

AlertNet: Deep convolutional-recurrent neural network model for driving alertness detection

P. C. Nissimagoudar, A.V. Nandi, Aakanksha Patil, Gireesha H. M.

School of Electronics and Communication Engineering, KLE Technological University, Hubballi, Karnataka, India

Article Info

Article history:

Received Jul 10, 2020

Revised Dec 23, 2020

Accepted Jan 19, 2021

Keywords:

Attention network
Bidirectional LSTM
Class imbalance
Electroencephalogram
Loss functions
ResNets
Sequence models

ABSTRACT

Drowsy driving is one of the major problems leading to many road accidents. Electroencephalography (EEG) is one of the most reliable sources to detect sleep on-set while driving as there is the direct involvement of biological signals. The present work focuses on using the deep neural network architecture, built using ResNets and encoder-decoder based sequence to sequence models with attention-decoder. The model is built to reduce the complex computations required for feature extraction. The model learns deeper with minimized loss and training error. The output of ResNets, the features are input to encoder-decoder based sequence to sequence models, built using Bi-directional long-short memories. Sequence to Sequence model learns the complex features of the signal and analyze the output of past and future states simultaneously for classification of drowsy/sleepstage-1 and alert stages. Also, to overcome the unequal distribution (class-imbalance) data problem present in the datasets, the proposed loss functions help in achieving the identical error for both majority and minority classes during the raining of the network for each sleep stage. The model provides an overall-accuracy of 87.92% and 87.05%, a macro-F1-core of 78.06%, and 79.66% and Cohen's-kappa score of 0.78 and 0.79 for the Sleep-EDF 2013 and 2018 data sets respectively.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

P. C. Nissimagoudar
School of Electronics and Communication Engineering
KLE Technological University
Hubballi, Karnataka, India-580031
Email: pcngoudar@gmail.com

1. INTRODUCTION

The physiological signals like EEG, ECG, EMG, and EOG are the proven measures for the analysis and detection of abnormalities in the area of clinical diagnosis, but recently these signals also have found their usage in many other applications. Diver alertness detection is one such application, where these physiological signals can also be used. Amongst the various physiological signals, Electroencephalogram (EEG) which varies in frequency and time-invariant features is found to be a direct indicator of driver's alertness level. The standard 10-20 system is used to collect EEG signals from different locations of the scalp using the electrodes [1]. The different frequency components along with the related amplitude levels observed over the time represent the condition of the brain [2]. There are 5 stages of sleep in which the brain produces distinguishable electric patterns which help in the classification of stages. The PSG signals are collected from a subject during the entire night of sleep and are manually scored by sleep experts into different sleep stages by visually analyzing the signals for a specific time frame [3]. The criteria for sleep stage scoring are proposed in rechtschaffen and kales (RK) [4] manual which was further developed by the

American academy of sleep medicine (AASM) [5]. According to RK manual sleep stages can be classified into the following stages, wake (W) stage, four non-rapid eye movement stage (NREM), and rapid eye movement (REM). The awakening stage is wake stage (W), NREM is the first stage of sleep and the S2 class is when the actual sleep stage begins, S3 is the deep sleep phase continued in the S4 stage, and in REM the eyes are closed with rapid movement. According to the AASM, sleep stage 3 (S3) and sleep stage 4 (S4) are considered as one class represented by SWS (slow-wave sleep) [6, 7]. EEG analysis and decision making include feature extraction [8] using frequency, time, and time-frequency methods, feature selection to retain only significant features, and classification using decision algorithms. Time-frequency transformations or non-linear feature extraction techniques are also being used in the literature for the extraction of relevant features for classification [9]. As the EEG signal has complex behavior, advanced processing, and machine learning algorithms are required, which learn the complexity of the signals and are over threshold based techniques [10, 11].

Recently, the recurrent neural networks (RNN's) have been proven to have better performance than the methods used in "classic" machine learning algorithms on datasets with temporal information (Mikolov *et al.* [12], Graves *et al.* [13], Karpathy and Fei-Fei, [14]). The most commonly used networks are Long short-term memory (LSTM's) neural networks, these networks are successfully applied on raw EEG signals (Davidson *et al.* [15]) as well as to sleep data (Supratak *et al.* [16]). The literature shows both machine learning and deep learning approaches being used for sleep stage scoring, using EEG signals. The standard machine learning approaches use hand-engineered feature extraction and selection methods but work well with a moderately sized data set. The deep learning approaches, like CNN [17] and RNN, learn the features automatically without requiring data preprocessing and feature extraction steps but require large data to train the network.

