochs of individual epochs, a lightweight attention mechanism normalization-based attention module (NAM) to suppress insignificant features, and a temporal convolutional network (TCN) network to capture dependencies between features of long time series. The recognition rate of 20-fold cross-validation with the NAMRTNet model for Fpz-cz channel data in the public sleep dataset Sleep-EDF was 86.2%. The experimental results demonstrate the network’s superiority in this paper, surpassing some state-of-the-art techniques in different evaluation metrics. Furthermore, the total time to train the network was 5.1 h, which was much less than the training time of other models.

Keywords: sleep stage classification; sleep quality assessment; single-channel EEG signals; deep learning; long time series

1. Introduction

Sleep is an inherently complex physiological process in humans, and sleep disorder problems can seriously endanger human health, such as memory loss, mental discomfort, and the induction of cardiovascular diseases [1]. Everyone goes through a periodic cycle of sleep stages during sleep, and those with sleep disorders will develop problems with inconspicuous or confusing processes. In 2020, the COVID-19 pandemic led to a large regional sequestration and an increasing number of people with sleep disorders, especially women [2]. Therefore, accurate classification of sleep stages is critical in diagnosing and treating sleep disorders.

Typically, polysomnography (PSG), also known as a sleep study, is a clinically approved sleep monitoring device that is used to assess sleep quality. The data recorded by PSG includes various physiological signals such as electroencephalography (EEG), electromyography (EMG), electrocardiography (ECG), and other environmental signals [3]. Clinically, sleep specialists classify sleep into five stages according to the sleep staging...
criteria of the AASM (American Academy of Sleep Medicine): W, N1, N2, and N3, REM. The need for manual classification and statistical analysis of sleep stages based on the characteristics of brain waves during different sleep periods and personal experience makes manual sleep staging a tedious and time-consuming task, and the staging results are highly subjective [4]. With the increasing number of people suffering from sleep disorders and the growing impact of other sleep problems, many researchers have developed automatic sleep detection classification based on machine learning and deep learning [5–27].

Based on traditional machine learning, such as decision trees [5], random forests [6,7], and support vector machines [8–10], the construction and selection of features for these methods are critical. Standard features can be broadly classified into time domain, frequency domain, time–frequency, and nonlinear features. Considering that all EEG sleep information is contained in the time domain waveform, the first approach is to extract valuable features directly from the time domain. Lubin used spectral analysis to classify EEG signals into five bands: α, θ, δ, β, and σ, and found that δ and σ waves can distinguish sleep states well [11]. After long-term research, the widely used time-frequency analysis methods include the short time Fourier transform [12], wavelet transform [13], and Hilbert–Yellow transform [14]. The nonlinear features are mainly based on entropy and complexity. Although the traditional machine learning-based sleep detection grading method has achieved some results, the feature data extracted by this method is not comprehensive enough, and the accuracy of sleep scoring is not high enough.

With the gradual advancement of research, neural network-based deep learning techniques have attracted a lot of attention from researchers. Compared with traditional machine learning, deep learning can automatically learn multidimensional abstract features of data from different perspectives by overlaying network layers and powerful neurons. Several studies have designed convolutional neural networks (CNNs) [15–21]. On study [15] developed a high-precision real-time automatic k-complex wave detection system using multiple CNN networks with migration learning to implement a convolutional neural network (faster R-CNN) detector with faster regions. Another study [16] used end-to-end training with two pairs of convolutional layers for filtering, pooling layers for subsampling, a backpropagation algorithm for iterative optimization training, and a class balancing processor for batch sampling to solve typical data imbalance problems. There are 14 layers of networks, with the 30 s epoch as the input, the first two epochs and the last one as the temporal background, and the original signal as a sample, without the need for domain-specific feature extraction methods [17]. On the other hand, [18] proposed a CNN network architecture based on network depth to improve the classification performance of CNNs, which achieved 81% classification accuracy. Ref. [19] using 90S EEG signal as an epoch as input, the network architecture of superimposed micro-neural network and compression-excitation (SE) block was proposed, which achieved 85.3% accuracy on the Sleep-EDF data set. Furthermore, [20] used a two-dimensional convolutional network to classify raw data from three channels (i.e., EEG, ECG, and EMG). Finally, [21] proposed a fast convolution method that uses 1D convolution and a layered Softmax method in the classification layer to improve computational efficiency. CNN models have achieved some results in automatic sleep classification, but it is clear that hand-designed features perform better than the original signal. It may be since when sleep experts score the sleep stages of a period, they usually find sleep-related events (e.g., frequency components: α, β, δ, and θ, k-complex, sleep spindle) in that period. The relationship between sleep stages of adjacent periods is then analyzed. The CNN model does not consider the temporal information the sleep expert uses in determining the sleep stages in each period. In addition, the CNN network has certain limitations that lead to a less-than-optimal model, so the application of an improved network is warranted.

