SpectroTemporalNet: Automated Sleep Stage Scoring with Stacked Generalization

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

***Abstract*—Polysomnography (PSG) has become a pivotal diagnostic tool in sleep medicine. The scoring obtained on the resultant data from sources including Electroencephalograms (EEG) for these studies are interpreted by Sleep Disorder Specialists (SDS) to identify various sleeping disorders. However this process is tedious and requires highly skilled experts. With the recent advent of Deep Learning techniques, neural networks can be utilized to assist in finding such irregularities in sleep data. This paper proposes a Deep Learning based ensemble model called SpectroTemporalNet for automating the process of sleep stage scoring from single-channel raw EEG data. SpectroTemporalNet uses Stacked Generalization to achieve a better annotation performance as compared to that of existing literature with an overall classification accuracy of 94.31% and a sensitivity of 94% on the test set. The proposed model utilizes both the spectral and the temporal features of EEG data simultaneously thereby improving automated sleep stage scoring.**

***Index Terms*—Convolutional Neural Networks, Electroencephalogram, PhysioNet, Sleep Stages, Spectrograms, Time Series, WaveNet**

I. Introduction

One of the most significant functions of the brain is sleeping. It impacts a person's physical as well as mental learning ability and performance. In an individual's day-to-day life, 7–9 hours are spent sleeping, which corresponds to almost one-third of the day. The American Academy of Sleep Medicine (AASM) has segmented human sleep into 5 stages [1]:

- **Wakefulness (W):** This stage consists of high frequency and low amplitude alpha signals (8-–12 Hz). This stage can be considered as the time just before and after sleep, it also consists of the short-lived awakenings.
- **Non-Rapid Eye Movement (Non-REM):** This stage is also called "light sleep" and consists of low frequency and high amplitude signals. The normal process of falling asleep begins with this stage. A person's brain waves slow down considerably and his/her eyes don't move back and forth rapidly during Non-REM, in contrast to later stages of sleep. It is further classified into three stages: Non-REM 1 (N1), Non-REM 2 (N2) and Non-REM 3 (N3).
- **Rapid Eye Movement (REM):** In this stage, a person's eyes do not send any optical signal information to the brain. Instead his/her eyes just keep moving in different directions. Most dreaming occurs during REM sleep.

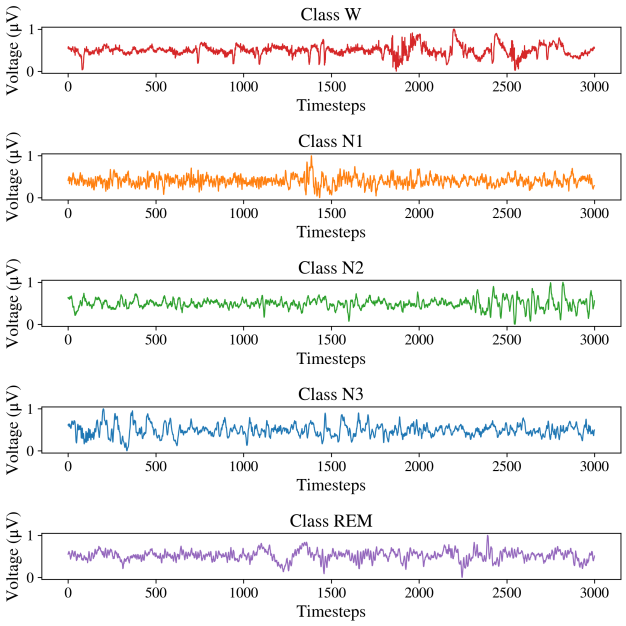

Fig. 1. Normalized 30 s EEG signals of AASM Sleep Stages from the PhysioNet Sleep-EDF Database sampled at 100 Hz.

A person without a sleep disorder normally goes through 4–6 sleep cycles switching between Non-REM and REM sleep, each for about 90 minutes. This sleep process gets disturbed in people having sleep disorders. These sleep disorders can mainly affect one's mood, energy level, and overall health. The symptoms for sleep disorders often develop gradually over a period of time making them difficult to diagnose [2]. In order to enable an effective treatment, accurate diagnosis is necessary.