The paper is organized as follows, after discussing the introduction and the related research in section 1, we discuss the research methods in section 2, which elaborates on the model architecture. Section 3 discusses implementation methods, which describes the details of data sets and their usage, training procedure, and parameter optimization. Section 4, discusses the results and discussion; the results for two different types of data sets, and two different EEG channels. We compare our results with the performance of other related state of art methods in this section. In the last section 5, we conclude on our results.

2. RESEARCH METHOD

This section discusses the detailed method of the proposed model developed for detecting driver's alertness automatically using EEG signals. The model is designed using deep recurrent neural networks and does not require the conventional artifact removing preprocessing steps and feature extraction/selection steps. The model uses residual networks, which are transformed form of CNNs to learn features and encoder-decoder based sequence models using RNN to decide on sleep stage categorization.

2.1. Pre-processing

Preprocessing of EEG signals involves the following three steps,

- The input is taken from Sleep-EDF 2013/2018 data, which is segmented into the 30-s epoch. The raw-EEG signals are used without any prior feature extraction
- The signal is labelled with the help of the annotation file the Hypnogram, which is available with the dataset for each subject.
- 30-s EEG epochs are normalized so that each epoch has mean value as zero and variance as one.

The pre-processing steps do not include any noise/artifact removal techniques or frequency band separation techniques. As we use neural networks there is no need to use noise removal techniques. The sample signal is shown in Figure 1.

2.2. The model architecture

The problem of detecting sleep on-set while driving is a problem of classification of sleep stages with the sequential in nature. Hence, sequence to sequence model proposed in Figure 2 has the following components, i) ResNets with skip connections for extracting features, ii) Bi-LSTMs which process past and future information simultaneously, and iii) Attention decoder, which learns only significant features. We are referring to three classes for classification, i.e. alert/awake state, sleep on-set stage/sleep stage 1, and sleep stage. The sequence to sequence model works on the idea of encoder and decoder technique. For each 30-s EEG epoch, the input sequence is encoded using the encoder part of the model, and the category/class of the input sequence is computed by the decoder part of the model. Both encoders and decoders are built using BI-LSTMs.

The sequence to sequence model works on the idea of encoder and decoder technique. For each 30-s EEG epoch, the input sequence is encoded using the encoder part of the model, and the category/class of the input sequence is computed by the decoder part of the model. Both encoders and decoders are built using BI-

LSTMs. The encoder captures the dependencies related to long-short contexts between the target classes and inputs. The input for the encoders is time-series features obtained from ResNets. The time series non-linear dependencies are captured for detecting the targets by the encoder. The output of the encoded sequence is fed to the attention network and further, they are decoded for detecting the category. Next, we will discuss each module in detail.



