Classification of Sleep Stages using EEG signals and Deep Learning Techniques

Siddharth Sanghavi [1], Palash Rathod [2], Parag Vaid [3], Kriti Srivastava [4]
Department of Computer Engineering
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
{[1] sanghavisidd, [2] rathodpalash3, [3] paragvaid3496}@gmail.com, [4] kriti.srivastava@djsce.ac.in

***Abstract*—Brain research has been highlighted as an area of interest in recent years. Study in this domain has the potential to impact various fields ranging from study of brain related medical conditions to neural control of devices to aid the handicapped. Our study aims at classifying the sleep stages which will help to gain a better understanding and diagnosis of sleep disorders such as sleep apnea, insomnia, etc. The classification of brain signals using machine learning approaches is a powerful tool in areas of research. However, deep learning architectures are seldom used for this task despite achieving state of the art accuracies. Therefore, we present a comparative study of two deep learning architectures used to classify the sleep stages based on electroencephalography (EEG) signals.**

Keywords—Deep Learning, EEG, PhysioNet, Sleep Stages, Spectrograms, Time Series

INTRODUCTION

Sleep is the primary function of the brain that has an essential role in every individual's performance, learning capabilities, and physical movement. On average humans spend around 7 to 9 hours sleeping i.e. one-third of their day. Human sleep may be a regular state of rest for the body during which the eyes aren't only usually closed, but even have several nervous centers being inactive; hence, rendering the person either partially or completely unconscious and making the brain a simpler network.

Sleep diseases such as insomnia and obstructive sleep apnea have a negative impact on the quality of human life. Therefore, the need to distinguish human sleep stages is very important for the diagnosis and treatment of sleep disorders such as sleep apnea, insomnia, and narcolepsy. Studies of sleep assist doctors to diagnose sleep disorders and supply the baseline for appropriate follow-up. The clinical sleep study design is based on polysomnography (PSG) in which different types of biological signals are acquired while the patient is asleep, including electroencephalography (EEG) for observing brain activity, electrooculogram (EOG) for eye movements, and electromyogram (EMG) to measure muscle tone .

The measurements are to classify sleep score, i.e., classify the sleep stages the patient goes through during the study and assess the existence of any dysfunction. Manually scoring sleep stages may be a tedious task requiring sleep experts to visually inspect PSG data recorded during the entire sleep study. With the increasing accessibility of EEG signals, there is a growing interest in sleep quantification based on EEG alone. The EEG signal many times presents significant bursts of rhythmic components. Thus, there has been a considerable effort over the past years to develop machine learning methods for automatic sleep scoring and the development of efficient sleep stage identification techniques using single-channel EEG signals would be very beneficial to the analysis, diagnosis, and treatment of sleep disorders.

The American Academy of Sleep Medicine (AASM) segmented sleep into five stages :

- W (wakefulness): This stage consists of alpha (8–12 Hz) rhythm, high-amplitude muscle contractions and movement artifacts on EMG and eye blinking on EOG, which may even be captured in low-frequency EEG (0.5-2 Hz).
- Nl (Non-REM 1): In this stage alpha (8–12 Hz) rhythm is attenuated and replaced by mixed frequency theta signal (4–7 Hz), decrease in muscle tone, and slow eye movements.
- N2 (Non-REM 2): It consists of K-complexes (negative peak followed by a positive complex and a final negative voltage) within the < 1.5 Hz range and sleeps spindles (burst of oscillatory waves) at sigma (12–15 Hz) band.
- N3 (Non-REM 3): This is when slow-wave activity exists (0.5-3 Hz), eye movements are unusual and EMG tone is low.
- R (REM): The existence of rapid eye movements clearly visible in EOG, relatively low-amplitude and mixed-frequency activity in EEG, and presents rock bottom muscular tonus on EMG.

LITERATURE SURVEY

To gain an in-depth knowledge about brain waves and studies conducted in the same field, we went through various research papers. The first thing that needed to be understood was the nature of brain waves and methods by which we can acquire them for further use. On studying

the paper **[Introduction to the identification of brain waves based on their frequency],** we became familiar with the concept of Electroencephalogram (EEG). As the communication between brain cells is in the form of electric impulses, EEG is the test which is used to evaluate the electrical activity in the brain. Each brain wave has a different frequency and EEG signals are evaluated based on these frequencies. The distribution of brain waves by frequency of band waves is shown in the table below.