Typically, a sleep study is the first step towards the diagnosis of sleep disorders. These studies are utilized to register the body's shift between different stages of sleep and identify abnormalities or disturbances related to sleep disorders. One of the most widely used sleep studies is called Polysomnography (PSG). It is a non-invasive, multi-parametric test used to collect physiologic parameters related to sleep and used as a diagnostic tool in sleep medicine. A Polysomnogram measures brain activity (EEG), eye movements (EOG), muscle

activity (EMG), heart rhythm (ECG), pulse oximetry, airflow and respiratory effort to evaluate underlying causes of sleep disruptions. Valuable information including a detailed picture of a person's unique sleep patterns can be extrapolated from PSG data to relate it to sleep disorders such as Obstructive Sleep Apnea (OSA), Periodic Limb Movement Disorder (PLMD), Narcolepsy, REM Sleep Behavior Disorder, Unexplained Chronic Insomnia.

For the purpose of this study, the EEG data obtained from PSG datasets of the PhysioNet database are used [3]. The EEG electrodes placed on a person's scalp provide a graph of the brain activity in terms of voltage (μV) and time (s). After the process of Polysomnography is complete, a sleep specialist interprets, analyzes and scores this graph data by reviewing it in "epochs" of 30 s [4]. The specialist then "scores" the EEG data into different stages of sleep, namely W, N1, N2, N3 and REM, that a patient cycles throughout the process and checks for the existence of any abnormality.

Manual classification of these sleep stages from raw PSG data is a tedious, error-prone and subjective process which takes up to two weeks. It also requires highly trained professionals to visually observe the PSG recording of each patient. Due to these reasons, there has been a growing interest in this field to automate the process of scoring sleep stages efficiently from fewer channel EEG data using Machine Learning techniques.

Recently, Convolutional Neural Networks (CNNs/ConvNets) have achieved remarkable performances on tasks such as image classification [5, 6] and sequence modeling [7]. Deep 2-dimensional (2D) CNNs with millions of parameters are able to learn complex patterns and objects from visual datasets consisting of images and videos. However the datasets for applications involving 1-dimensional (1D) signals such as EEG, ECG, EMG, etc. are scarce or application-specific. Recently, 1D CNNs have been proposed to handle 1D data and have demonstrated state-of-the-art performance levels in several areas like personalized biomedical data classification, human activity recognition, structural health monitoring, anomaly detection, etc. [8].

Deep CNNs have shown to successfully classify sleep stages in current implementations [9]. Hence, the aim of this paper is to discuss and evaluate the hypothesis of applying CNNs for the task of automated sleep stage scoring using EEG data. The task of Sleep Stage scoring is treated as a visual classification as well as time series classification problem allowing us to achieve better performance than existing approaches. To this end, this paper proposes a novel Deep Learning model architecture called SpectroTemporalNet. It consists of a stacked ensemble of a 2D CNN called SpectralNet and a 1D CNN known as TemporalNet. This enables it to take advantage of the spectral (in terms of spectrograms) as well as the temporal (in terms of time series) nature of EEG data. SpectroTemporalNet uses Stacked Generalization to obtain a classification accuracy of 94.31% and a sensitivity of 94% on the test set.

The contributions put forth by this approach are as follows:

- This paper proposes a stacked ensemble model [10] called SpectroTemporalNet which utilizes two sub-models to extract features from EEG data. This architecture ensures more accurate classification results and minimizes the generalization error.
- Two sub-models are proposed for the stacked ensemble. The first model captures the temporal aspects from preprocessed EEG signals, while the second model encapsulates the spatial aspects of their corresponding RGB spectrograms, enhancing the extraction of relevant features.
- The temporal network of the ensemble is based on the WaveNet architecture [11]. Hence it can capture the long term dependencies prevalent in data involving long sequences such as EEG signals. This attribute helps the ensemble model in overcoming the issue faced by traditional Long Short Term Memory (LSTM) networks and Gated Recurrent units (GRUs) wherein they cannot efficiently handle long sequences, both in terms of computation and performance.