Figure 1. Sample EEG Signals

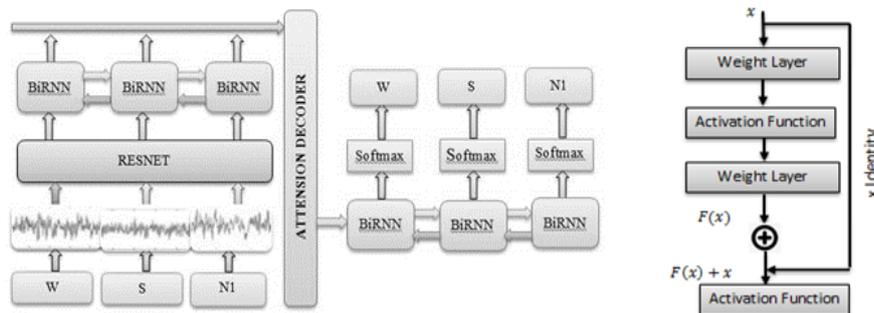


Figure 2. The model architecture with the basic ResNet block

2.3. The ResNets

ResNet a residual neural network is a deep neural network that uses shortcuts to jump some layers, called skip connections [18]. The basic ResNet block is shown in Figure 2. It was developed in the view of avoiding the degradation problem which is encountered in deeper neural networks, it was observed that as the depth of the neural networks increases, accuracy gets saturated and decreases rapidly. The ResNet model is implemented with two or three skips that contain ReLU activation function and BatchNorm in between the skip layers, this will help to avoid vanishing gradient, as the network reuses activation from the previous layer hand till the adjacent layer learns the weights, only the weights of adjacent layers are considered, this provides best results when a nonlinear layer is skipped or the consecutive layers are linear. The skip connections use only fewer layers in the starting training stages, which simplifies the layer. The learning is thus faster reducing the vanishing gradients' impact. On the later part of the training, the network restores the layer which was skipped to learn the feature space. In the end, the layers are usually expanded which stay close to manifold for faster learning we use this functionality of ResNets to capture frequency information, the residual connection helps us to maintain features from the previous layer. These features are then fed to the RNN model for classification. The features obtained from ResNets learn the complex features which help in classification. In the above figure, the shortcuts can be directly used if the dimension of the input and the output are the same, denoted by expression (1).

$$y = F(x, W_i) + x \tag{1}$$

Where x is input and y is out vectors for the layers considered for skip connection. The $F(x, W_i)$ is the function representing the residual map. If the number of layers of the residual block is two, the function F can be represented as $F = W_2\sigma(W_1x)$ with σ as ReLU, and neglecting biases. The shortcut connection and element-wise addition are used to operate $F + x$. If the dimension of the input and output are not same, then zero paddings are done and the shortcut is used to match the dimension using the following formula,

$$y = F(x, W_i) + x + W_sx \tag{2}$$

If the matrix dimensions of x and F are matching the (1) is used, for changing dimensions (2) is used. The W_s term in (2) represents the linear projection which is performed using the shortcut or skip connections. The ResNet is used for feature extraction in our model. The residual network that is the “identification of shortcut connection” for retaining the features from consecutive layers is important. The features required for the identification of three classes can be obtained by the extraction of features using ResNet. The ResNets do not require two filters to extract temporal and frequency-based features, rather using ResNet helps to retain features in the consecutive layers. It is a simple notion to increase features we need to increase the number of layers, that simply stacks the layers, but this can cause a vanishing gradient problem because the increase in layers will also increase the back propagation multiple times across the layers. Due to multiplication, the gradient becomes infinitely small and the gradient saturates.

EEG signal is fed to the neural network as an array. $1 \times 1(1D)$ Convolutions are carried out throughout the network for feature extraction. The filter sizes start from 64 and go up to 512. As there are skip connections in ResNet model to retain features from previous layers, whenever there is change in the input filter_size and output filter_size zero padding has been done, it is represented as filter_size/2 indicating the change in filter_size. ResNets helps in retaining the features and reduces overfitting which is caused due to usage of a fully connected layer. Also, there is no max-pooling layer used due to the use of a global average pooling (GAP) layer. The details of ResNet architecture is shown in Figure 3.

