**II. METHODOLOGY**

In the following sections, we present a detailed description of our proposed novel model developed to automatically score each sleep stage from a given EEG signal.

A. Pre-processing

The input to this method is a sequence of 30-s EEG epochs. In order to extract the EEG epochs from a given EEG signal, we follow two simple steps:

1) Segmenting the continuous raw single-channel EEG to a sequence of 30-s epochs and assigning a label to each epoch (i.e., sleep stage) based on the annotation file.

2) Normalizing 30-s EEG epochs such that each one has a zero mean and unit variance.

It is worth mentioning that, these pre-processing steps for the sleep stage extraction are very simple and do not involve any form of filtering or noise removal methods.

B. The architecture

Figure 1 illustrates the proposed network architecture for automatic sleep stage classification. We applied the same bidirectional recurrent neural network and attention decoder architecture provided by [SleepEEGNet: Automated Sleep Stage Scoring with Sequence to Sequence Deep Learning Approach]. Through this paper, we already know that using CNN helps to benefit from the time domain and frequency domain functions in the classification task.

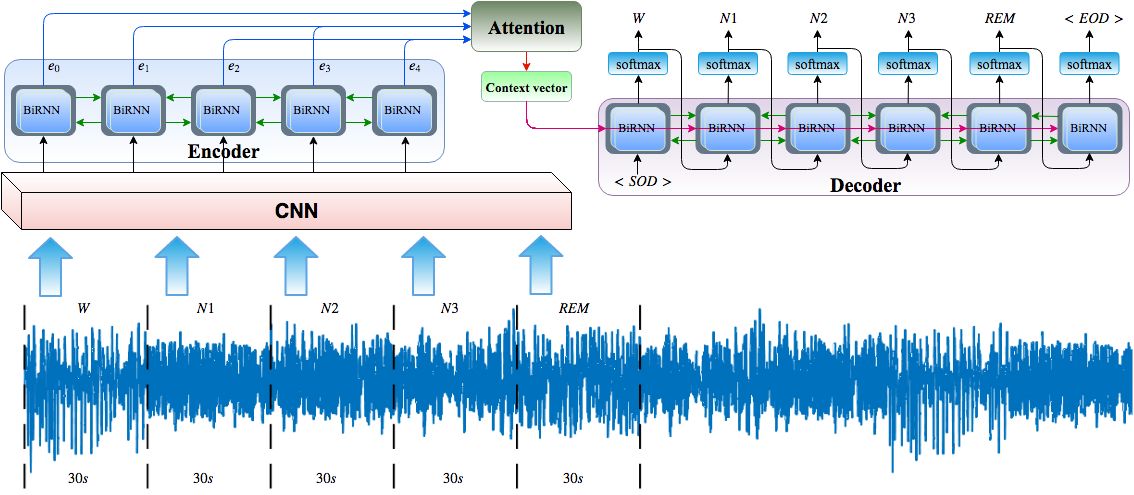


Figure 1: Illustration of sequence to sequence deep learning network architecture used for automated sleep stage scoring.

We hope to deepen the network on this basis, in the hope of getting a higher dimension and more effective features to be sent to BiRNN. Skip connections made the training of very deep networks possible and have become an indispensable component in a variety of neural architectures.

As can be seen in [SKIP CONNECTIONS ELIMINATE SINGULARITIES], several singularities that hinder the depth of the network have been identified in previous work: (i) overlapping singularities caused by the displacement symmetry of the nodes in a given layer, ( Ii) eliminate singularities corresponding to node cancellation (ie, consistent deactivation), (iii) singularities generated by singularities. The linear dependence of the node. These singularities result in the loss of degenerate manifolds in the landscape, which slows down learning. [SKIP CONNECTIONS ELIMINATE SINGULARITIES] considers that skipping joins eliminates the symmetry of the nodes by eliminating the symmetry of the nodes and eliminating them by reducing the likelihood of node elimination and making the nodes independent of linearity. Based on the above, we first made a slight change in the original CNN structure. Each convolutional layer is passed to a rectified linear unit (ReLU) nonlinearity. The first layer is followed by the largest pooling layer and one missing block. Only one missing block is after the last convolutional layer. After the first largest pooling layer, we added skip connection. At each time step of the training/test model, a sequence of 30 seconds EEG period (the size of the maximum time) is fed to the CNN for feature extraction. Finally, the output of the CNN section is connected, followed by a missing block, so that the encoder network encodes the sequence input. Figure 2 depicts the detailed CNN (skipconnect) structure.