Table. 1. Various classes of EEG signal bands

Name of the band wave	**Frequency of the band wave (Hz)**
Alpha α	8-13
Beta β	13 - 30
Delta δ	0.5 - 4
Gamma γ	> 30
Theta θ	4 - 8

Thus, we came to the conclusion of using EEG signals for our study since they are multi-dimensional, and they directly measure the neural activity.

The next task after acquiring EEG signals was to find an appropriate way to preprocess these signals and select a deep learning architecture for the same. **[Deep learning for Electroencephalogram (EEG) Classification tasks: a review]** gave an in-depth comparison of various deep learning architectures as well as various input formulations. These input formulations included Signal Values, Calculated Features, and Images. The general deep learning strategies included Convolutional Neural Network (CNN), Hybrid-CNN, Recurrent Neural Network (RNN), Multi-Layer Perceptron (MLP), etc. After a deep-rooted analysis of the various architectures and input formulations provided by the paper, we decided to use two input formulations, namely, images and signal values. Images were used in the form of a spectrogram while the signal values were used in the form of a time series data. 2-Dimensional CNN architecture was used for spectrograms whereas 1-Dimensional CNN was used for the time series data.

Spectrogram is a visual representation of "loudness" or the strength of the signal over time at various frequencies in a particular waveform. [**Spectrum Analysis of EEG signals using CNN to model a patient's Consciousness Level based on Anesthesiologists' experience**] proposes a novel approach of using spectrogram images produced from EEG signals which can be used as input for CNN. Image input layer receives a spectrogram of EEG signals from training images and transforms this image into specific dimensions so that it can be delivered to the convolution layer correctly. The proposed method achieved an impressive accuracy of 93.50%. However, the training time was large for which we considered a different input formulation, namely, signal values.

For 1D signals such as EEG, ECG, EMG, etc., 1D CNNs have recently been proposed and have achieved state-of-the-art performance levels in several applications. In EEG data analyses, isolation of channel specific activity and localization of the activity in time is done by convolving filters or kernels with EEG data in 1D CNN. Studies have shown that for certain applications 1D CNNs are advantageous and thus preferable to their 2D counterparts in dealing with 1D signals. Thus, after studying various research papers, the advantages and the disadvantages of the methods proposed in these papers, we propose a comparative study of two different deep learning architectures for the classification of sleep stages based on EEG signals.

IMPLEMENTATION

(A) Dataset

The dataset used in this project is the Physionet Sleep-EDF dataset which is a publicly available dataset from the PhysioNet database. Its associated hypnogram files contain sleep patterns like each subject. All hypnograms were manually scored by well-trained technicians according to Rechtschaffen and Kales manual. The hypnograms represent an event for segments (epochs) of 30 seconds.

The Sleep-EDF dataset includes whole-night polysomnograms (PSGs) sleep recordings at the sampling rate of 100 Hz. Each record contains EEG readings 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 blood heat.

Each stage was considered to belong to a special class (stage). The classes include wake (W), rapid eye movement (REM), N1, N2, N3, N4, and M (movement time). According to the American Academy of Sleep Medicine (AASM) standard, we integrated the stages of N3 and N4 in one class named N3 and excluded the M (movement time) stage to have five sleep stages: W, N1, N2, N3, REM corresponding to classes 0, 1, 2, 3 and 4 respectively. Stages 1 and a couple of - 3 are the sunshine sleep time during which stage N1 is the lightest stage and features a short period of time. The time taken by stage N2 - N3 is greater than the stage N1, as it constitutes approximately 40-60% of total sleep time . Stage N3 is called deep sleep and the REM (Stage 5) is known as the dreaming stage taking 90 - 120 minutes per night.

(B) Dataset Split

This step involves randomly dividing the available set of observations into three parts, a training set, a validation set and a testing set. Training dataset is the sample of data used to fit the model. The actual dataset that we use to train the model (weights and biases in the case of Neural Network). The model sees and learns from this data. Validation dataset is the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. Test dataset is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. The Test dataset provides the gold standard used to evaluate the model. It is only used once a model is completely trained(using the train and validation sets).

In our case 70% of the dataset is allocated to the training set, 20% to the validation set and the remaining 10% is allocated to the testing dataset. This split is done for a dataset with 54,587 samples; the final distribution is given below.

Table. 2. Train, Validation and Test Split

Train Set	40967
Validation Set	10890
Test Set	2730

The data for each subject consists of a time series data collected from 7 channels out of which there are 2 EEG channels from Fpz-Cz and Pz-Oz electrode locations, for single-channel EEG data we select the Fpz-Cz channel. The recording on an average for each subject is of 22 hours and the sampling frequency of the data is 100Hz meaning at every second 100 readings are taken. For each subject, the total number of data points becomes around 7,920,000 data points (22 x 60 x 60 x 100).