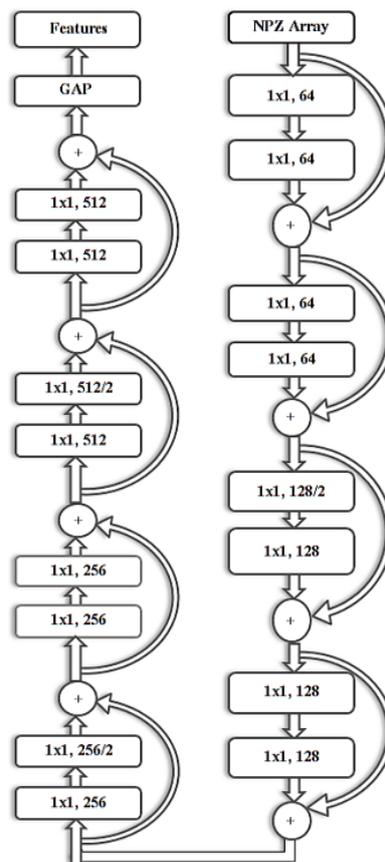


Figure 3. ResNet architecture

2.4. Bi-LSTM based sequence to sequence models with attention

Sequence to sequence method is a deep learning approach that uses an encoder-decoder based machine translation technique to translate the given input sequence into an alternative output sequence with a tag and attention weightage. It uses two recurrent neural networks (RNNs) which work together to predict the next output sequence from the previous input sequence with a special token. The cases in which we need to predict the next state based on the previous state, like predicting driver experiencing the drowsy state which depends on the behavior of past EEG signal behavior, sequence models can be used. As compared to the conventional neural networks where all the inputs to the network and the corresponding outputs are independent, we are choosing bi-directional LSTMs wherein the next state or the output is predicted from the current and past input. The Bi-LSTMs, shown in Figure 4, which have two series RNNs can remember the previous state and with that information, they can predict the next state. LSTM units have a rich internal structure. The various “gates” determine the propagation of information and can choose to “remember” or “forget” information. Compared to the traditional unidirectional RNNs, the operation of which only depends on previous input state, Bi-directional LSTMs process data both in forward and backward directions simultaneously. Hence, the Bi-directional LSTMs are used to remember both past and future data points, and the inputs run in both the directions, one from future to past and one from past to future using two hidden states. In Bidirectional LSTM, the replica of the first recurrent layer is created and the input is given to the first layer in the normal time order, $t=1, \dots, T$, while the reversed input in the time order $t=T, \dots, 1$, is provided to the second or backward layer [19]. The output is computed as the weighted sum of the two layers. The same is represented as (3)-(5).

$$\vec{h}_t = \tanh(Wx_t + \vec{V}h_{t-1} + \vec{b}) \tag{3}$$

$$\overleftarrow{h}_t = \tanh(\overleftarrow{W}x_t + \overleftarrow{V}h_{t+1} + \overleftarrow{b}) \tag{4}$$

$$y_t = (U[\vec{h}_t; \overleftarrow{h}_t] + b_y) \tag{5}$$

The hidden state and feed-forward network’s bias are represented as (\vec{h}_t, \vec{b}) ; the hidden state and backward network’s bias is represented as $(\overleftarrow{h}_t, \overleftarrow{b})$; x_t and y_t are the input and the output of Bi-LSTM, respectively. The sequence-to-sequence model used in our model consists of an encoder and a decoder built with LSTMs. The encoder takes the input as one sequence at a time in the form of vector representation, and the decoder estimates the class for each 30-s input sequence. The long-short-term memory units of the encoder capture the context dependencies of the input and the output target. The decoder thus computes the information of hidden states and predicts the output with the help of Softmax [20]. Since there are three classes to be classified the length of the encoded vector will be three, which is e_1, e_2 , and e_3 .

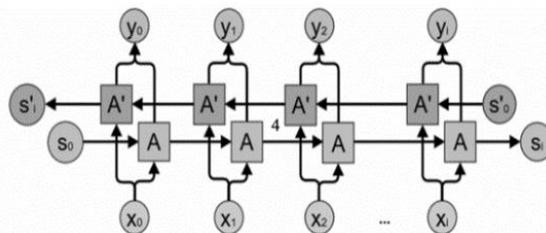


Figure 4. Basic Bi directional LSTM

2.5. The attention network

The encoded sequence of every epoch is further used to get the target sequence using attention network, which is a decoder part of the network. The decoder is also built using LSTMs. In the standard decoder, for every sequence of inputs, the decoder generates the new representation of the input sequence along with a target input element. The last input coming to the decoder is the last effect to update for the hidden state of encoder. Thus, the model has to be biased according to the last element. So, the use of attention mechanism in the model can address such a problem. The attention network learns different portions of the output sequence of encoder for each decoding step along with considering the entire encoder

representation. Hence the decoder learns only the significant input sequence parts during decoding stage. Without the attention mechanism, the decoder operation relies on the hidden vector of the decoder's Bi-LSTM. The sequence to sequence model including attention mechanism is more effective as it includes both encoder's representation and decoder with hidden vector calling the context or attention vector, represented as (ct). Attention weights are computed as a function of $f(\cdot)$, before computing attention vector (ct). The context or attention vector (ct) is probabilities (α_i), relating to the significance of each hidden state, multiplied by a hidden state (e_i).

$$f(h_{t-1}, e_i) = \tanh(W_h h_{t-1} + W_e e_i) \quad (6)$$

$$\alpha_i = \text{softmax}(f(h_{t-1}, e_i)) \approx \frac{\exp(f(h_{t-1}, e_i))}{\sum_{j=1}^n \exp(f(h_{t-1}, e_j))} \quad i = 1, 2, 3 \dots n. \quad (7)$$

$$c_t = \sum_{i=0}^n \alpha_i e_i \quad (8)$$

where α_i is the significant of part i of the hidden state. The $f(\cdot)$, is a combination of the encoder's hidden state (e_i), and decoder's hidden state (h_{t-1}), with the \tanh layer followed by. Later, $f(\cdot)$, is given to the softmax module to calculate α_i for n parts. Then the computation of c_t , is performed by the attention module, which is a weighted sum of all (e_i) and α_i vectors. Hence, while decoding, the model can only consider the important regions of the input vector sequence.

