Driver alertness detection using CNN-BiLSTM and implementation on ARM-based SBC

PRABHAVATHI C. NISSIMAGOUDAR[1]
ANILKUMAR V. NANDI[2]
GIREESHA H M[3]

School of Electronics and Communication Engineering
KLE Technological University
Hubballi, Karnataka, India-580031
[1]pcngoudar@gmail.com
[2]anilnandy@bvb.edu
[3]gireesh_hm@bvb.edu

Abstract. Driver alertness detection is one of the significant automotive-related features to Advance Driver Assistance Systems (ADAS). Electroencephalogram based alertness detection is a direct method of determining consciousness level. In this paper, an algorithm using a one-dimensional convolution neural network and bidirectional LSTM to learn the alertness level from EEG signals is proposed. The algorithm is implemented on an ARM-based single-board computer (SBC) for performance analysis. Real-time detection of drowsiness is necessary to alert the driver whenever he is about to sleep. Most of the existing methods focus on off-line analysis for interpreting the driver's state. The proposed method uses deep learning techniques to characterize and train the system, and the trained Model is ported to ARM SBC for real-time performance. Physionet sleep edf data with single-channel FPz-Cz is used for training the Model. The trained CNN-LSTM based Model gave an accuracy of 93.3 and the test model gave an accuracy of 89.4 percentage when tested with real-time signals using the Neurosky mind wave electrode. To reduce road accidents occurring due to the driver's drowsiness, it is necessary to monitor driver alertness and alarm when necessary continuously.

Keywords: Electroencephalogram(EEG), Feature extraction, Driver state, Convolutional neural network, Bidierectional Long-Short Memory (Bi-LSTM).

(Received September 6th, 2020 / Accepted November 16th, 2020)

1 Introduction

According to the AAA Foundation for traffic safety, drowsy driving is a safety hazard as drunk driving, and 20 percentage of all fatal accidents in the USA are due to drowsiness. Even the National Highway Traffic Safety Administration (NHTSA) agrees to it, and also CSI research center says that drowsy driving is more dangerous than drunk driving and texting while driving [1]. Driver state detection is an essential feature in automotive, that provide an early warning system to avoid accidents. Various literature methods deal with analyzing driver behavior like eye blinking and vehicle behavior like vehicle drifting from the lane. Measurement of driver's physiological signals like Electroencephalogram (EEG) gives a more direct and accurate detection method than conventional techniques. EEG represents the neural activity happening inside the human brain. EEG has information on both frequency and amplitude. The brain activity corresponding to different conditions is represented by various frequency bands of EEG known as EEG rhythms. These rhythms are named as theta (0-4Hz), delta (4-8Hz), alpha (8-12 Hz), beta (12-30 Hz) and gamma (30-100 Hz) frequency bands. The extraction and analysis of these frequency bands' representative features is vital in determining the human brain's different states. EEG analysis includes various steps comprising of data acquisition using EEG sensors/electrodes, pre-processing to separate different EEG rhythms, extraction and selection of representative features, and finally, classification of EEG features according to varying conditions of the brain.

Many existing studies use machine learning techniques wherein the classification is performed on hand-engineered extracted features from the EEG signal. Recently the availability of large data sets of EEG has led to applying deep learning techniques to EEG analysis. Deep learning techniques provide robust automatic learning of EEG features compared to conventional hand-engineered feature learning models. This paper discusses methods for the analysis and classification of EEG signals to detect the driver's cognitive state. The algorithm is also implemented on the ARM single board computer for real-time testing. To classify EEG signals deep learning-based CNN-LSTM technique is adopted, a combination of convolution neural network and recurrent neural network (RNN).

The structure of the paper is as follows; in section two we discuss the related existing literature for the proposed work, section three discusses methodology related model architecture, based on deep convolution neural networks and Bi-LSTMs. section four details about experiments and results involving datasets, virtual simulation environment, parameter optimization, porting the Model to ARM SBC. Section four details observations and discussions on the Model's parameters and classifier results and the test results obtained with the Neurosky sensor and ARM SBC hardware. Section five concludes the paper.