The window for classification is kept as 30 seconds so the data for each subject are divided into samples of 30 seconds. Thus, the average number of samples for each subject becomes 2,640 where each sample consists of data points for 30 seconds i.e. 3000 data points (30 x 100).

(C) Dataset Preprocessing

2D CNN:

From the input received in the previous stage, an image of the spectrogram is generated and fed into the network. Each sample which is a time-series data of 30 seconds sampled at 100 Hz is used to create a spectrogram for that sample. Digitally sampled data, within the time domain, is split into chunks, which usually overlap, and Fourier transformation is applied to calculate the magnitude of the frequency spectrum for every chunk. Each and every chunk then matches up to a vertical line within the image; a representation of measurement of magnitude versus frequency for a selected moment in time (the midpoint of the chunk). These spectrums or time plots are then kept side by side to form the image or a three-dimensional surface .

An image of a spectrogram generated from one of the samples is shown below.

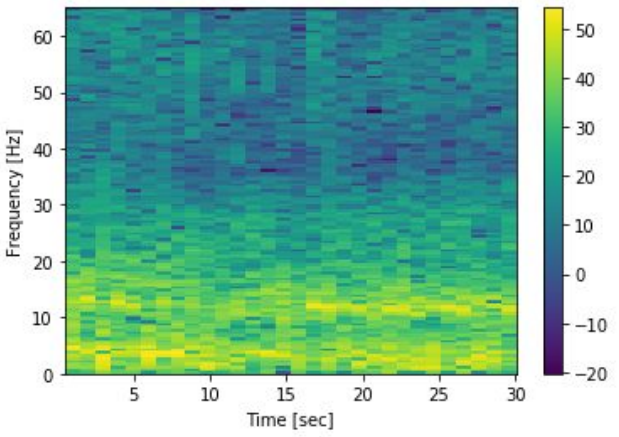

Fig. 1. Spectrogram of the 30s EEG signal sample

The acquired image data is usually messy and comes from different sources. To feed them to the ML model (or neural network), it needs to be standardized and cleaned up. More often than not, preprocessing is used to conduct steps that reduce the complexity and increase the accuracy of the applied algorithm. It is not possible to write a unique algorithm for each of the conditions in which an image is taken, thus, when an image is acquired, It is converted into a form that allows a general algorithm to solve it.

The image preprocessing pipeline on the Physionet Sleep dataset involves the following steps:

1. Image normalization is an important step that ensures that each input parameter (i.e, a pixel in this case) has a similar data distribution. This makes convergence faster while training the neural network.

2. Each image is converted from uint8 to float32 format. And then the pixel intensities of each image is normalized by dividing it by 255.0 so that it lies between the range [0.0, 1.0].

3. This data generated is fed into the neural network in batches of 16.

1D CNN:

A 1D CNN is very effective when you expect to derive interesting features from shorter (fixed-length) segments of the overall data set. This applies well to the analysis of time sequences. It also applies to the analysis of any kind of signal data over a fixed-length period (such as EEG signals).

The data obtained from the loading dataset step is a time series data of 30 seconds for multiple samples which can be directly fed into our 1D CNN. However, it is necessary

to perform normalization for 1D CNN as well because it improves the numerical stability of the model and often reduces training time. The data preprocessing pipeline on the Physionet Sleep dataset involves the following steps:

1. The normalization for each data in a sample is performed in the following way:

   $$Xi = (X - Xmin) / (Xmax - Xmin) \quad (1)$$

   where Xi is the normalised value of the data point, X is the actual value of that data point, Xmin is the minimum value of a datapoint and Xmax is the maximum value of a datapoint in the selected time series.

2. Since a neural network takes input data in batches, we add a superfluous channel dimension to the training, validation and testing data points. Hence the new shape of the instances now correspond to (no_of_images, time_steps, features). Here, the features dimension has a value of 1 as the instances in the PhysioNet Sleep dataset have been taken from a single EEG channel.

3. We then One Hot Encode the Y labels to form a tensor of shape (batch_size, 5).

4. This preprocessed data is then fed as input into the 1D CNN in mini-batches of 16.

(D) Model Architecture

During the last decade, Convolutional Neural Networks (CNNs) have become the de facto standard for various Computer Vision and Machine Learning operations. Among different kinds of neural networks, Convolutional neural networks (CNNs) is one of the main categories to do image recognition, images classifications. Object detections, face recognition, etc., are some of the areas where CNNs are widely used.