3. IMPLEMENTATION

This section elaborates the details of the implementation of the algorithm; the data set preparation, training procedure, loss calculation, and the evaluation of the model using various metrics.

3.1. The data set preparation

The datasets used for this study are common public datasets of sleep-edf 2013 and 2018 versions which consist of 61 and 197 polysomnograms (PSGs) respectively. Table 1 shows the data corresponding to different sleep classes. We consider the data from Fpz-Cz/Pz-Oz EEG channels for our all analysis. The data set used here does not have an equal distribution of all sleep classes, the sleep stages W and other sleep stages are greater in number compared to N1-state. Such a class imbalance problem is better addressed using deep learning methods compared to conventional machine learning techniques. The loss calculation method used in this paper also helps in dealing with the class imbalance problem. In addition to this, the data set is oversampled wherever required to balance the number of all sleep stage classes.

Table 1. EEG data sets

Data Set	Wake	N1	Sleep	Total
Sleep-EDF-13	8,055	604	6,529	15,188
Sleep-EDF-18	53,641	20,215	1,15,562	1,89,418

3.2. Training procedure with optimizing parameters and hyper-parameters

We feed 30s-epoch to the ResNet for extraction of frequency component related to the sleep stages, which is further connected to sequence-to-sequence models. For each fold, one part is taken for testing rest and is used for training. Finally, all the evaluation results are combined.

The model is evaluated using k -fold cross-validation. The Sleep-EDF 2013 dataset is trained by setting k value to 20 and the sleep edf 2018 data set is trained by setting k value to 10. Cross-validation is used to evaluate the machine learning models with the help of a resampling procedure. The single parameter called ' k ' is used and it denotes to the number of groups to be spilt on the available dataset. Cross-validation is applied to check the behavior of the machine learning model on the unseen dataset. So, when validation is done it is to check whether the increase in the accuracy of the training data also leads to an increase in the accuracy of the dataset which is not previously seen by the network. It is done to minimize overfitting. This method is less biased and optimized for a simple train/test split.

The network is trained for 120 epochs, with RMS prop as the optimizer, this is similar to Gradient descent but the oscillations in the vertical direction are restricted, helping the model to move in the horizontal direction to converge faster with the increase in the learning rate. The mini-batches of size 20 are used with a learning rate set to $\alpha = 0.00001$ and the L2 regularization element with $\beta = 0.001$ to minimize overfitting.

3.3. Loss calculation

There is a problem of data imbalance in the Sleep-EDF dataset; to reduce the effect of this issue we use MSE and MSFE for multiclass classification. The mean squared error (MSE) is a very effective means to determine loss functions in deep learning models. It performs well for a balanced data class, but for an imbalanced dataset it fails, this is due to the fact that it averages the loss by essentially summing up all the errors in the whole dataset. This can effectively estimate the errors if the dataset of both minority and majority class is the same. When the dataset is imbalanced the loss tends to get biased to the majority class as it contributes more to the loss when compared to the minority class. This results in the loss which captures the error of the majority class only. The MSE with MSFE can be used wherein the mean squared false error (MSFE), firstly it averages the error separately in each class and then adds them up.

$$l(c_i) = \frac{1}{c_i} \sum_{j=1}^{c_i} (y_j - \hat{y}_j) \quad (9)$$

$$l(MSE) = \sum_{i=1}^N l(c_i) \quad (10)$$

$$l(MSFE) = \sum_{i=1}^N (l(c_i))^2 \quad (11)$$

where c_i is the class label, C_i is the number of samples, N is the number of available classes, $l(c_i)$ is the error calculated for class c_i . With the help of MSE and MFSE, the loss of both minority and majority classes is considered [21, 22].