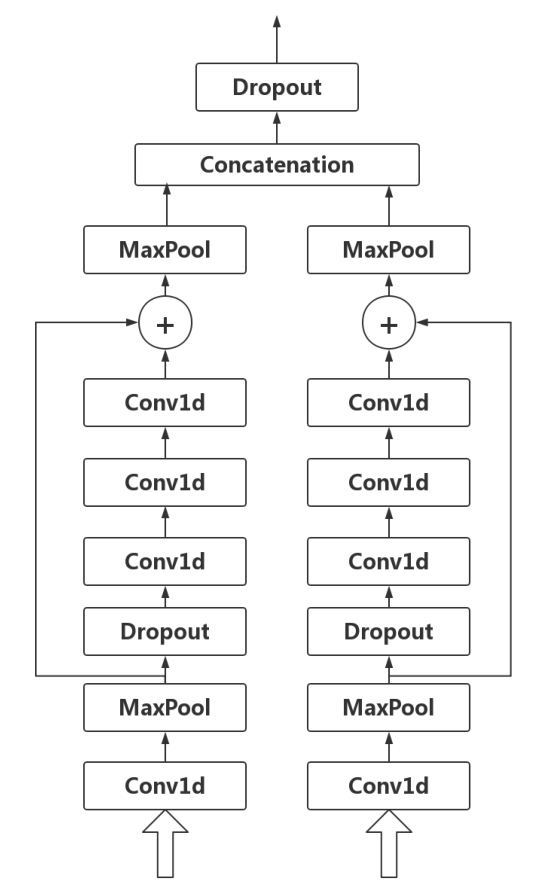


Figure 2: Detailed sketch of the used CNN(skipconnect) model used in the proposed work.

For the second time, we tried a completely different network framework. [FractalNet: Ultra-Deep Neural Networks without Residuals] introduces a neural network macro architecture design strategy based on self-similarity. Repeated application of simple extension rules generates a deep network whose structural layout is a precisely truncated fractal. [FractalNet: Ultra-Deep Neural Networks without Residuals] thinks this network has the ability to transition, during training, from benefits shallow to deep. The second attempt used a convolution to obtain shallow features, then obtained deep features from FractalNet from shallow to deep, and used them for subsequent models. Figure 3 depicts the detailed CNN (conv) structure.

When using EGG for prediction, the previous signal actually has an effect on the subsequent predictions, so the third attempt is based on the second, replacing the first layer of convolution with a divided convolution. [MULTI-SCALE CONTEXT AGGREGATION BY DILATED CONVOLUTIONS] presented a module that uses dilated convolutions to systematically aggregate multiscale contextual information without losing resolution. The architecture is based on the fact that dilated convolutions support exponential expansion of the receptive field without loss of resolution or coverage . Under the same parameters, classified convolution can be associated with longer time features. Figure 3 depicts the detailed CNN (dilatedconv) structure.

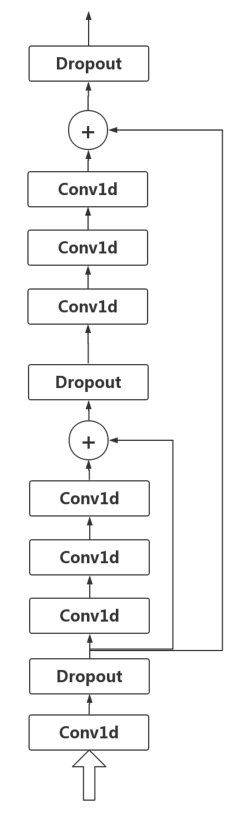
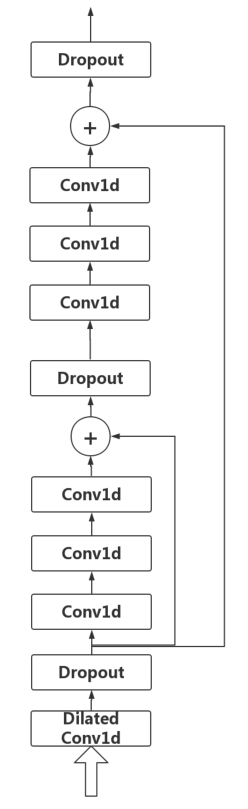
 

Figure 3: Detailed sketch of the used CNN(conv) model and CNN(dilatedconv) used in the proposed work.