2D CNN:

Since a spectrogram image is the visual representation of the frequency spectrum of a signal, a deep learning method like 2D CNN is employed to perform feature extraction and classification from spectrogram images. 2D CNN image classifications take an input image, process it, and classify it under certain categories in our case (W, N1, N2, N3, R). In a CNN model, each input image will undergo a series of convolution layers with filters (Kernels), Convolution is the first layer to extract features from an input image. Convolution preserves the connection between pixels by learning image features using small squares of input data. It is a mathematical process that takes two inputs such as an image matrix and a filter or kernel. An image matrix of dimensions ($h \times w \times d$) and a filter of dimension ($h_f \times w_f \times d_f$) outputs a volume dimension ($h\text{-}h_f+1$) × ($w\text{-}w_f+1$) × 1.

Then after the convolution pooling operation is performed, the Pooling layers section would scale back the number of parameters when the images are too large which reduces the dimensionality of each map but retains important information. The layer we call the Fully Connected layer, we flatten our matrix into vectors and feed it into a fully connected layer like a neural network. The feature map matrix is going to be converted as a vector. Using the fully connected layers, we combined these features together to make a model. Finally, we have an activation function such as softmax to classify the outputs as W, N1, N2, N3, or R.

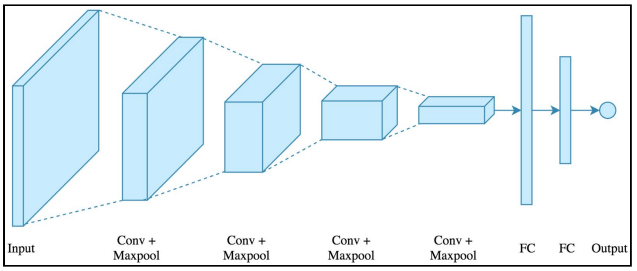

Fig. 2. 2D CNN Architecture

The proposed 2D CNN is modelled after VGG16 architecture. It takes an input tensor of the shape (batch_size, height, width, no_of_channels) and outputs a tensor of shape (batch_size, 5). Here the input images are the RGB spectrograms of shape (36, 54, 3) of each datapoint in the dataset built during the preprocessing phase. Our 2D CNN model for sleep stages classification is demonstrated in the table below.

Table. 3. Proposed 2D CNN Architecture

LAYER	**OUTPUT**	**FUNCTION**
INPUT	36, 54, 3	
CONV_1	34, 52, 32	2D Convolution
CONV_2	34, 52, 64	2D Convolution
POOL_1	17, 26, 64	2D MaxPooling
CONV_3	17, 26, 128	2D Convolution
POOL_2	8, 13, 128	2D MaxPooling
CONV_4	8, 13, 256	2D Convolution
...	...	...
POOL_3	4, 6, 256	2D MaxPooling
CONV_7	4, 6, 512	2D Convolution
...	...	...
POOL_4	2, 3, 512	2D MaxPooling
FLATTEN	3072	Flatten
DENSE_1	4096	Dense

DENSE_2	5	Dense

1D CNN:
Deep 2D CNNs with many hidden layers and millions of parameters have the ability to learn complex objects and patterns providing that they can be trained on a massive size visual database with ground-truth labels. With proper training, this unique ability makes them the primary tool for various engineering applications for 2D signals such as images and video frames. Yet, this may not be a viable option in numerous applications over 1D signals like EEG, ECG, EMG, etc. especially when the training data is scarce or application-specific. To address this issue, 1D CNNs have recently been proposed and immediately achieved state-of-the-art performance levels in several applications such as personalized biomedical data classification and early diagnosis, structural health monitoring, anomaly detection and identification in power electronics and motor-fault detection.

Convolution in the time domain is an extension of the dot product, in which the dot product is computed repeatedly over time. Convolution can also be performed over space, but for EEG analyses, convolution over time is used. In EEG data analyses, convolution is used to isolate channel specific activity and to localize that specific activity in

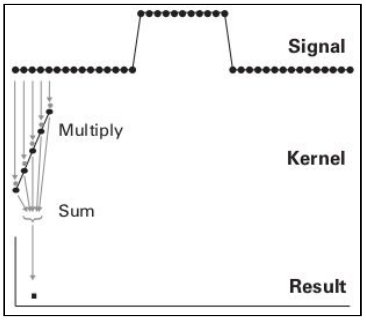

Fig. 3. Convolution in time domain

time. This is done by convolving filters or kernels with EEG data in the 1D CNN. As the convolution kernel is dragged along the EEG data (the convolution signal), it reveals when and to what extent the EEG data contain features that look like that filter/kernel. When convolution is repeated on the same EEG data using 1D kernels of different sizes, a different amplitude-time representation can be formed.