3.4. Evaluation metrics

We evaluate the model using overall-accuracy, recall, precision, Cohen's Kappa and F1-score. The overall-accuracy is represented by the ratio of the correct number of predictions to the number of complete input data samples. Precision represents the ratio of correctly classified samples (true positives) to the summation of true positives and classes which are wrongly classified as positive (false positive). It checks out of the ones which the model predicts as positive, how many are actually positive. Recall is represented as the ratio of correctly classified samples to the summation of true positives and false negatives (classified wrongly as negative). It checks out of the ones which are actually positive in the input data samples how many did the model predict positive. Cohen's Kappa is a statistical approach that measures intra(inter) rater reliability for the categorical (qualitative) objects. This method is preferred over direct percent agreement calculations, as it considers the probability of the agreement occurring by chance. It is expected to be a more robust measure [23, 24]. F1-score is the calculation of harmonic mean between precision and the recall.

4. RESULTS AND DISCUSSION

The data sets used are 2013 and 2018 sleep edfs as referred in previous sections as shown in Table 1. The results consisting of confusion matrix and the per-class performance for both versions of data and both FPz-Cz and Pz-Oz, EEG channel is given in Tables 2 and 3 respectively. According to literature, the model can be evaluated by two methods, one method is to use epochs from the same subject for both training and validation which is called as the intra-subject paradigm and the other is the inter-subject paradigm, wherein we use the epochs from different subjects for training and testing. In our study used the second approach that is both trying and testing epochs come from different subjects. Tables 2 and 3, represent the confusion matrices and the related performance parameters for 2013 and 2018 sleep edf data; also, both tables include the results of both FPz-Cz and Pz-Oz channels. True positive values are represented in the main diagonal of the confusion matrix. For all columns, true positive numbers are higher compared to other numbers. The prediction performance parameters, precision, F1 score, recall, and specificity also shown in the table. Performance is slightly low for N1 class compared to other classes, but recall significantly convincing compared to the existing literature. The literature available has mostly recorded the analysis performed for sleep stages, where the N1 stage is part of it. The results of the N1 classification compared to other sleep classes are not convincing. Our results show that the model we proposed works significantly better for the N1 sleep stage also.

The performance is verified for both EEG channels and for both data sets. The performance of the model is improved for the following reasons: i) The sleep stages Alert, drowsy, and sleep are sequential in nature and every next stage is the transition from the previous stage and is related to the previous state. Hence the application of sequence to learning approach is a preferred choice; ii) The use of attention decoder and Bi-LSTM has improved the performance; iii) Use of ResNets allows us to have a deeper network, without compromising on training error and also allows learning temporal and frequency domain

features without having extra layers; iv) The loss calculation procedure used helps to deal with class imbalance problems existing in the data sets, iv) A similar approach with minimum changes can be adapted for classification problems which have sequential behavior and also class imbalance problem.

Table 2. Confusion matrix and the per-class performance for 2013 sleep edf data

Data Sets	Electrodes	Classes	Predicted			Performance matrix for each class (%)			
			Wake	N1	Sleep	PR	RE	F1-score	SP
2013 Sleep EDF	Fpz-Cz	Wake	7212	521	322	89.53	94.70	91.74	88.97
		N1	86	402	114	66.77	40.20	51.58	98.55
		Sleep	317	410	5802	88.86	93.01	90.88	91.87
2013 Sleep EDF	Pz-Oz	Wake	7198	520	348	89.92	93.72	91.78	88.45
		N1	90	410	102	68.10	41.88	53.42	98.62
		Sleep	392	356	5781	88.54	92.77	90.60	91.65

Table 3. Confusion matrix and the per class performance for 2018 sleep edf data

Data Sets	Electrodes	Classes	Predicted			Performance matrix for each class (%)			
			Wake	N1	Sleep	PR	RE	F1-score	SP
2018 Sleep EDF	Fpz-Cz	Wake	48,676	1964	3001	90.74	79.67	84.88	96.13
		N1	6083	12112	2020	59.92	63.09	61.46	95.23
		Sleep	6336	5121	104106	90.08	95.39	92.65	85.73
2018 Sleep EDF	Pz-Oz	Actual Wake	48492	1988	3161	90.40	79.78	84.75	95.99
		N1	5998	12082	2135	59.76	62.67	61.18	95.21
		Sleep	6292	5204	104067	90.05	95.15	92.52	85.64