Some researchers have started to apply RNNs (recurrent neural networks) to sleep staging [22–27]. RNN is able to maintain internal memory, condition historical information, and learn temporal information from input sequences using feedback connections. Their main strength is that they can be trained to understand patterns of variation between
EEG signals and to obtain a temporal correlation of the data, for example, to identify the next possible sleep stage in a series of ephemeral information transformation rules [22]. Elman [23] used RNNs to provide both feedforward and feedback connection channels, facilitating the capture of relevant information before and after sleep. Another study [24] used RNNs to obtain information about the time interval at which CNNs acquire features in each epoch, using a sleep staging model based on a single-channel EEG signal, using CNNs with different convolutional kernels to extract features from the original signal separately and then feeding them into a long short-term memory (LSTM) for learning, on the publicly available dataset Sleep-EDF obtained an overall accuracy of 82.0%. Phan et al. [25] used RNN to analyze representative features extracted from sub-epochs within the temporal context within and between epochs, achieving an accuracy of 85.3% on the Sleep-EDF dataset. Furthermore, [26] used a combination of CNN and LSTM to extract the time–frequency features and learning time information of EEG signals and train a classifier for sleep staging. Finally, [27] proposed a sleep learning segmentation algorithm based on low sampling frequency pressure sensing signals. Convolutional neural networks were used to extract sleep segmentation features, and LSTM neural networks were used to extract time series features. The results show that considering both intra-epoch and inter-epoch temporal context is beneficial to improve automatic sleep scoring. However, LSTM has many structural parameters, complex computations, and long model training time, which affects the training efficiency of the model [28].

This paper proposes NAMRTNet, an automatic sleep stage staging model based on an improved hybrid network ResNet-TCN and NAM attention mechanism. NAMRTNet analyzes the temporal context mainly at the level of epochs and sub-epochs and uses an enhanced ResNet network combined with the NAM attention mechanism to encode each epoch of EEG signal sub-epochs into corresponding representative features, and suppress the insignificant features and ignore the noise effects. The temporal features are then obtained using TCN networks to learn the transition rules between phases and solve the dependence problem of long-time sequences. NAMRTNet is an end-to-end deep learning-based model that uses the original single-channel Fpz-cz signal without any data pre-processing operation. Extensive experiments on the dataset Sleep-EDF show that the NAMRTNet model outperforms some state-of-the-art techniques for automatic sleep stage classification. The network in this paper achieves an accuracy of 86.2%, and due to the TCN network’s parallelism, the NAMRTNet model’s training time is 5.1 h, which is significantly less than other models, effectively saving resources.

The remainder of the paper is organized as follows: Section 2 presents the basic framework of the model and some improvements to this paper. Section 3 describes the dataset and evaluation metrics, the comparison experiments, the network parameter settings, the sleep stage score performance, and the comparison results against state-of-the-art methods. Finally, in Section 4, the work of this paper is summarized.