2 Related Work

The accuracy of EEG classification depends on the representative features selected and feature learning models. A lot of research has been carried out using hand-engineered methods of feature learning using traditional machine learning techniques.

Mohammed Diykh et al. discussed identifying six sleep stages using the statistical time-domain approach with the structural graph similarity and K-means classifier. [3][16]. Farhan Riaz et al. proposed a feature extraction method that used a time-frequency analysis of non-stationary EEG signals using empirical mode decomposition (EMD). [4][17]. Ning Wang, et al. suggested a compact feature representation method to determine epileptic seizure conditions. The proposed feature elimination method overcomes the redundancy and noise components for obtaining low dimensional and independent feature vector [5][14]. Robert Jenke et al. proposed a method to determine emotions using EEG signals. They experimented with different feature selection methods and suggested features selection using multivariate methods to give accurate results compared to univariate methods. [6].

Traditional methods of EEG classification used machine learning approaches, wherein the feature extraction selection plays a major role. The features of the signals in time/frequency/time-frequency domains [19] are extracted and with a suitable elimination technique, only significant features were retained. Some methods also use non-linear techniques like wavelet transforms [7][13] and also empirical methods of feature extraction and selection are discussed in the literature.

Recently many researchers are exploring deep learning methods for EEG signal analysis. The availability of huge data size for such signals is triggering their research. The literature has recorded that deep learning approaches proved efficient for image and video analysis.

Yaguang Jia et.al proposed a method to extract significant features for classifying signals with high dimensionality and multi-channel properties from a steady-state motion visual evoked potential signals using deep belief neural (DBN) network and stacked restricted Boltzmann machine (RBM). The proposed method extracts local features from every channel and further fuses the information before giving it to classifiers. To achieve higher accuracy and lower inter-subject variability.[8][15]. Mehdi Hajinoroozi et al. proposed an approach which incorporates, channel-wise convolution neural network with restricted Boltzmann machine. The algorithm showed convincing results for Independent Component Analysis (ICA) transformed data compared to raw EEG data [9][12].

Deep learning methods using multiple layers of linear and non-linear processing have been used in some of the recent studies. O. Tsinalis, et al. used Convolution neural networks (CNN) to extract time-invariant features using a single-channel electrode [4]. Wang N et al proposed Deep Belief Nets (DBNs) to learn features based on probabilistic representations from pre-processed raw polysomnograms [5]. These methods mainly focused on time-invariant features and did not consider temporal information. The traditional hand-crafted feature extraction methods which require the model to choose a suitable feature extraction and selection methods before using the classification technique. These techniques provide the desired accuracy with proper tuning at feature extraction/selection and classification levels. The recent deep learning approach does not require a handcrafted approach to obtain significant features, instead, the model learns the features automatically.

Compared to the conventional neural networks, the RNNs have a memory element, from which the decision is based on the previous history. This gives an advantage of decisions based on the prediction of the next state[20]. In this paper, we propose a Convolution Neural Network (CNN) based feature learning model for determining driver alertness level that also uses a BiLSTM. The combination of CNN based feature learning model and RNN based classification technique is adopted in this work. The model automatically learns the features and can be effectively used for classification with higher accuracy for predicting the driver states. The model is ported to ARM SBC for real-time testing of the algorithm.

3 Methodology

The proposed model uses two main parts as shown in fig:1. One training model and the other is the prediction model. The model is trained using CNN and Bidirectional LSTM and the trained model is ported to ARM-based SoC for testing. The training model is further divided into two parts as shown in fig:2, the first part is the feature learning model which learns time-invariant features using two layers of CNN and the second part is RNN based bidirectional LSTM which used for learning temporal features such as state transition rules. The data used by this network is single-channel EEG 5-s epochs.