Table 4. Performance comparison with other related algorithms

Authors	Data set	Electrodes	No.of Classes	CV	Overall Accuracy	Overall F1 score	Cohen Kappa
A. R. Hassan <i>et al.</i> [8]	Sleep-edf data	Pz-Oz	05	20	90.8	80.0	--
A. Supratak <i>et al.</i> [16]	Sleep-edf data	Fpz-Cz	05	20	82.0	76.9	0.76
O. Tsinalis <i>et al.</i> [25]	Sleep edf data	Fpz-Cz	05	20	78.9	73.9	--
Sajad Mousavi <i>et al.</i> [21]	Sleep edf-2013 data	Fpz-Cz	05	20	84.26	79.66	0.79
Sajad Mousavi <i>et al.</i> [21]	Sleep edf-2013 data	Pz-Oz	05	20	82.83	82.83	0.77
Sajad Mousavi <i>et al.</i> [21]	Sleep edf-2018 data	Fpz-Cz	05	10	80.03	80.03	0.73
Sajad Mousavi <i>et al.</i> [21]	Sleep edf-2018 data	Pz-Oz	05	10	77.56	77.56	68.94
Mikito Ogino <i>et al.</i> [26]	MindWave, Neurosky	Fp1-A1	05	20	72.70	NA	--
CNN-BiLSTM	Sleep edf-2013 data	Fpz-Cz	02	20	90.25	93.32	--
(Our alternate method)							
AlertNet (Fpz-Cz)	Sleep edf-2013 data	Fpz-Cz	03	20	87.92	78.06	0.78
AlertNet (Pz-Oz)	Sleep edf-2013 data	Pz-Oz	03	20	87.73	78.60	0.79
AlertNet -(Fpz-Cz)	Sleep edf-2018 data	Fpz-Cz	03	10	87.05	79.66	0.79
AlertNet -(Pz-Oz)	Sleep edf-2018 data	Pz-Oz	03	10	86.92	79.48	0.79

5. CONCLUSION

The proposed deep neural network-based model architecture for EEG based driver alertness detection uses ResNets and Bi-LSTM, sequence to the sequence learning approach. The ResNets are used for feature extraction, with the skip-connections helps to retain the information from the alternate layers. This helps in retaining the features without adding any extra layers and learns deeper into the network with no increase in training error. The sequence to sequence model helps in learning the complex dependencies present in the EEG signal. The model's performance for the N1 sleep stage is better compared to existing models. Hence, the model can be used for future usage of automatic classification of sleep using raw EEG signals.

REFERENCES

- [1] O. Rich, Tonya L, and Bernadette T Gillick, "Electrode Placement in Transcranial Direct Current Stimulation-How Reliable Is the Determination of C3/C4," *Brain sciences*, vol. 9, no. 3, pp. 69, 2019.
- [2] Rajeev Agarwal and Jean Gotman, "Computer-assisted sleep staging," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 12, pp. 1412-1423, 2001.
- [3] Sleep Foundation, 2020, [Online]. Available: <https://www.sleepfoundation.org/how-sleep-works/stages-of-sleep>.
- [4] Rechtschaffen A, "A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects," *Brain information service*, 1968.