2. Materials and Methods

2.1. Dataset

In this paper, we chose the publicly available dataset Sleep-EDF, which includes PSG records labeled by human sleep experts and the corresponding sleep stages. The Sleep-EDF dataset comprises two sets of subjects, one of the healthy subjects SC without sleep-related disorders and the second of subjects ST with temazepam effects on sleep. As the Fpz-cz channel is more suitable than the Pz-oz channel data for sleep stage staging, this paper uses the EEG signal from the Fpz-cz channel at a sampling rate of 100 hz without any pre-processing operations. All recordings within the dataset will be divided into multiple 30 s segments, each segment symbolizing a stage, and the sample distribution of the Sleep-EDF dataset is shown in Table 1.
Table 1. Number of epochs by class in the dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>W</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>REM</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep-EDF</td>
<td>8285</td>
<td>2804</td>
<td>17,799</td>
<td>5703</td>
<td>7717</td>
<td>42,308</td>
</tr>
</tbody>
</table>

2.2. Model Overview

This section introduces the NAMRTNet network model of this paper. Figure 1 shows the overall architecture of the NAMRTNet network model.

Figure 1. The network structure of NAMRTNet. The green box plots indicate representative features extracted from each sub-epoch (purple boxes), aggregating into a sequence of features for a single epoch (green boxes). We used a sequence of four epochs to explore temporal dependence (orange boxes), (* represents multiplication sign).

Sleep experts performing sleep stage staging will observe the frequency characteristics of the EEG signal and the transition relationships between these sleep-related events, such as the k-complex and sleep spindle waves. In this paper, we retrieve sub-epoch (as shown in the purple box in Figure 1) features from the 30 s (one epoch) EEG signal with a modified ResNet network to learn sleep-related events, and the attention module suppresses the unimportant features. Then, temporal features are obtained at the ephemeral and sub-ephemeral levels with a TCN network to learn transition rules between sleep stages, and finally, sleep stages are classified [29].

Precisely, to extract more productive features, this paper uses a method of dividing the 30 s EEG sequence into k-segment series (each sequence is a sub-epoch), with representative features of each segment extracted by a modified ResNet and NAM network. In addition, to capture the changes experienced by the transitions between each stage, each segment
sequence contains some overlapping features of the previous segment. The orange box in Figure 1 shows the features extracted by the ResNet and attention networks. The extracted ones are then sent in left-to-right order to the TCN network for final classification. The TCN network can learn the dependencies between features of long-time sequences, and in order to analyze the temporal contextual relationships, four epochs are input in this paper, which can study the transition rules between sleep stages at both epoch and sub-epoch levels, capturing the temporal correlation between sequences the temporal correlation. Finally, the activation function softmax completes the classification output for the sleep phase. Next, the paper describes the individual modules in detail.

2.3. Multi-Sub-Epoch Feature Learning

Figure 2 displays the network structure of feature extraction at the sub-epoch level in this paper, consisting of an improved ResNet 34 and attention mechanism (MRNANet). MRNANet extracts the features of each sub-epoch, the NAM attentional mechanism is used to reduce the weight of less significant features, and then establish a feature sequence for each epoch.

\[
\{f_{1,1}, f_{1,2}, \ldots, f_{1,h}\}
\]

Figure 2. Shows our feature extraction module from the sub-epoch, consisting of an improved ResNet network and attention mechanism (* represents multiplication sign).
The output feature sequence of 4 epochs can be expressed as:

\[ F = \{ f_{1,1}, f_{1,2}, \ldots, f_{1,j}, \ldots, f_{4,1}, \ldots, f_{4,j} \} \]  

\( f_{i,j} \) is the representative feature of the \( j \) sub-epoch and of the \( i \) epoch. The last layer, the pooling layer, can filter out useful features from many features to prevent over-fitting problems in classification tasks. MRNANet will be introduced in detail below.

2.4. Improved ResNet 34

ResNet 34 for this article includes a start-up phase, four major phases, and a dropout layer. Each central stage consists of several ResBlocks.