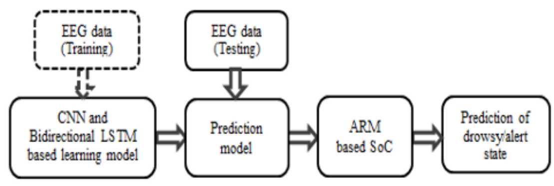

Figure 1: Block diagram of the proposed approach

3.1 Time in-variant Feature learning model

The feature learning model uses two CNNs with different filter sizes. One CNN to learn temporal information which has a smaller filter size and other CNN with a larger filter size to learn frequency information [2][10]. Four CNN layers and two max pooling layers are used. The convolution layer performs 1D convolution with its filters, followed by normalization. The activation function used at the output layer is rectified linear unit (ReLu). Drop out is further used to reduce the feature size.

$$h_i^1 = CNN_1(a_i) \tag{1}$$

$$h_i^2 = CNN_2(a_i) \tag{2}$$

$$X_i = h_i^1 || h_i^2 \tag{3}$$

Where hi1 are temporal features from layer 1 of CNN and hi2 are frequency features of CNN layer 2 and Xi is the concatenated feature set.

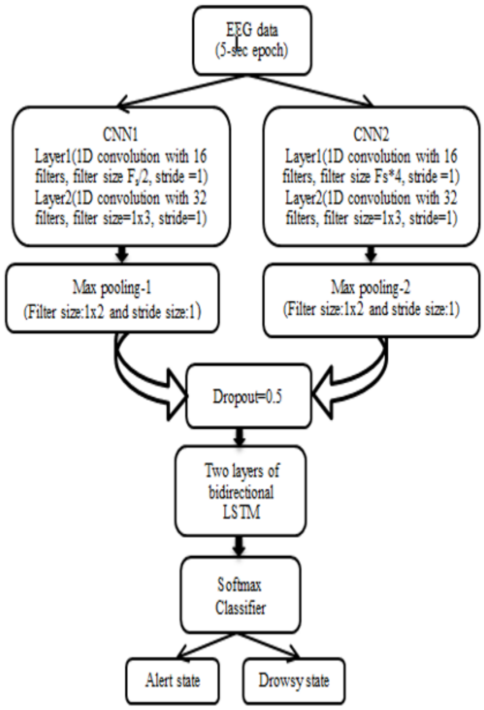

Figure 2: CNN-BiLSTM based training model

3.2 RNN based bi-directional LSTM for temporal features

Bi-directional LSTM is used to learn temporal features such as state transition rules. The concatenated information from CNN layer 1 and layer 2 are input to this section. The Bi-LSTM uses both forward and backward propagation which run independently to learn from past and future representation. The next state is predicted previous(memory) conditions. Suppose there are Xi input features from previous state representing 1-s EEG epoch, bi-directional LSTM learning is represented as,

$$h_t^f, c_t^f = LSTM\theta_f(h_(t-1)^f, c_(t-1)^f, x_t) \quad (4)$$

$$h_t^b, c_t^b = LSTM\theta_b(h_(t+1)^b, c_(t+1)^b, x_t) \quad (5)$$

Where LSTM is performed on data $x_t, h_t^f, c_t^f, \theta_f$ and h_t^b, c_t^b, θ_b are parameters of forward and backward propagation.

3.3 Model specifications

For the time-invariant CNN based feature learning model, Two CNNs with two layers each. Layer 1 of CNN1 with filter size as Fs/2 and stride size of 1 and layer1 of CNN2 had a filter size of Fs*4 and stride size of 1. Layer 2 of CNN1 and CNN2 had the number of filters as 32, filter size as 3, and stride size was 1. The pooling layer used max-pooling with filter size 2 and stride size 1. Then 0.5 dropout operation was performed on concatenated data. For Bi-LSTM cell size was 512. From this, we could retain only significant features and avoiding the problem of overfitting.

3.4 Algorithm

The model is trained end-to-end using the backpropagation technique. A two-level training approach is used to train the model. At the first level model learns for time-invariant features using two CNNs. The model further learns for state transition rules to predict the state using BI-LSTM. Level1: The model is trained using two CNNs using a supervised approach and Softmax function at the output layer. Then the model uses Adam optimizer with learning rate lr. Level 2: The pretrained model is further trained using Bi-LSTM which uses forward and backpropagation techniques to learn from the past and future. The state is predicted using state transition rules.