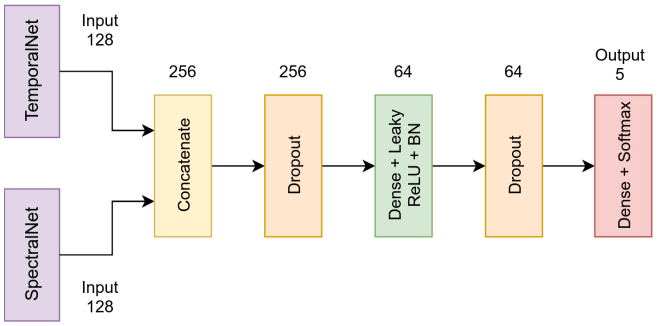

Fig. 2. Block diagram of the SpectroTemporalNet architecture

II. Related Work

A person cycles through the five sleep stages during a typical night and all these stages are linked to specific brain signals and neural activity. The nature of brain waves and methods for their acquisition have been identified in [12]. According to authors in [12], 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 as follows: Alpha (α) waves are in the frequency range of 8 – 13 Hz, Beta (β) waves between 13 – 30 Hz, Delta (δ) waves between 0.5 – 4 Hz, Gamma (γ) waves >30 Hz and Theta (θ) waves between 4 – 8 Hz.

EEG signals are noisy which makes it necessary to select appropriate preprocessing techniques and deep learning architectures. In [13], the authors give an in-depth comparison of various deep learning architectures as well as various input formulations. Liu et al. [14] proposed a novel approach of using spectrogram images produced from EEG signals which can be used as input for a CNN. The proposed method achieved an impressive accuracy of 93.50%. In [15], Nilufar et al. used spectrogram analysis to select features from speech or music

and discriminate them from each other using multiple kernel learning and achieved accuracies close to 98%. Apart from Machine Learning approaches, Deep Learning architectures have also been used for spectrogram analysis. Ruffini et al. [16] attempted to diagnose Rapid Eye Movement Behavior Disorder by classifying the stacked multi-channel EEG spectrograms from idiopathic patients and healthy controls using a 5 layer deep CNN. They also compared the performance of this CNN model and a Recurrent Neural Network (RNN) with stacked LSTM and GRU cells. These models reached a classification accuracy of 80% ($\pm$1%) which was further improved by using EEG signals acquired from the best EEG channel.

Cui et al. [17] demonstrated the performance of CNNs in classifying sleep stages by using a five layer CNN architecture combined with a fine-grained segment in multiscale entropy. They trained their model on the ISRUC-Sleep public dataset achieving an average accuracy of 92.2%. Ozal et al. [18] implemented a nineteen layer deep CNN for classifying five classes on the PhysioNet Sleep European Data Format (EDF) [3], [19] public database, obtained an accuracy of 90.83%. The study presented by Aboalayon et al. in [20] also used the Sleep-EDF dataset to train various models on it including Decision Trees, Support Vector Machines (SVM), Feedforward Neural Networks (NN), and k-Nearest Neighbors (KNN) achieving accuracies of 93.29%, 92.37%, 91.70% and 89.38% respectively. Michielli et al. [21] make use of cascaded LSTM blocks based on RNNs and attained an accuracy of 86.7% on the Sleep-EDF dataset. Supratáka et al. [22] proposed a combination of CNN and LSTM for classification of sleep stages, achieving an accuracy of 82.0%.

The survey presented by Kiranyaz et al. [7] shows that the recently propsed 1D CNNs for applications involving 1D signals such as EEG, ECG, EMG, etc.have performed remarkably well. Such studies have shown that for certain applications, 1D CNNs are advantageous and thus preferable to their 2D counterparts in dealing with 1D signals. Tsinalis et al. [23] used CNN for classification achieving an overall accuracy of 74% over all subjects of the Sleep-EDF dataset from single-channel EEG data. Chambon et al. [24] proposed the first end-to-end deep learning approach for sleep stage classification using multivariate and multimodel PSG signals with a eleven layer deep 2D CNN which achieved an accuracy of 91%. Acharya et al. [25] creatively proposed a novel sleep stage classification method based on high-order spectra (HOS), in which the authors extracted features based on unique bispectrum and bicoherence plots of various sleep stages and then used the Gaussian mixture model (GMM) classifier for automatic identification achieving an accuracy of 88.7%.