Figure 3a, the original ResBlock bottleneck, contains two convolution layers (one convolution layer kernel is \( 1 \times 1 \) and another is \( 3 \times 3 \)), two batch normalizations, and two ReLUs. The gray arrow in the original ResBlock represents information transmission, and the ReLU activation function exists on its path. ReLU will return to zero when it meets a negative signal, which may cause inaccurate information transmission.

![Residual component architecture](image)

**Figure 3.** Residual component architecture: (a) original [30]; (b) proposed ResStage. (The first BN in the first intermediate ResBlock is eliminated at each stage, * represents multiplication sign).

Each of the four major phases of the improved ResNet 34 consists of a start ResBlock, an end ResBlock, and an intermediate ResBlock. Figure 3b demonstrates three differences between these three residual blocks. First, we changed the kernel of all convolution layers to \( 1 \times 1 \), so one-dimensional processing is more suitable for electrical signals. Secondly, the distribution is different. In the start ResBlock, after the last convolution, there is a batch normalization (BN) layer ready to add elements using the projection shortcut. The end ResBlock ends with the BN layer and Gaussian error linear unit activation function (GeLU). The end ResBlock is the preparation for the next phase of execution, which allows information to flow more smoothly across the network. Finally, the GeLU function replaces each residual block’s original linear rectifier function (ReLU). The form of the ReLU function is:

\[ \text{ReLU}(x) = \max(0, x) \]  

(3)
which determines that when using the ReLU activation function in calculating the gradient, if many values are lower than 0, there will be many differences. The weights and biases will be updated. The GeLU function has the form:

$$\text{GeLU}(x) = 0.5x \left(1 + \tanh \left( \frac{\sqrt{2\pi}}{x + 0.044715x^3} \right) \right)$$  \hspace{1cm} (4)

It can effectively solve the problem of negative input, so we replace all ReLU functions with GeLU functions.

2.5. NAM Attention Mechanism

An improved channel spatial attention mechanism is added to precede the bare block and dropout layers in the improved ResNet 34 network to suppress less significant features in neural network channels or spaces. NAM, a lightweight attention mechanism used in this paper, adopts the module integration approach of CBAM [31]. It applies a sparse weight penalty to attention modules, which can not only maintain similar performance but also make their calculation efficiency higher.

The channel attention module is shown in Figure 4a, and we apply the scale factor in batch normalization (BN) to the channel dimension.

$$B_{out} = BN(B_{in}) = \gamma \frac{B_{in} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta$$  \hspace{1cm} (5)

\(\mu_B\) is the mean of mini-batch \(B\), and \(\sigma_B\) is the standard deviation of mini-batch \(B\). \(\gamma\) and \(\beta\) are trainable affine transformation parameters (scale and displacement), through the scaling factor measures the variance of the channel [32], the more significant the dissent, said the more meaningful the channel change, and the information contained in the channel would be more abundant, and more critical, those channels with little variation in variance, with single information and little importance, can be ignored.

The formula for the output feature is \(M_c\):

$$M_c = \text{sigmoid}(W_i(BN(F_i)))$$  \hspace{1cm} (6)

The formula for the weight is:

$$W_i = \frac{\gamma_i}{\sum_{j=0}^{\gamma_j}}$$  \hspace{1cm} (7)

As shown in Figure 4b, we apply the scaling factor in batch normalization (BN) to the spatial dimension, outputting features \(M_s\):

$$M_s = \text{sigmoid}(W_s(BN_s(F_s)))$$  \hspace{1cm} (8)

---

**Figure 4.** NAM notes the mechanism module, (a) Channel Attention Module; (b) Pixel Normalization.
The formula for the weight is:

\[ W_j = \frac{\lambda_i}{\sum_{j=0}^{\gamma} \lambda_j} \quad (9) \]

In the two submodules, \( \lambda \) and \( \gamma \), are scaling factors, and the loss function can be expressed as:

\[ \text{Loss} = \sum_{(x,y)} I(f(x,W),y) + p\sum g(\gamma) + p\sum g(\lambda) \quad (10) \]

The regularization terms \( g(\gamma) = |\gamma|g(\lambda) = |\lambda| \) are added to the loss function. While \( p \) is used to balance \( g(\gamma) \) and \( g(\lambda) \), the last two terms are the regularization of the scale factor.