4 Experiments, Hardware Implementation, and Results

The experimentation was carried out using data labeled as alert and drowsy conditions. The feature learning model of CNN-Bi-LSTM based network was experimented for varying model parameters and verified on ARM hardware. The model experimented with two data sets i) Publicly available sleep edf ii) Neurosky mind wave mobile sensor data collected from a virtual driving simulation environment

4.1 Sleep edf data sets

The data sets used for training were obtained from publicly available sleep data from Physionet database [11]. The sleep data files known as Polysomnograms(PSG.) are in the European data formats(edf). The related annotations are in hypnogram files (hypnogram.edf). The data consists of data recorded from 61 subjects for the whole night and has been categorized into different sleep stages(W(wake), Sleep stage 1 to 4, Rapid Eye movement(REM), and unidentified data as ?(Not scored)). Sleep edf data consists of recording from Fpz-Oz and Pz-Oz locations of EEG electrodes. The sampling frequency of the data is 100Hz. For the experimentation purpose, the data is separated as W and S1, the sleep stage 1 or drowsy data, using Polyman tool.

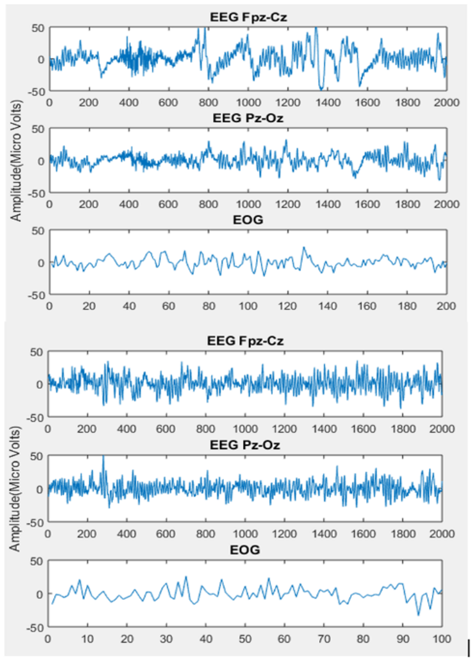

Figure 3: EEG signals for sleep stage S1 and wake stage W

4.2 Data collection using the Neurosky Mindwave sensor in Virtual driving environment

The raw EEG data is obtained from the brain using the Neurosky Mindwave sensor (NeuroSky Inc., San Jose, CA, USA)., a commercially available headset which is a noninvasive type of brain-computer interface. The sensor consists of a single dry electrode with a dimension of 12mm x 16 mm, which is to be placed on Fp1(Frontal left lobe), according to the 10-20 international system. The data is provided through TGAMI(ThinkGear ASIC

Module) integrated circuit. The sensor uses an electrode clipped to the left ear lobe as a reference. The sampling rate of the data is 512 Hz. The sensor is placed on the scalp to acquire raw EEG information. The acquired raw EEG is used for further processing.

4.3 The virtual driving simulation environment

A virtual driving simulation environment (figure 4) is used to collect data using the Neurosky sensor. The environment provides a vision of 180Â° from a driver seat. Three front projectors simulate the real driving views with inner and outer rear view mirrors with a perceived field of vision close to 360Â°. The sound system is also provided with acoustics from inside and outside the vehicle.

The recordings from the Neurosky mind wave sensor consist of EEG signals from 18 participants, who took the virtual driving simulator test. The duration of the data was collected for three to four hours of monotonous driving between the periods of 1.30 pm to 5.30 pm. EEG signals acquired are from FP1. The speed limit of the vehicle was set to 60km/h. The data was collected in the form of raw EEG into EDF(European Data Format) using MATLAB software.