Bai et al. [26] show that Temporal Convolution Networks (TCNs) outperform canonical recurrent architectures such as LSTMs and GRUs owing to dilated causal convolutions. This type of convolutions enable TCNs to have the following advantages:

- The convolutions can be performed in parallel since the filter used in each layer is the same hence achieving parallelism.
- The receptive field size can be altered in various ways and can be made flexible.
- Since the filters are shared across a layer, the memory requirements of TCNs are far less as compared to that of LSTMs and GRUs.

Vilamala et al. [8] applied TCNs to detect anomalies in time series data suggesting their capability in learning time series behaviors. Oord et al. [11] introduced WaveNet, a state-of-the-art generative model which is currently being used for producing realistic-sounding Google Assistant voices in Japanese and US English. WaveNet is a fully probabilistic and autoregressive deep neural network for raw audio data which makes use of dilated causal convolutions and a residual learning framework. The framework comprises of skip connections and identity mappings for faster convergence and training.

III. Methodology

A. Stacked Generalization

Stacking or Stacked Generalization [10] is a supervised ensemble method which utilizes high-level models (meta-learners) to combine the predictions of several low-level models (base-learners) in the most optimal way. This in turn helps the meta-learner to achieve better prediction accuracy and minimize the generalization error as it averages out the noise from the various base-learners. Also, a stacking model is able to obtain more information on the hypothesis space by using the predictions of the base learners as features and conditionally weigh them resulting in better performance.

Usually, the meta-learner outperforms each of the base-learners because of its smoothing nature and offsetting of individual deficiencies of base-learners. It has been shown to represent an asymptotically optimal system for learning [27]. Stacking is well-established due to its advantages and has been used for the tasks of classification [10], regression [28] and unsupervised learning [29].

B. SpectroTemporalNet

Ensemble models yield better results when the base-learners have significant diversity, high variability and uncorrelated predicted values. Since SpectroTemporalNet treats sleep stage scoring as a visual as well as a temporal task, stacked generalization leads to a better performance than existing approaches. To that end, this paper puts forward a stacked ensemble model called SpectroTemporalNet by stacking two neural networks called TemporalNet and SpectralNet as base-learners. The architectures of these neural networks are explained in Sections III-C and III-D.

The SpectroTemporalNet architecture is formed by concatenating the SpectralNet and the TemporalNet. For each of the base-learners, the top layers are removed keeping their respective global average pooling layers intact. The resulting networks output 1D tensors of shape 128 which are combined to form a 1D tensor of shape 256 using a concatenate layer. This layer is followed by an alternating sequence of dropout layers and dense layers. The two dropout layers with a dropout