- [5] R. B. Berry, R. Brooks, C. E. Gamaldo, C. E., Harding, S. M., Marcus, C., and Vaughn, B. V., "The AASM Manual for the scoring of sleep and associated events, Rules, Terminology, and Technical Specifications," *American Academy of Sleep Medicine*, vol. 176, pp. 1-7, 2012.
- [6] Moser, D., Anderer, P., Gruber, G., Parapatics, S., Loretz, E., Boeck, M *et al.*, "Sleep classification according to AASM and Rechtschaffen & Kales: effects on sleep scoring parameters," *Sleep*, vol. 32, no. 2, pp. 139-149, 2009.
- [7] Libourel, P. A., Corneyllie, A., Luppi, P. H., Chouvet, G., and Gervasoni, D., "Unsupervised online classifier in sleep scoring for sleep deprivation studies," *Sleep*, vol. 38, no. 5, pp. 815-828, 2015.
- [8] A. R. Hassan and A. Subasi, "A decision support system for automated identification of sleep stages from single-channel EEG signals," *Knowledge-Based Systems*, vol. 128, pp. 115-124, 2017.
- [9] E. Alickovic and A. Subasi, "Ensemble SVM Method for Automatic Sleep Stage Classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1258-1265, 2018.
- [10] Sunagawa, G. A., Séi, H., Shimba, S., Urade, Y., and Ueda, H. R., "Faster: an unsupervised fully automated sleep staging method for mice," *Genes to Cells*, vol. 18, no. 6, pp. 502-518, 2013.
- [11] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773-780, 1989.
- [12] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., and Khudanpur, S., "Recurrent neural network-based language model," *Eleventh annual conference of the international speech communication association*, 2010, pp. 1045-1048.
- [13] A. Graves, A. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645-6649.
- [14] Andrej Karpathy and Li Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 664-676, 2017.
- [15] P. R. Davidson, R. D. Jones, and M. T. R. Peiris, "Detecting Behavioral Microsleeps using EEG and LSTM Recurrent Neural Networks," *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 5754-5757.
- [16] A. Supratak, H. Dong, C. Wu, and Yike Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998-2008, 2017.
- [17] A Sors, S. Bonnet, Sebastien Mirek, L. Vercueil, and J.-F. ois Payen, "A convolutional neural network for sleep stage scoring from raw single-channel EEG," *Biomedical Signal Processing and Control*, vol. 42, pp. 107-114, 2018.
- [18] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia tools and applications*, vol. 77, no. 9, pp. 10437-10453, 2018.
- [19] A. Sherstinsky, "Fundamentals of the recurrent neural network (rnn) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, 2020, Art. no. 132306.
- [20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3104-3112, 2014.
- [21] Mousavi, S., Afghah, F., and Acharya, U. R., "SleepEEGNet: Automated sleep stage scoring with sequence-to-sequence deep learning approach," *PloS one*, vol. 14, no. 5, 2019, Art. no. e0216456.
- [22] S. Wang, Wei Liu, Jia Wu, L. Cao, Q. Meng, and P. J Kennedy, "Training deep neural networks on imbalanced data sets," *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, 2016, pp. 4368-4374.
- [23] S. M. Vieira, U. Kaymak, and J. Mc Sousa, "Cohen's kappa coefficient as a performance measure for feature selection," *International Conference on Fuzzy Systems*, Barcelona, Spain, 2010, pp. 1-8.
- [24] H. Jeong and W. Park, "Developing and Evaluating a Mixed Sensor Smart Chair System for Real-time Posture Classification: Combining Pressure and Distance sensors," *IEEE Journal of Biomedical and Health Informatics*, 2020.
- [25] Tsinalis, O., Matthews, P. M., Guo, Y., and Zafeiriou, S., "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," *arXiv preprint arXiv:1610.01683*, 2016.
- [26] Mikito Ogino, Yasue Mitsukura, "Portable Drowsiness Detection through Use of a Prefrontal Single-Channel Electroencephalogram," *Sensors*, vol. 18, no. 12, 2018, Art. no. 4477.

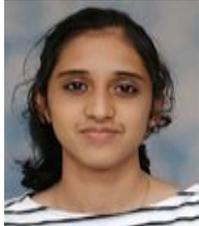
BIOGRAPHIES OF AUTHORS



Prabhavathi C. Nissimagoudar is currently pursuing PhD and working as Associate Professor, in School of ECE, KLETU, Hubballi. Her research is in the areas of automotive embedded systems, Advance Driver Assistance Systems (ADAS) and Intelligent Biomedical systems.



Anilkumar V. Nandi is currently working as Professor, in Department of ECE, B.V.C.E.T Hubballi, India. He post graduated from IIT Kharghpur, India in the area of MEMS and received PhD from VTU, Belagavi, India. His research is in the area of VLSI design, Embedded Systems, MEMS and Intelligent Biomedical Systems.



Aakanksha K. Patil is working as an undergraduate research student at the KLETU, India. Her research interests include development of deep learning algorithms for automotive intelligent systems. She worked at Continental automotive as an internship trainee during her final year of graduation.



Gireesha H. M. is working as Assistant Professor in the School of ECE at KLE Technological University, Vidyanagar, Hubballi. He post graduated in Biomedical Signal Processing and Instrumentation from SJCE Mysore. His research is in the area of soft computing, signal and image processing, and advance driver assistance systems.