Figure 4: Performance of training and validation for sleep edf data

4.4 Algorithm Implementation

The implementation of the algorithm was done using TensorLayer, the extended library from Google Tensorflow. The numerical computations involving CPU and GPUs during training and validation are performed using the TensorLayer library. The model is trained and validated from the collected dataset a 2 GB Nvidia 840 MX GPU system; dependencies like python, NumPy, CUDA toolkit 8.0 and CuDNN v5, TensorFlow-gpu, matplotlib , scipy, pandas , tensor layer and sci-kit learn were installed. The obtained checkpoints are saved as a model that is deployed on ARM SBC with the preinstalled TinkerOS (a Debian Linux derivative). All the dependent libraries such as python, NumPy, and sci-kit learn, are installed which are compatible with the hardware to run the inference. The training time for each fold was around 2.5 hours and testing time was nearly 40ms for 20 epochs.

4.5 Model Extraction

The trained model is extracted and saved in two methods,

- The weights are saved as .h5 file and the architecture is saved as .json file. While loading the model, first the architecture is loaded, and then weights are added to it.
- Alternatively, the model with both training weights and architecture is saved as .model file.

4.6 Hardware Implementation

Hardware implementation to deploy a neural network requires special attention. Training and testing both could be performed on the microcontroller having higher RAM and a high-performance processor. Training on any personal computer and deploying the model file on embedded hardware seems to an intelligent choice. We selected the ARM-based Single board computer with ARMv7-A 32-bit architecture, CPU with quad-core 1.8GHZ ARM Cortex-A7, 600MHz Mali T760 MP4 GPU[18]. This processing ability was sufficient to run our application.

4.7 Virtual Environment

The testing of the algorithm is performed on the virtual environment. A virtual environment tool that provides a platform wherein the dependencies are required by each project can be maintained separately by creating isolated virtual environments. Python process uses a heavy library, as in our case Keras, TensorFlow, TensorLayer, and also the hardware ARM SBC also has several processes of the operating system. In such conditions, the virtual environment helps maintain dependencies of both Python and hardware OS. In the above case of two processes, we are cutting off the dependency of our application process with that of the default processes of the ARM SBC. Thus by using a virtual environment for application processes, the load on OS is reduced i.e. the dependencies of each process are isolated from the system and each other.

4.8 Experimenting with the model

Experimentation was done for variations in different architecture for the first part, feature learning model, which includes CNN layers. The drop out of 0.5 and Bidirectional LSTM with 512 cell size was fixed. The variations in the CNN layers are shown in table 1 along with the results. Convolution layers, number of filters, filter size, and stride size were varied and the results are compared. The model also experimented with various tunable parameters, which included, based on the number of epochs, batchsize, numunits4, embedsize, inputdepth, nchannels, maxtimestep, outputmaxlength. After experimentation finally, the architecture was freezed to two CNNs with two layers and two Bi-LSTM. The model was evaluated using k-fold cross validation where k was set to 10. For each fold (Ts-Ts/k) recordings to train the model and Ts/k to test the model where Ts is the total number of samples in the data set. This procedure was iterated 10 times to test all the readings and computed the performance metrics. Table 2 and 3 show the accuracy results of training, validation, and testing for both sleep edf and Neurosky data.

Table 1: Results for different architectures for feature learning model

CNN Architecture	Convolution Layer-1	Convolution Layer-2	Maxpool layer
CNN-1	16,Fs/2,1	32,Fs*4,1	2,1
CNN-2	16, Fs*4,1	32,3,1	2,1
CNN-1	32, Fs/2,1	64, Fs*4, 1	2,1
CNN-2	32, Fs*4,1	64,3,1	2,1
CNN-1	128, Fs/2,1	128, Fs*4, 1	2,1
CNN-2	128,Fs*4,1	128,3,1	2,1

Table 2: Accuracy results of training, validation, and testing (Sleep EDF)

	Training	Validation	Testing
Wake	94.39	92.73	89.84
Sleep Stage1	93.23	91.67	90.67

Table 3: Accuracy results of training, validation, and testing (Neurosky)

	Training	Validation	Testing
Wake	92.56	90.73	88.56
Sleep Stage1	91.42	91.15	89.34

The average accuracy of around 90% is observed for both the classes. The performance of the same is shown in figure 5 and 6. The accuracy remains around 90% for varying epoch size. The epoch size of 60 is found to be appropriate.