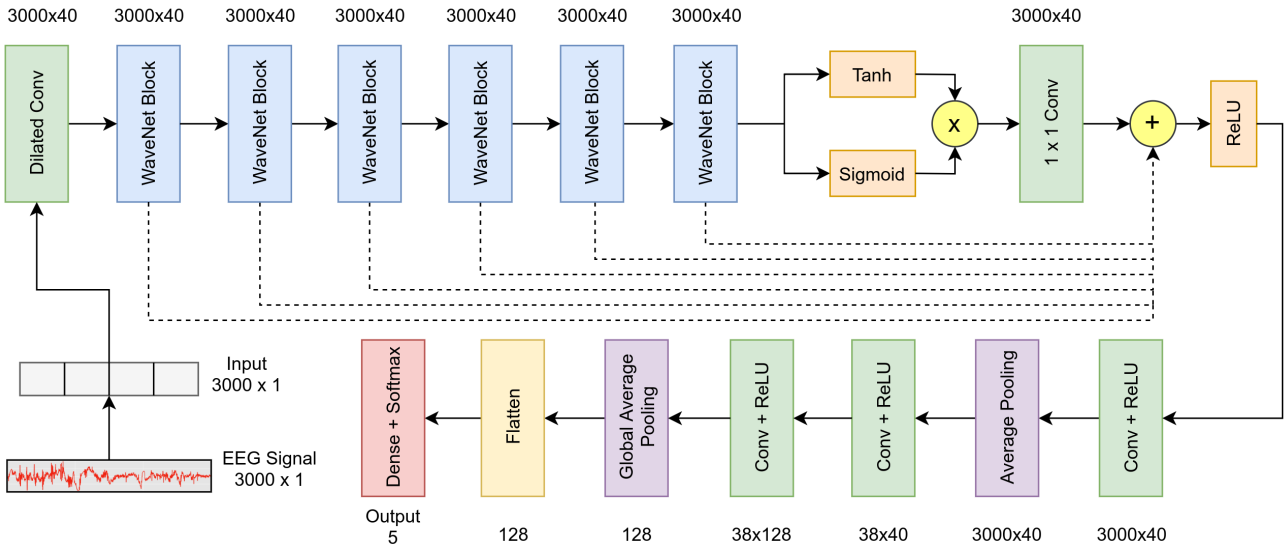

Fig. 3. Block diagram of the TemporalNet architecture

rate of 0.5 provide a regularization effect on the model and reduce overfitting [30]. The first dense layer has a leaky rectified linear unit (ReLU) activation with a negative slope of 0.2 which minimizes the dying ReLU problem [31, 32]. It is followed by batch normalization which aids in better optimization and convergence [33]. The second dense layer has a softmax activation to output five class probabilities, one for each of the five sleep stages corresponding to W, N1, N2, N3 and REM.

The multi-headed stacking ensemble so formed as illustrated in Fig. 2 can be treated as a single large model which allows the intermediate outputs of the two base-learners to be directly fed to the meta-learner. The primary objective of training the meta-learner then becomes to learn optimal weights for the weighted average of the base-learners.

C. TemporalNet

Although WaveNet is a generative model for audio, it can flexibly adapt to various tasks including the use case of a classifier since the waveforms of EEG signals are similar to that of audio sequences [11, 34]. Hence, this study proposes a 1D CNN based on the WaveNet architecture called TemporalNet as shown in Fig. 3. The input for this network is a normalized EEG signal mini-batch represented by a 3D tensor of shape $(batch\_size \times timesteps \times features)$.

Dilated causal convolutions are pivotal to the proposed temporal network. These convolutions enable long effective history sizes, i.e. they aid the network in having the ability to look very far into the past to make a prediction. The stacking of layers with an increasing dilation rate allows the receptive field of TemporalNet to grow exponentially with depth while having fewer layers. This means that it can handle long term dependencies prevalent in EEG data by covering thousands of timesteps and learn dense features from them. Additionally, a WaveNet based architecture not only ensures to be more accurate than canonical recurrent networks such as LSTMs and GRUs, but it also has the advantages of parallelism and flexible receptive field sizes.

Causal convolutions are pivotal as they ensure that the model does not violate the ordering of how the data is modeled and there is no information "leakage" from future to past. Since recurrent connections are absent from models with causal convolutions, it is faster to train TemporalNet than traditional RNNs on long sequences such as EEG data. But causal convolutions only permit the receptive field size to increase linearly with depth. To circumvent this issue, the authors of [11] employ dilated convolutions that enable an exponentially large receptive field [34, 35, 36]. The network can effectively work on a coarser scale than with traditional convolutions without greatly increasing the computational cost.

Also, a residual module is used to stabilize the network as demonstrated in Fig. 4 [37]. Skip connections let the data completely bypass the convolution layers and provide the ability to influence the predictions over an arbitrary number of future periods in time. In addition to that, gated activation units are used as the non-linearity for each node in the network as they work relatively better than ReLU activations [11]. The gated non-linearity is given by (1). These characteristics overcome the vanishing gradient problem and speed up the convergence while training the model.

$$z = tanh(Wf, k * x) \odot \sigma(Wg, k * x) \tag{1}$$

where $*$ denotes a convolution operator, $\odot$ denotes an element-wise multiplication operator, $\sigma(\cdot)$ is a sigmoid function, k is the layer index, f and g denote filter and gate, respectively, and W is a learnable convolution filter.