**III. EXPERIMENTAL RESULTS**

A. Dataset and Data Preparation

In this study, we used the Physionet Sleep-EDF dataset [B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, and J. J. Oberye, “Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg,” IEEE Transactions on Biomedical Engineering, vol. 47, no. 9, pp. 1185–1194, 2000.] that contributed in 2013 with 61 polysomnograms (PSGs). The Sleep-EDF dataset contains two different studies including (1) study of age effects on sleep in healthy individuals (SC = Sleep Cassette) and (2) study of temazepam effects on sleep (ST = Sleep Telemetry). The dataset includes whole-night polysomnograms (PSGs) sleep recordings at the sampling rate of 100 Hz. Each record contains EEG (from Fpz-Cz and Pz-Oz electrode locations), EOG, chin electromyography (EMG), and event markers. Few records often also contain oro-nasal respiration and rectal body temperature. The hypnograms (sleep stages; 30-s epochs) were manually labeled by well-trained technicians according to the Rechtschaffen and Kales standard [4]. Each stage was considered to belong to a different class (stage). The classes include W, REM, N1, N2, N3, N4, M (movement time) and ’?’ (not scored). According to American Academy of Sleep Medicine (AASM) standard, we integrated the stages of N3 and N4 in one class named N3 and excluded M (movement time) and ? (not scored) stages to have five sleep stages [3]. Table I presents the number of sleep stages.

Table I: Details of number of sleep stages.



B. Experimental Design

The distribution of sleep stages in the Sleep-EDF database is not uniform. Hence, the number of W and N2 stages are much greater than other stages. The machine learning approaches do not perform well with the class imbalance problem. To address this problem, in addition to using the novel loss functions described in Section II-E, the dataset is oversampled to nearly reaching a balanced number of sleep stages in each class. We have used the synthetic minority over-sampling technique (SMOTE) to generates the synthetic data points by considering the similarities between existing minority samples [N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.].

Our proposed model was evaluated using k-fold cross-validation. We set k to 20 and 10 for version 1 and version 2 of the Sleep-EDF dataset, respectively. In other words, we split the dataset into k folds. Then, for each unique fold, (1) fold is taken as test set and the remaining folds as a training set and (2) trained the model using the training set and evaluated the model using the test set. Finally, all evaluation results were combined.

The network was trained (for each dataset) with a maximum of 400 epochs. RMSProp optimizer was used to minimize the l MFE loss with mini batches of size 20 and a learning rate of α = 0.0001 . We also applied an additional L2 regularization element with β = 0.001 to the loss function to mitigate the overfitting. Python programming language and Google Tensorflow deep learning library were utilized to implement our proposed approach.

C. Evaluation Metrics

We have used different metrics to evaluate the performance of the proposed approach including, overall accuracy, precision, recall (sensitivity), specificity, Cohen’s Kappa coefficient ( κ ) and F1-score. We also computed macroaveraging of F1-score (MF1) which is the sum of per-class F1-scores over the number of classes (i.e., sleep stages).

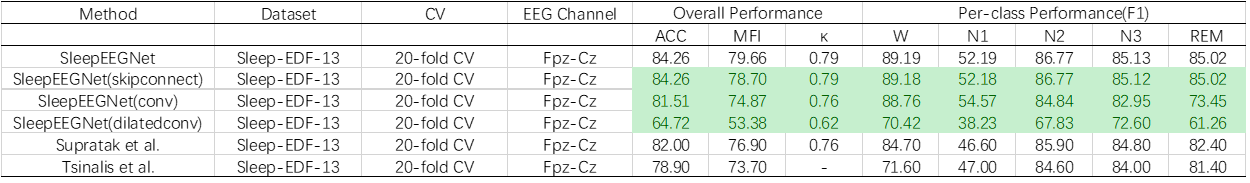
Table II: Confusion matrix and per-class performance achieved by the CNN(dilatedconv) using Fpz-Cz EEG channel of the EDF-Sleep-2013 database.



Table II: Confusion matrix and per-class performance achieved by the CNN(conv) using Fpz-Cz EEG channel of the EDF-Sleep-2013 database.



Table IV: Comparison of performance obtained by our approach with other state-of-the-art algorithms.



IV. CONCLUSION

You can see that we used fewer training epochs and relatively fewer parameters (CNN\_skipconnect trained 120 rounds, each layer convolved with 128 channels of output, CNN\_conv trained 50 rounds, each layer convolved with 128 channels of output, CNN\_dilatedconv trained 30 rounds, each layer convolved with 5 channels of output), and we got fairly good results. In fact, due to limitations in computing resources,and my teammate still have other models to run, time is too late.We only used five channels of output in the dilated convolution. Although this greatly reduces the parameters, it also greatly affects the accuracy. Because other networks, including the original network, the output channels are 128 channels. I personally think that if you increase the number of training rounds or increase the amount of parameters, the model can also achieve a good result..