Imagine a time series of length n and width k. The length is the number of timesteps, and the width is the number of variables in a multivariate time series. For example, for EEG it is the number of channels (electrodes on the head of a person), and for a weather time series it can be such variables as temperature, pressure, humidity etc. The convolution kernels always have the same width as the time series, while their length can be varied. This way, the kernel moves in one direction from the beginning of a time series towards its end, performing convolution. It does not move to the left or to the right as it does when the usual 2-D convolution is applied to images.

The elements of the kernel get multiplied by the corresponding elements of the time series that they cover at a given point. Then the results of the multiplication are added together and a nonlinear activation function is applied to the value. The resulting value becomes an element of a new "filtered" univariate time series, and then the kernel moves forward along the time series to produce the next value. The number of new "filtered" time series is the same as the number of convolution kernels. Depending on the length of the kernel, different aspects, properties, "features" of the initial time series get captured in each of the new filtered series.

The next step is to apply global max-pooling to each of the filtered time series vectors: the largest value is taken from each vector. A new vector is formed from these values, and this vector of maximums is the final feature vector that can be used as an input to a regular fully connected layer.

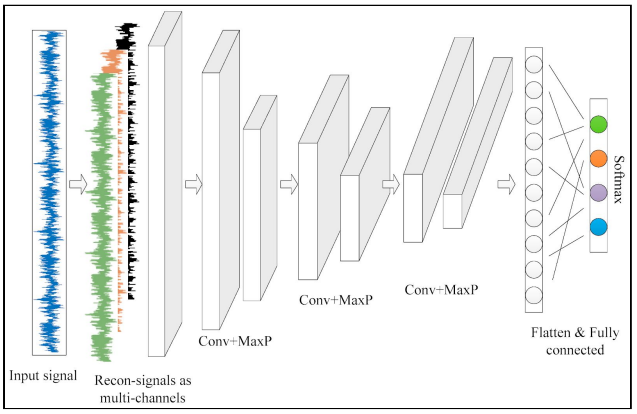

Fig. 4. 1D CNN Architecture

The proposed 1D CNN takes an input tensor of the shape (batch_size, time_steps, features) and outputs a tensor of shape (batch_size, 5). Here the input is the EEG wave vector of shape (batch_size, 3000, 1) of each datapoint in the dataset built during the preprocessing phase. The summary of the model is mentioned below. Our 1D CNN model for sleep stages classification is demonstrated in the table below.

Table. 4. Proposed 1D CNN Architecture

LAYER	**OUTPUT**	**FUNCTION**
INPUT	3000, 1	
CONV_1	3000, 16	1D Convolution
POOL_1	1500, 16	1D MaxPooling
CONV_2	1500, 32	1D Convolution
POOL_2	750, 32	1D MaxPooling

CONV_3	750, 64	1D Convolution
POOL_3	375, 64	1D MaxPooling
CONV_4	375, 128	1D Convolution
POOL_3	187, 128	1D MaxPooling
FLATTEN	23936	Flatten
DENSE_1	512	Dense
DENSE_2	5	Dense

(E) Learning Methodology and Parameter Tuning

The aforementioned neural network models are trained using Mini-batch Gradient Descent Algorithm. It is a first-order optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our models. Parameters refer to weights w and biases b in our neural network. Optimization refers to the task of minimizing/maximizing an objective function f(x) parameterized by x. In machine learning terminology, it is the task of minimizing the cost/loss/objective function J(w) parameterized by the model's parameters w, b ∈ R^d.

A Loss Function J(w) tells us "how good" our model is at making predictions for a given set of parameters. The cost function has its own curve and its own gradients. The slope of this curve tells us how to update our parameters to make the model more accurate. Here we use Categorical Cross Entropy Error as our loss function.