The dilation rate is doubled for every layer of the TemporalNet up to a limit given by the sequence: $\{2, 4, 8, 16, \ldots, 512\}$. The advantages of this configuration are two-fold. Firstly, the

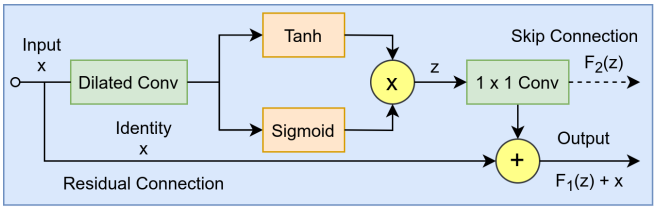

Fig. 4. The WaveNet residual block that is adapted for TemporalNet

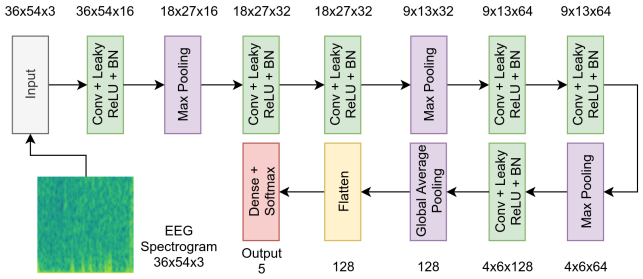

Fig. 5. Block diagram of the SpectralNet architecture

exponential growth of the receptive field with depth is made possible by exponentially increasing the dilation rate [38]. Secondly, stacking these layers increases the temporal support and the capacity of the model. As shown in Fig. 3, several WaveNet blocks are sequentially stacked together. The initial dilated causal convolution in each of these blocks (Fig. 4) is of stride 1 with a filter size of 2 and dilation rate of 1.These residual blocks are followed by a series of ReLU activation, convolution, average pooling and global average pooling layers to decrease the size of the feature maps to 128. The pool size for the average pooling layer is set to 100. Finally, a dense layer with a softmax activation is used to output class probabilities corresponding to the five sleep stages.

D. SpectralNet

Since a spectrogram is the way of visually representing the frequency spectrum of a signal, a deep 2D CNN is employed to perform feature extraction and classification from spectrogram images. Therefore, a sequential 2D CNN inspired from the popular VGG-16 architecture [6] called SpectralNet is proposed. The aim of this model is to provide an additional signal to the stacked ensemble based on the spectral features of EEG signals. A mini-batch of normalized spectrograms of the shape $(batch\_size \times height \times width \times channels)$ is fed as an input to the SpectralNet.

The spectrograms of EEG signals undergo a series of six convolution operations with a stride of (1×1), filter size of (3×3) and a padding of $(1, 1)$. Each convolution is followed by a leaky ReLU activation and batch normalization as presented in Fig. 5. Spatial pooling is done using 3 max pooling layers with a pool size of (2×2) and stride (1×1). The size of the spectrogram feature map is reduced to 128 using a global average pooling layer. The top of the network includes a dense layer with a softmax activation outputting five class probabilities for the different sleep stages similar to that of TemporalNet.

A global average pooling layer is utilized in both the architectures replacing the fully connected layers present in a classical CNN. This configuration is advantageous in two ways. Firstly, it allows both the base-learners to be fully convolutional (FC) except the last softmax layers for classification. In turn, the size of the inputs to the base-learners need not be fixed beforehand. Secondly, the global average pooling layer doesn't have parameters to optimize thus reducing the tendency of overfitting. Moreover, it enables the feature maps to be more closely related to the classification categories [39]. As a result, SpectroTemporalNet achieves a good compromise between the computational complexity and the modeling performance.

IV. Experimental Setup

A. Dataset Overview

For the purpose of this study, the EEG data obtained from the PhysioNet Sleep-EDF public database [3, 19] is used. The hypnogram files present in the database consists of sleep patterns for each subject. The hypnograms represent an event for segments (epochs) of 30 s.

The Sleep-EDF dataset contains Polysomnograms (PSGs) sleep recordings with a sampling frequency of 100 Hz. Each entry in the database consists of EEG readings from two channels, $F_pz - Cz$ and $Pz - Oz$ electrode locations [40]. All the sleep stages mentioned are assigned to a special class. The stages W, N1, N2, N3 and REM correspond to classes 0, 1, 2, 3 and 4 respectively.

The data for each subject consists of a time series data collected from seven channels out of which there are two EEG channels and their corresponding electrode locations are $F_pz - Cz$ and $Pz - Oz$. Since best results are observed for the channel $F_pz - Cz$, this work uses the single-channel $F_pz - Cz$ EEG data. The recording on an average for each subject is of 22 hours and the sampling frequency of the data is 100 Hz. For each subject, the total number of data points are $(22 \times 60 \times 60 \times 100)$.