2.6. Long Time Series Dependencies

Due to the limitation of the convolution kernel, the traditional convolutional neural network is unsuitable for modeling timing problems. Recent studies have found that specific convolutional structures can achieve better results [33]. At first, the extended convolution of TCN was used to capture the long time mode. TCN can not only avoid the gradient disappearance and gradient explosion of RNN but also achieve the effect of modeling long time series. The TCN model uses the transformation of a one-dimensional fully convolutional network [34] to predict sequences, passing the sequence information through its multilayer network layer by layer until the prediction result is obtained.

Specifically, the length of each EEG characteristic output by the modified ResNet block is \( T : \{f_1, \ldots, f_T\} \), through the TCN network, which can produce the predictive output of the same length \( Z : \{z_1, \ldots, z_T\} \). The structure of the TCN model is shown in Figure 5a. It can satisfy the \( f : f^{T+1} \rightarrow z^{T+1} \) mapping relationship, and because the causal limit, \( z_T \) only depends on the \( \{f_1, \ldots, f_T\} \), i.e., at time \( t \), the extended convolution’s output depends only on time \( t \) and the inputs of the previous layer, not on any future inputs.

Figure 5. (a) TCN network. (b) Residual block of the TCN network. (c) Residual block \((k,d)\), (* represents multiplication sign).

The formula of extended convolution is defined as follows:

\[ H(K) = \sum_{i=0}^{l-1} f(i) \cdot X_{k-d-i} \quad (11) \]

where \( d \) is the expansion factor, \( j \) is the filter size, and \( k - d \ast i \) represents the past direction. When \( d = 1 \), the dilation convolution becomes a regular convolution, and the input range increases as \( d \) increases [35]; Figure 5c shows the change in the TCN network’s field of view for \( d = 3 \). Thus, each layer of the TCN network can increase the perceptual field of
the TCN by increasing the dilation factor \( d \) or the filter size \( k \) [36], where the compelling history of one such layer is \((j - 1) \times d\).

If we define \( L \) as the number of samples required for each iteration and derive a vector \( z = \{z_i\} \), where \( i = 1, 2, \ldots \) at each time step, the network \( F \) is created by minimizing the predicted value with the proper label. The expected loss is calculated between:

\[
F^* = \min_{f_1, \ldots, f_L, z_1, \ldots, z_T} \text{loss}(f_1, \ldots, f_L, z_1, \ldots, z_T)
\]

At layer \( i \) of the network:

\[
d = O(2^i)
\]

This ensures that there are some filters to capture each input [32], thus achieving a long and compelling history. The residual block shown in Figure 5b is used to accelerate convergence and stability training. Residual connections can train deep networks and transfer information across layers. Each residual block has two convolution and nonlinear mappings that regularize the network using the WeightNorm and Dropout. A 1 \times 1 convolution reduces the dimension and solves the inconsistency of the number of channels in the feature graph. The residual block contains a branch whose output is added to the block’s input \( x \) through a series of transformations:

\[
O = \text{Activation}(x + \phi(x))
\]

This effectively allows layers to learn changes to the identity map rather than the entire transformation, which has repeatedly proven beneficial for deep networks. In addition, the GeLU function is still used as the activation function in TCN residual block.

3. Results and Discussion

3.1. Evaluation Metrics

To fully evaluate the model named NAMRTNet, we used four parameters: accuracy (ACC), Cohen Kappa (\( \kappa \)) [37], F1 score (F1), and MF1:

\[
\text{ACC} = \frac{\sum_{i=1}^{N} e_{ii}}{\sum_{i=1}^{N} \sum_{j=1}^{N} e_{ij}}
\]

\[
\kappa = \frac{\text{ACC} - p_e}{1 - p_e}
\]

\[
p_e = \frac{\sum_{N=1}^{N} a_{N} \times b_{N}}{K \times K}
\]