The weights and biases are updated using the Backpropagation Algorithm. The use of Mini-batch Gradient Descent in the neural network setting is motivated by the high cost of running back propagation over the full training set. It can overcome this cost and lead to fast convergence. Generally each parameter update in Mini-batch Gradient Descent is computed w.r.t a few training examples or a mini-batch as opposed to a single example. The reason for this is twofold: first this reduces the variance in the parameter update and can lead to more stable convergence, second this allows the computation to take advantage of highly optimized matrix operations that should be used in a well vectorized computation of the cost and gradient. Hence we use a mini-batch size of 16 (spectrogram images for 2D CNN and EEG vectors for 1D CNN) for 15 epochs.

We use the Adam Optimizer with an initial learning rate (alpha) of 0.0001 along with Mini-batch Gradient Descent. Adaptive Moment Estimation (Adam) is an adaptive learning rate method that computes adaptive learning rates for each parameter. Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with Momentum

RESULTS AND DISCUSSION

(A) 2D CNN

Table. 5. Accuracy and Loss Metrics for our 2D CNN

Dataset	**Loss**	**Accuracy (%)**
Training	0.2122	92.13
Validation	0.2292	92.10
Test	0.2477	91.94

The testing data doesn't overlap with the training data. As can be seen in the table above, the training and testing accuracy values are close to each other. It has also achieved a validation accuracy of 92.1%. Hence, it implies that the proposed model has good generalization ability.

Table. 6. Classification Report for our 2D CNN

	Precision	**Recall**	**f1 - score**	**Support**
Class W	0.97	1.00	0.98	1819
Class 1	0.60	0.29	0.39	65
Class 2	0.88	0.91	0.89	490
Class 3/4	0.85	0.75	0.80	157
Class R	0.72	0.73	0.72	199
Micro avg	0.92	0.92	0.92	2730
Macro avg	0.80	0.74	0.76	2730

From the table above we can see an f1-score of 0.98 was obtained for the wake-sleep stage. Since the bulk of data belong to this stage, the learning model showed a trend toward learning the data in this stage. The lowest f1-score 0.39 was observed in class 1. The amount of data in class 1 is less than others so that the proposed model had difficulty in learning this stage.

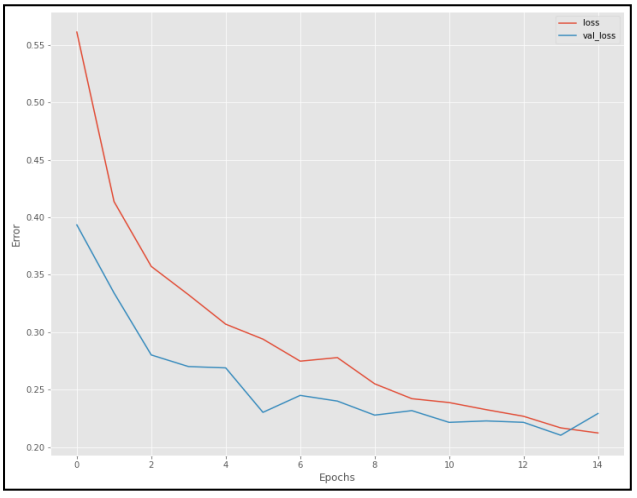

Fig. 3. Graph of Cross Entropy Error vs. No. of Epochs for our 2D CNN

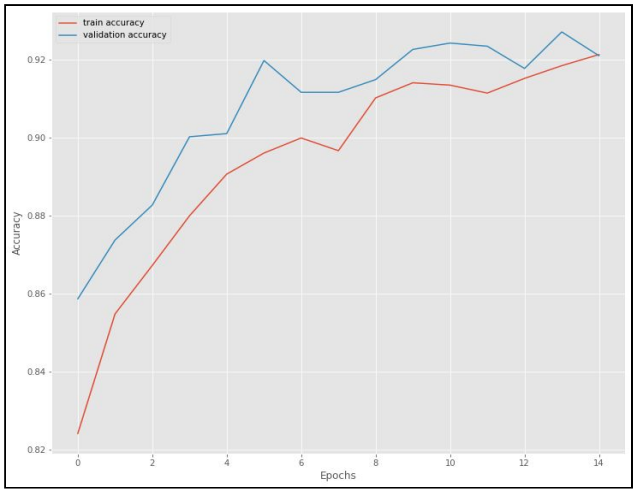

Fig. 4. Graph of Accuracy vs. No. of Epochs for our 2D CNN

As can be seen from the performance graphs above, no overfitting problem occurred for all of the classes. The training and validation accuracy curves represent positive learning.