B. Dataset Preprocessing

The Sleep-EDF dataset consists of 1D EEG time series data of 30 s for multiple subjects. The process of epoching is performed on the EEG signals for each subject by segmenting them into signals of 30 s. Thus, the average number of samples for each subject are 2640. Each sample consists 3000 timesteps, i.e. (30×100). Since, the SpectralNet takes normalized spectrograms as an input and the TemporalNet requires normalized EEG signals, we first preprocess the raw EEG samples, each having a shape of 3000. The EEG samples are first converted to corresponding RGB spectrogams using Short-time Fourier transform (STFT) with overlap as shown in Fig. 6.

A Hanning window of 0.0002 s with 100 data points in each block for the FFT and 99 data points of overlap between blocks is applied to 1D EEG signals to create spectrograms of shape

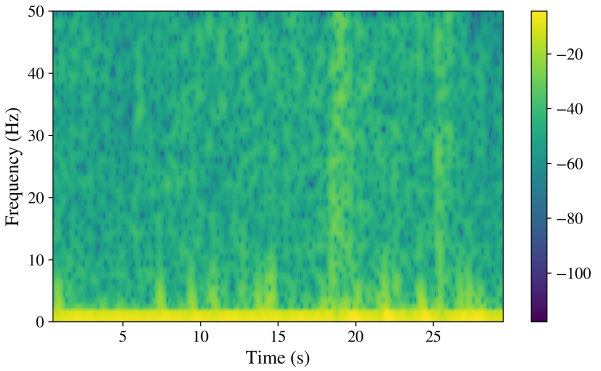

Fig. 6. A spectrogram representing the power spectral density of a 30 s EEG signal from Class W of the PhysioNet Sleep–EDF Database

$(36 \times 54 \times 3)$. It is necessary to perform normalization to rescale the values of the data into a range of $[0.0, 1.0]$ as it improves the numerical stability of the model and leads to faster convergence. Consequently, both the 1D EEG signals and their 2D spectrograms are normalized using Min-Max Scaling given in (2). Also, the corresponding labels for each EEG samples are one-hot encoded. A superfluous channel dimension is added to the normalized EEG data as it is taken from a single EEG channel. Thus, each of the resulting 1D EEG sample has the shape of (3000×1).

$$X_{scaled} = \left(\frac{X - X_{min}}{X_{max} - X_{min}} \right) \times (max - min) + min \quad (2)$$

where X_{scaled} is the normalised value of the feature, X is the original value of the feature, X_{min} and X_{max} are the minimum and maximum values of features in a timestep and $[min, max]$ represents the range of the transformed data. In this scenario, $min = 0.0$ and $max = 1.0$

In this study, 85% of the dataset is allocated to the training set, 34% of the training set is allocated to the validation set and the remaining 15% of the whole dataset is allocated to the test dataset. This split is done for a dataset with 54587 samples.

C. Model Training and System Specifications

For the experiments, an Intel Xeon CPU @ 2.30 GHz processor equipped with a Nvidia Tesla P100 GPU was utilized. All three of the aforementioned models were trained from scratch on the PhysioNet Sleep-EDF dataset with a mini-batch size of 16. All the weights were initialized using the random initialisation procedure of [41].

The training of the models was carried out in two phases by minimizing the categorical cross-entropy error ($\mathcal{L}_{CCE}$) given in (3). In the first phase, TemporalNet and SpectralNet were trained individually on the training and validation sets. The input to the TemporalNet comprised of preprocessed EEG signals of shape $(batch\_size \times 3000 \times 1)$, while that of the SpectralNet contained their corresponding RGB spectrograms of shape $(batch\_size \times 36 \times 54 \times 3)$. These two models were trained using the Adam optimizer with AMSGrad [42]. The learning rate was initially set to 0.0005 and then decreased linearly by a factor of 0.95. β_1 and β_2 were set to 0.9 and 0.999 respectively. The training for both the models was regularized by weight decay (both L1 and L2 penalty multipliers set to 0.001) and was stopped after 20 epochs as the validation set accuracy didn't improve.

$$\mathcal{L}_{CCE}(y_{o,c},\ \hat{y}_{o,c}) = \sum_{c=1}^{M} y_{o,c} * \log(\hat{y}_{o,c}) \quad (3)$$

where M is the number of classes (For this implementation, $M = 5$), $y_{o,c}$ is binary indicator (0 or 1) indicating the ground truth class label c for observation o and $\hat{y}_{o,c}$ is the predicted probability of class label c for observation o.