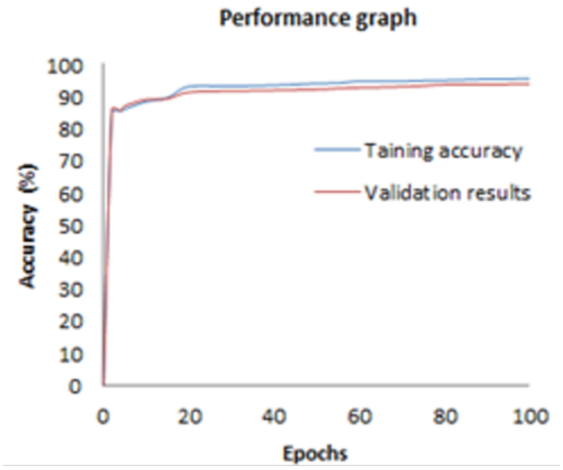

Figure 5: Performance of training and validation for sleep edf data

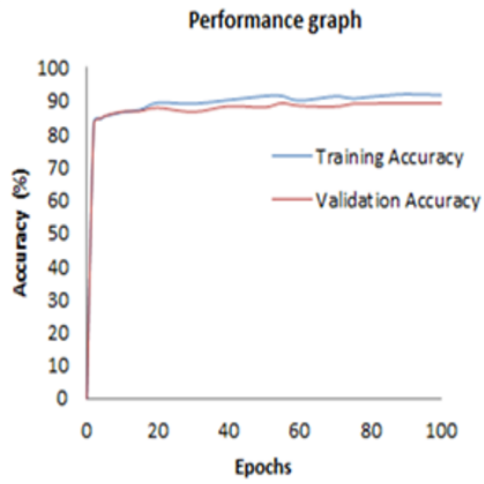

Figure 6: Performance of training and validation for sleep edf data

4.9 Parameter Tuning

The first part of training which included CNN based representation learning was trained using ADAM optimizer with a learning rate as 10-4, beta1 as 0.9, and beta2 as 0.999. The second part of the training which is done using sequence residual learning is also with ADAM optimizer, but with learning rates as lr1 and lr2 as 10-6 and 10-4 respectively.

4.10 Performance metrics

The network was evaluated for different performance measures viz. each class accuracy, overall accuracy, each class precision (PR), each-class recall (Re), and each class F1 score(F1) and sensitivity(S). To compute per class metrics we considered a single class as a positive class and remaining classes as a negative class. Table 4 consists of the confusion matrix and Table 5 contains the results of different evaluation parameters. These results are for recorded sleep edf and Neurosky sensor readings. We could use the same model without changing the parameters for both sleep edf and Neurosky data. The results show similar performance for both.

Table 4: Confusion Matrix

Data Sets	Classes	Wake	Sleep Stage1
Sleep edf	Wake	3326	274
	Sleep stage1	186	3414
Neurosky sensor Data	Wake	1084	114
	Sleep stage1	66	389

Table 5: Performance parameters

Data Sets	Classes	PR	RE	F1-score	SP
Sleep edf	Wake	92.38	94.70	92.98	92.57
	Sleep stage1	94.83	92.57	93.67	94.70
Neurosky sensor	Wake	90.48	94.26	92.33	77.33
	Sleep stage1	85.49	77.33	81.20	94.42

4.11 Performance Comparison with existing approaches

Also, when the test results are compared with the existing state of art methods that use only one channel data, our results are convincing for both sleep edf and Neurosky sensor test data. The comparison is shown in table 6 for the overall accuracy and overall F1 score compared with existing methods. The results of sleep edf gave better accuracy compared to Neurosky sensor data as the sleep edf data used for training was large compared to the Neurosky data. The results are 90.25% and 83.67% for sleep edf and Neurosky sensor data respectively which is higher compared to existing methods.