(B) 1D CNN

Table. 7. Accuracy and Loss Metrics for our 1D CNN

Dataset	**Loss**	**Accuracy (%)**
Training	0.7795	92.91
Validation	0.7567	92.94
Test	0.7563	92.38

The testing data doesn't overlap with the training data. As can be seen in the table above, the training and testing accuracy values are close to each other. It has also achieved a validation accuracy of 92.1%. Hence, it implies that the proposed model has good generalization ability.

Table. 8. Classification Report for our 1D CNN

	Precision	**Recall**	**f1 - score**	**Support**
Class W	0.98	0.98	0.98	1819
Class 1	0.44	0.28	0.34	65
Class 2	0.85	0.86	0.86	490
Class 3/4	0.82	0.88	0.85	157
Class R	0.74	0.77	0.76	199
Micro avg	0.92	0.92	0.92	2730
Macro avg	0.77	0.76	0.76	2730

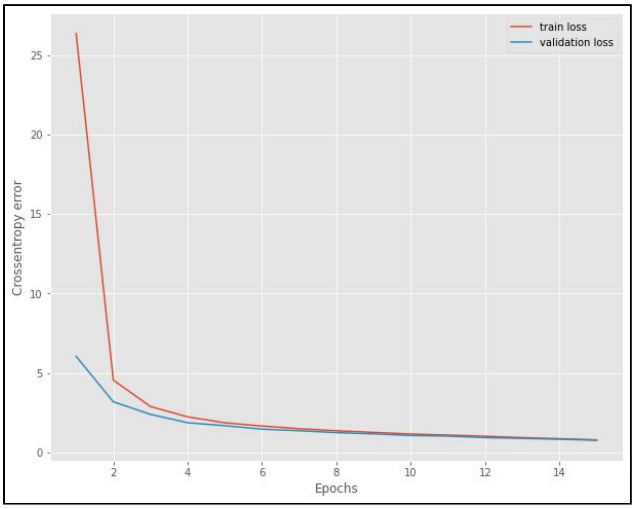

Fig. 5. Graph of Cross Entropy Error vs. No. of Epochs for our 1D CNN

Studies have shown that for certain applications 1D CNNs are advantageous and thus preferable to their 2D counterparts in dealing with 1D signals due to the following reasons:

- Rather than matrix operations, FP and BP in 1D CNNs require simple array operations. This means that the computational complexity of 1D CNNs is significantly lower than 2D CNNs.
- Recent studies show that 1D CNNs with relatively shallow architectures (i.e. small number of hidden layers and neurons) are able to learn challenging tasks involving 1D signals. On the other hand, 2D CNNs usually require deeper architectures to handle such tasks. Obviously, networks with shallow architectures are much easier to train and implement.

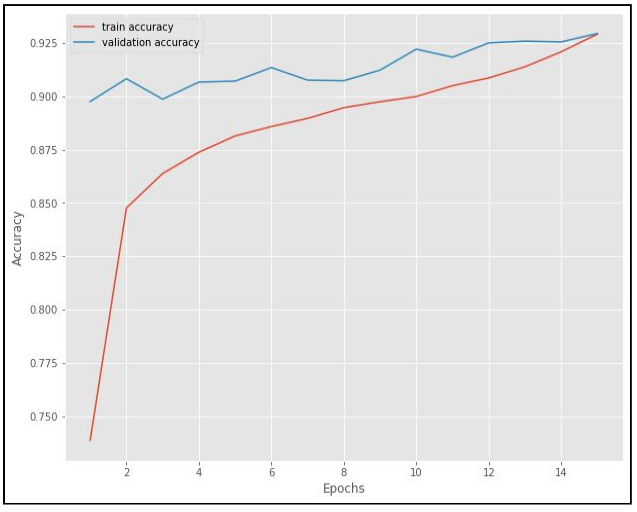

Fig. 6. Graph of Accuracy vs. No. of Epochs for our 1D CNN

- Usually, training deep 2D CNNs requires special hardware setup (e.g. Cloud computing or GPU farms). On the other hand, any CPU implementation over a standard computer is feasible and relatively fast for training compact