\[
F1 = \frac{2}{\frac{1}{\text{PR}} + \frac{1}{\text{RE}}}
\]

\[
\text{PR} = \frac{e_{ii}}{\sum_{j=1}^{N} e_{ij}}
\]

\[
\text{RE} = \frac{e_{ii}}{\sum_{j=1}^{N} e_{ji}}
\]

\[
\text{MF1} = \frac{\sum_{N=1}^{N} F1}{N}
\]

where \( e_{ij} \) represents the elements of the \( i \)th row and \( j \)th column, \( N \) is the number of categories, \( K \) is the total number of epochs, \( \text{PR} \) represents the accuracy of distinguishing sleep stages from other stages, \( \text{RE} \) represents the accuracy of predicting sleep stages,
\( a_N \) represents the number of epochs of the Nth category, and \( b_N \) represents the number of epochs predicted to be the Nth category.

### 3.2. Parameters of the Optimizer

Under the condition of no data pre-processing, the Nadam optimizer is used. To avoid overfitting, L2-Norm regularization methods include weight_decay = \( 10^{-6} \), a batch training size of 64, and learning_rate = 0.001. This study did not use any balanced data processing or model training methods. We followed the training for ten consecutive sessions using a cost-tracking approach and stopped training when no improvement was shown. For each fold cross-validation, the best model in the test set was selected for evaluation. This study did not use any method of balancing data processing or model training to achieve an early stop by tracking the validation cost. The training process is implemented in python 3.6.13 and pytorch 1.10.1, on the RTX 3070.

### 3.3. ResNet Layer Number Settings

In order to improve the model performance, this study aimed to select the optimal number of network layers for ResNet. Figure 6 shows the results of experiments with 20-fold cross-validation on the sleep-EDF dataset with different network layers (18, 34, 50, 101, and 152). When the number of network layers is 34, the accuracy, F1 score, and Cohen’s Kappa (\( \kappa \)) are significantly higher than the other network models. Therefore, the number of layers of ResNet is chosen to be 34 in this paper to ensure the depth features that can be extracted effectively without degrading the network performance.

![Figure 6. Comparison results of recognition rate, F1 score, and Cohen's Kappa (\( \kappa \)) under different network layers.](image)

### 3.4. TCN Hidden Layer Setting

It is vital to select the number of channels of TCN hidden layers. Too few channels will lead to a decrease in the learning ability of the network and even a decrease in the ability to predict information. Conversely, too many choices will lead to a more complex network. Such a network not only fails to improve its performance but also tends to fall into local minima during the training process, which will reduce the learning speed of the network. In order to select the most suitable number of hidden layers, the network performance is tested in this paper when the hidden layers are 32 \( \times \) 1, 64 \( \times \) 1, 128 \( \times \) 1, and 64 \( \times \) 1. Figure 7 shows the comparison results of recognition rate, F1 score, and Cohen’s Kappa (\( \kappa \)) with different numbers of channels. It is clear that the 64 \( \times \) 1 network model has better learning ability.

![Figure 7. Comparison results of recognition rate, F1 score, and Cohen's Kappa (\( \kappa \)) under different numbers of channels in TCN hidden layer.](image)
Figure 7. Comparison results of recognition rate, F1 score, and Cohen’s Kappa (κ) under different numbers of channels in TCN hidden layer.

3.5. Comparative Experiments

The NAMRTNet in this paper consists of the modified ResNet-34, the NAM attention mechanism, and the TCN network. Two comparative studies were carried out to analyze the effectiveness of the network modules of NAMRTNet in this paper. The first group was ResNet+TCN (ResNet 50) and ResNet+TCN (modified ResNet 50). As ResNet-TCN does not support a 34-layer network structure, the exact condition of network level 50 was chosen for validation in order to verify the effectiveness of the modules in this paper.