Table 6: Comparison with existing approaches

Authors	Data set and Electrodes	Overall Accuracy	Overall F1 score
O. Tsinalis, P. M. Matthews, Y. Guo, S. Zafeiriou.[21]	Sleep edf Fpz-Cz	74.8	69.8
A. R. Hassan, A. Subasi[23]	Sleep-edf Pz-Oz	90.8	80.0
A. Supratak, H. Dong, C. Wu, and Y. Guo[24]	Sleep-edf Fpz-Cz	82.0	76.9
O. Tsinalis, P. M. Matthews, Y. Guo, [22]	Sleep edf Fpz-Cz	78.9	73.9
Mikito Ogino, Yasue Mitsukura[25]	Neurosky Mindwave mobile Fp1-A1	72.7	NA
CNN-BiLSTM (Proposed method)	Sleep edf Fpz-Cz	90.25	93.32
CNN-BiLSTM (Proposed method)	Neurosky Mindwave mobile Fp1-A1	83.67	81.76

4.12 Discussion

The model was verified for training for both Neurosky and Sleep edf data. The performance was evaluated for overall accuracy, per class accuracy and other performance metrics. Results showed that the accuracy was around 90% when trained on the laboratory machines, as discussed in the earlier sections. Also, as the algorithm was expected to work in the real-world environment, the trained model was tested on a embedded hardware and real-time signals. Hence, the trained model was ported to hardware, in our case it is ARM-based SBC and tested for real-time signals of Neurosky mindwave sensor. The results of the tests are as shown in table 7 and 8. The test data was provided from the Neurosky sensor used in a virtual simulation environment. The model behaved with a test accuracy of 80.95%, which is slightly low compared to the testing and validation results on the desktop PC. The results are in the acceptable range and hence the model can be used for real-time testing of driverâs alertness. The model works independently with a response time of 30ms.

Table 7: Confusion matrix for Neurosky test data with ARM SBC

Data Sets	Classes	Wake	Sleep Stage1
Neurosky data	Wake	108	17
	Sleep stage1	7	34

Table 8: Evaluation of Performance parameters for test data with ARM SBC

Classes	PR	RE	F1-score	SP	Acc
Wake	86.4	93.1	89.62	66.66	86.40
Sleep stage1	82.92	66.6	73.90	93.10	80.95

5 Conclusion

The proposed CNN and Bi-LSTM based deep learning model automatically learns the driver's alertness using single-channel EEG. The model is trained end-to-end using the backpropagation technique. A two-level training approach is used to train the model to overcome the class imbalance problem and to encode temporal information into the model. The model learns time-invariant features using convolution neural networks and transition rules related to stages using bidirectional LSTMs from the EEG epochs. The model was evaluated using two different types of data sets i.e sleep edf and for the Neurosky mindwave senor data. The implementation of the algorithm was done on ARM-based SBC and tested for Neurosky sensor data. The trained CNN-LSTM based model gave an accuracy of 93.3% and the test model gave an accuracy of 89.4% when tested with real-time signals using the Neurosky mind wave electrode. As the model learns automatically without using hand-engineered features, it is expected to be the better approach for detecting driver alertness for automotive applications.

References

[1] A. J. Lim and W. D. Winters. *A Practical Method for Automatic Real-Time EEG Sleep State Analysis,* in IEEE Transactions on Biomedical Engineering, vol. BME-27, no. 4, pp. 212-220, April 1980.

[2] T. P. Yunck and F. B. Tuteur. *Comparison of decision rules for automatic EEG classification,* in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-2, no. 5, pp. 420-428, Sept. 1980.

[3] M. Diykh, Y. Li and P. Wen. *EEG Sleep Stages Classification Based on Time Domain Features and Structural Graph Similarity,*in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 24, no. 11, pp. 1159-1168, Nov. 2016.

[4] F. Riaz, A. Hassan, S. Rehman, I. K. Niazi and K. Dremstrup. *EMD-Based Temporal and Spectral Features for the Classification of EEG Signals Using Supervised Learning,*in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 24, no. 1, pp. 28-35, Jan. 2016.

[5] Wang N, Lyu MR. *Extracting and Selecting Distinctive EEG Features for Efficient Epileptic Seizure Prediction,*IEEE J Biomed Health Inform.pp.1648-1659,2015.