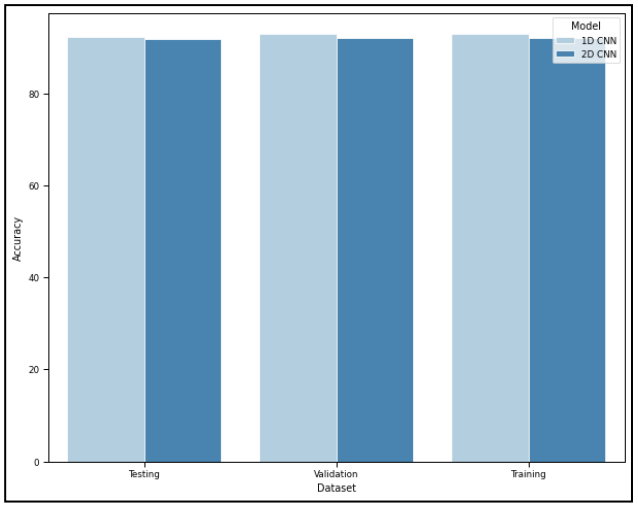

Fig. 7. Results comparison for 2D CNN and 1D CNN

1D CNNs with few hidden layers (e.g. 2 or less) and neurons (e.g. < 50).

- Due to their low computational requirements, compact 1D CNNs are well-suited for real-time and low-cost applications especially on mobile or hand-held devices.

As evident from Fig. 7., both the models are performing extremely well and have generalized well on the test dataset. However 1D CNN seems to be performing better than 2D CNN in all the test cases.

CONCLUSION

Our study successfully classifies the sleep stages based on EEG signals which is pivotal in quantitative analysis of polysomnographic recordings. The deep learning architectures used for the task of automating the classification show promising results. Observations show that 1D CNN is slightly more accurate than 2D CNN. The reason might be due to all the pre-processing involved in the input formulation of 2D CNN. 1D CNN does not involve much pre-processing as it directly works on the time series data (signal values). However, 2D CNNs can be useful in specific applications as spectrograms preserve all three features of the wave: time, frequency, and power. An improvement in the overall accuracy can be achieved by using a dataset which has a more even distribution of the data points amongst its class labels. Sourcing the data from different laboratories, pathologies and increasing the number of subjects can lead to a more diverse dataset which can be useful for better training.

Future Scope

Successful classification of sleep stages based on EEG signals with the help of deep learning techniques indicates that this can be extended to other fields of neural engineering as well as commercial applications. Two applications where this technique can prove to be of utmost help are:

(A) Autism Spectrum Disorder Detection

Autism Spectrum Disorder (ASD) is a syndrome that adversely affects a child where the behavioral symptoms start to appear during the first year of life. This early childhood onset includes symptoms such as lack in social interaction and very slow language skills development as stated by researchers. A continuous character and behavioral assessment are conducted by specialists to detect autistic presence in a child. A documented analysis done by pediatrics stated that an autistic child at approximately 24 months, is still unable to produce two meaningful words that do not involve imitating and repeating. Despite so much research being conducted, the exact factors to why this disorder occurs remain unanswered. As of why this atypical behavior is very difficult to detect is maybe due to the barely noticeable changes of the primary neural impairment itself.

The relation of ASD with EEG signal is that there is a significant decline of EEG complexity perceived in an autistic child. The current diagnostic method for ASD is time consuming. By applying deep learning architectures to evaluate the patterns of EEG signals of autistic and normal kids, a model can be created for the early detection of autism. Publicly available datasets of EEG signals of an autistic patient are limited. Perhaps, working towards building a more extensive dataset can help in the creation of a much accurate model for detecting autism.

(B) Brain Password

Biometrics is the measuring and statistical analysis of people's physical and behavioral attributes. This technology can be used to define an individual's unique identity, often employed for security purposes. The traditional biometric traits are face recognition, retina or iris scanning, fingerprints, hand geometry, palm print signature, keystroke entry pattern, and voice recognition. However, many of these traits can be forged or stolen. Fingertips, for example, can be damaged by an injury or can be forged by a gummy finger. Disguises can trick face recognition applications. Brain waves or Electroencephalogram (EEG) are the electrical activity of an individual's brain that is unique and cannot be tampered with. Hence, EEG is proposed as an alternative or an additional way of securing biometric applications.

EEG signals are gathered from electrodes that are placed in several locations on the scalp. Because everyone's brain is structured differently each EEG signal is unique for each person. EGG uniqueness makes the biometric un-forgeable or un-duplicable. The current EEG-based biometric procedures are based on eye blinks, visual stimuli, and rest states. Data can be collected via hardware devices and after preprocessing which can involve spectrogram computation, feature extraction, feature selection, etc., the pre-processed data can be trained on classifiers such as SVM or ANN for pattern recognition.

ACKNOWLEDGMENT

The authors are thankful to Mrs. Kriti Srivastava, Assistant Professor, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India, for providing necessary resources and infrastructure to conduct this project work and for her encouragement.

REFERENCES