As shown in Figure 8, the accuracy, F1 score, and Cohen’s Kappa (κ) of the improved ResNet network were 84.8, 77.7, and 0.79, respectively, which were higher than the results of the original network (85.8, 79.0, and 0.804). According to the confusion matrix results, N1, N3, and REM are the main stages with a facilitative effect. Thus, the improved network can effectively improve the problem of inaccurate information transfer, and the one-dimensional convolution is more suitable for such long time series of EEG data. Placing Max-pool between the Conv3x and Conv4x layers also reduces the length of the feature sequence by half, reducing the parameters and enhancing the expressiveness of the network.

Figure 8. Confusion matrix of ResNet-TCN and improved ResNet-TCN. (a) ResNet 50-TCN confusion. (b) MResNet 50-TCN confusion.
The second set of experiments was designed to verify the effectiveness of TCN. The comparison results of ResNet+NAM+LSTM (improved ResNet 34) and ResNet+NAM+TCN (improved ResNet 34) are shown in Figure 9. The application of TCN network structure improves the final recognition rate by 2.2%, and the prediction of each stage is enhanced to a certain extent, indicating that TCN can capture the time correlation of long time series well, and the improved network structure can overcome the imbalance of PSG data to a certain extent.

Figure 9. (a) ResNet + NAM + LSTM (improved ResNet 34) confusion. (b) ResNet + NAM + TCN (improved ResNet 34) confusion.

3.6. K-Fold Crossover Experiment

To better represent the network performance, cross-validation experiments (10-fold, 15-fold, and 20-fold) were performed in this paper to find the best classification for the dataset and to avoid the selection of unexpected hyperparameters and models that cannot generalize due to particular classifications. The results in Table 2 show that the evaluation metrics, such as classification accuracy (PR), classification F1 score (F1), and overall accuracy (Acc) of the experiments, conducted with 20-fold cross-validation are better than the other methods, so the dataset partitioning method of randomly dividing the subjects into training, validation, and test sets in the ratio of 15:4:1 maximizes the model performance.

Table 2. The 10, 15, and 20-fold cross-validation results.

<table>
<thead>
<tr>
<th>K-Folds</th>
<th>Overall Performance</th>
<th>Per-Class Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>MF1</td>
</tr>
<tr>
<td>10</td>
<td>83.7</td>
<td>76.4</td>
</tr>
<tr>
<td>15</td>
<td>84.8</td>
<td>77.9</td>
</tr>
<tr>
<td>20</td>
<td>86.2</td>
<td>79.8</td>
</tr>
</tbody>
</table>

3.7. Sleep Stage Scores

As shown in Table 3, the confusion matrix is calculated by summing the scored values of all test data. Each row represents the number of samples classified by the expert, and each column represents the number of epochs predicted by the model. The table also shows the accuracy, recall, and F1 score for each category. According to the confusion matrix, the values in the diagonal position are higher than the other values, which proves that the NAMRTNet model is adequate. In addition, the score of F1 can reach 92.0 in stage W, 89.4 in stage N2, and 88.6 in stage N3. Due to the similarity of the signal characteristics of the N2 and REM periods, it may need to be more accurate in the classification process. The REM stage F1 score is 83.5, and these four stages are above 80. We need to care that the
N1 stage score is relatively low. First of all, from Table 1, it can be seen that the number of N1 stage samples is only 2804, which is far less than other stages. In addition, for N1, as a transition period, there will be many confusing features before and after the transition period, which causes great difficulties for staging.

Table 3. Confusion matrix obtained from 20-pass cross-validation on the Fpz-cz channel of the sleep-EDF dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Predicted</th>
<th>Per-Class Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>N1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1</td>
</tr>
<tr>
<td>W</td>
<td>7497</td>
<td>393</td>
</tr>
<tr>
<td>N1</td>
<td>442</td>
<td>1122</td>
</tr>
<tr>
<td>N2</td>
<td>77</td>
<td>223</td>
</tr>
<tr>
<td>N3</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>REM</td>
<td>76</td>
<td>295</td>
</tr>
</tbody>
</table>