[6] R. Jenke, A. Peer and M. Buss.*Feature Extraction and Selection for Emotion Recognition from EEG,*in IEEE Transactions on Affective Computing, vol. 5, no. 3, pp. 327-339, 2014.

[7] Nissimagoudar P.C., Nandi A.V., Gireesha H.M. *A Feature Extraction and Selection Method for EEG Based Driver Alert/Drowsy State Detection,*Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019). SoCPaR 2019.

[8] Yaguang Jia, Jun Xie, Guanghua Xu, et al.*A separated feature learning based DBN structure for classification of SSMVEP signals,* Conf Proc IEEE Eng Med Biol Soc. 2017;2017:3356-3359.

[9] Garrett D, Peterson DA, Anderson CW, Thaut MH. *Comparison of linear, nonlinear, and feature selection methods for EEG signal classification,*IEEE Trans Neural Syst Rehabil Eng. 2003;11(2):141-144.

[10] Piotr W Mirowski, Yann LeCun, Deepak Madhavan, and Ruben Kuzniecky. *Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG,*In Machine Learning for Signal Processing, 2008. pp. 244â249. IEEE, 2008

[11] A Rechtschaffen, AE Kales. *A manual of standardized terminology, techniques, and scoring systems for sleep stages of human subjects,*Los Angeles, CA: UCLA Brain Information Service. Brain Research Institute 10 (1968).

[12] https://physionet.org/physiobank/database/sleep-edfx/

[13] Ali H Shoeb and John V Guttag. *Application of machine learning to epileptic seizure detection,*In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 975â982, 2010.

[14] D. Garrett, D. A. Peterson, C. W. Anderson and M. H. Thaut. *Comparison of linear, nonlinear, and*

feature selection methods for EEG signal classification, in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 11, no. 2, pp. 141-144, June 2003.

[15] Berthomier C, Drouot X, Herman-StoÃ¯ca M, Berthomier P, Prado J, Bokar-Thire D, Benoit O, Mattout J, d'Ortho MP. *Automatic analysis of single-channel sleep EEG: validation in healthy individuals,* pp.1587-95, Nov 2007.

[16] Zoubek, LukÃ¡Å¡ Charbonnier, Sylvie Lesecq, Suzanne Buguet, Alain Chapotot, Florian. *Feature selection for sleep/wake stages classification using data driven methods. Biomedical Signal Processing and Control,* pp.171-179, 2007.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks,* In Advances in neural information processing systems, pages 1097â1105, 2012.

[18] https://www.asus.com/uk/Single-board-Computer/TINKER-BOARD/

[19] 19. P.C. Nissimagoudar, Anilkumar V Nandi *Precision Enhancement of Driver Assistant System Using EEG Based Driver Consciousness Analysis Classification,* Computational Network Application Tools for Performance Management, 247-257,2020.

[20] Nissimagoudar P.C., Nandi A.V., Gireesha H.M. *Deep Convolution Neural Network-Based Feature Learning Model for EEG Based Driver Alert/Drowsy State Detection,* In Abraham A., Jabbar M., Tiwari S., Jesus I. (eds) Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019). SoCPaR 2019. Advances in Intelligent Systems and Computing, vol 1182.

[21] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou. *Automatic sleep stage scoring with single-channel EEG using convolutional neural networks,* 2016.

[22] O. Tsinalis, P. M. Matthews, and Y. Guo. *Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders,* Ann.Biomed. Eng., vol. 44, no. 5, pp. 1587â1597, 2016.

[23] A. R. Hassan and A. Subasi. *A decision support system for automated identification of sleep stages from single-channel EEG signals,* Knowledge Based Syst., vol. 128, pp. 115â124, Jul. 2017.

[24] Supratak A, Dong H, Wu C, Guo Y. *DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG,* IEEE Trans Neural Syst Rehabil Eng. 2017 Nov;25(11):1998-2008.

[25] Ogino, Mikito; Mitsukura, Yasue. *Portable Drowsiness Detection through Use of a Prefrontal Single-Channel Electroencephalogram,* Sensors 18, no. 12: 4477,2018.