3.8. Comparison with the Forward Approach

Table 4 compares our approach with existing methods. The improved 1-max CNN of [38] is used in combination with a DNN learning frequency domain filter bank to preprocess time–frequency image features. VGG-FE [39] utilizes multi-cone spectroscopy, a typical migration learning approach. SleepEEGNet [40] proposes a CNN-BIRNN network architecture along with a loss function to tackle the issue of class imbalance within EEG datasets. CCRRSleepNet [41] uses hybrid relational induced bias to optimize the network and uses multiple convolutional blocks to extract complex features. AttnSleep [42] utilizes small-core and wide-core dual-branch CNN to extract features of different frequencies and use a multi-head attention (MHA) mechanism to capture time dependencies between features. Furthermore, [43] used migration learning with SeqSleepNet+ and DeepSleepNet+ based on a sequence-to-sequence sleep staging framework to overcome the problem of small datasets and insufficient data in sleep studies. IITNet [44] utilizes ResNet and LSTM to learn the time context within and between epochs of learning. TinySleepNet [45] simplifies the DeepSleepNet model structure and incorporates data augmentation. Overall, our method is relatively novel and achieves better performance because the shared convolutional structure allows the TCN to process long sequences in parallel, so the TCN is more memory efficient than the recurrent network, thus, the training time is shorter, with a total cross-validation time of 5.1 h for 20 folds, which is significantly better than [42] (about 7 h).

Table 4. Comparison of NAMRTNet with other methods.
4. Conclusions

This paper proposes a new network structure, NAMRTNet, to classify raw single-channel EEG data into sleep stages. NAMRTNet consists of two parts. First, a modified ResNet-34 network combined with the NAM attention mechanism extracts ephemeral invariant features to improve the network effectively. Then, the TCN network is used to learn the dependencies between long time sequences. Experimental results on the Sleep-EDF dataset show that the model proposed in this paper is superior to some state-of-the-art methods for a variety of evaluation metrics.

Furthermore, from the experimental procedure, the training time of the model in this paper is significantly outperformed by the literature [42], which is a good indication of the correctness of the choice of the TCN network, which can handle time series efficiently. This paper effectively selects the optimal network experimentally to maximize the model's performance. Finally, NAMRTNet is end-to-end and does not require pre-processing of the data so that it can be directly used for applications in healthcare and other fields. In the future, we intend to enhance our NAMRTNet so that it can be applied to multimodal signals collected by wearable devices containing EEG, EOG, EMG, etc., to fuse multimodal signals, remove noise interference using state-of-the-art techniques, optimize feature extraction, reduce the impact caused by too little data in the N1 stage, and effectively improve the automatic scoring of sleep stages to improve people’s quality of life effectively.

Author Contributions: X.X.: Conceptualization, Investigation, Writing—review and editing. C.C.: Methodology, Writing—original draft. K.M.: Validation, Writing—review and editing. L.L.: Formal analysis, Writing—review and editing. X.C.: Data Curation, Writing—review and editing. H.F.: Data Curation, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61673316), the Department of Education Shaanxi Province, China (Grant 16JK1697), and by the Key Research and Development Program of Shaanxi Province (Grant 2017GY.071).

Data Availability Statement: The [Sleep-EDF] data used to support the findings of this study have been deposited in the [Sleep-EDF Database Expanded] repository.

Acknowledgments: We sincerely thank the National Natural Science Foundation of China and the Department of Education of Shaanxi Province for their financial support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

NAM: Normalization-based Attention Module; TCN: Temporal Convolutional Network; PSG: Polysomnography; EEG: Electroencephalography; EMG: Electromyography; ECG: Electrocardiography; AASM: American Academy of Sleep Medicine; RNNs: Recurrent Neural Networks; R-CNN: Faster Regions with Convolutional Neural Networks; LSTM: Long Short-Term Memory; CNNs: Convolutional Neural Networks; ReLU: Gaussian Error Linear Units; ReLU: Rectified Linear Unit; MRNANet: Improved ResNet 34 and Attention Mechanism; CBAM: Convolutional Block Attention Module; MHA: Multi-Head Attention.

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.