The second phase of the training consisted of freezing the parameters of TemporalNet and SpectralNet. The last layers of both the models were removed keeping the global average pooling layers as it is and were concatenated as explained in Section III-B. The meta-learner (SpectroTemporalNet) was trained for 15 epochs with the Adam optimizer and an initial learning rate of 0.0003. The learning rate was linearly decayed by a factor of 0.93 and β_1 and β_2 were set to 0.5 and 0.999 respectively. Regularization of the training for SpectroTemporalNet was achieved by the two dropout layers with a keep probability of 0.5.

V. Results & Discussion

A. Results of the Proposed Models

The evaluation of SpectroTemporalNet's performance is done using metrics such as per-class precision, per-class recall, per-class F1-score, macro-average F1-score and overall accuracy as demonstrated in Table II. Additionally, Table I also shows the validation and test accuracies of all three models.

TABLE I
Loss & Accuracy of the Proposed Models

Model	Loss		Accuracy	
	Validation	*Test*	*Validation*	*Test*
SpectroTemporalNet	0.392	0.371	0.946	0.943
TemporalNet	0.890	0.180	0.934	0.934
SpectralNet	0.477	0.383	0.944	0.939

TABLE II
Classification Report for SpectroTemporalNet

	Precision	Recall	F1-score
Class W	0.98	0.99	0.99
Class N1	0.59	0.30	0.40
Class N2	0.88	0.91	0.90
Class N3	0.91	0.85	0.88
Class REM	0.78	0.83	0.81
Micro-avg.	0.94	0.94	0.94
Macro-avg.	0.83	0.78	0.79
Weighted-avg.	0.94	0.94	0.94

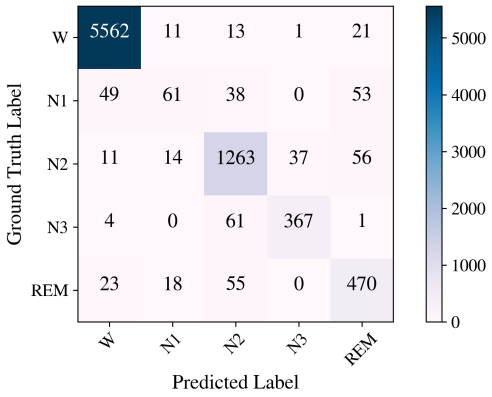

Fig. 7. Confusion Matrix for SpectroTemporalNet

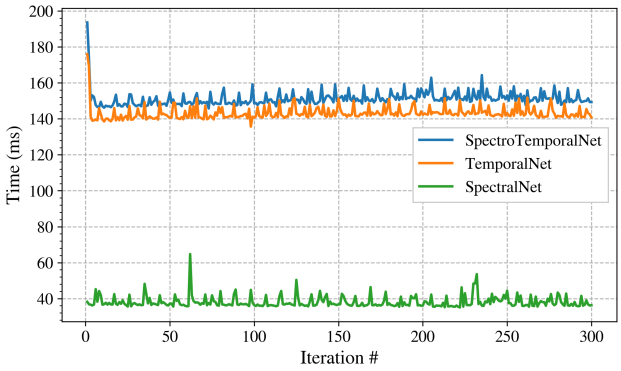

Fig. 8. Inference time of the proposed models on an input batch of 100 samples

As given in Table I, the training and test accuracies obtained are 94.6% and 94.3% respectively. Since the test loss is lesser than the train loss, it implies that the proposed model has good generalization ability and has not overfitted on the dataset. Table II shows that an F1-score of 0.99 was obtained for the Class W. Since the bulk of data belong to this stage, the model showed a trend towards learning the data in this stage. The lowest F1-score 0.40 was observed for class N1. The model found it difficult to classify class N1 examples as it has the least amount of data in terms of data distribution. This is further supported by the confusion matrix shown in Fig. 7.

B. Comparison With Existing Approaches

The results of SpectroTemporalNet are compared with existing methods for the same number of classes. As shown in Table III, SpectroTemporalNet has achieved highest accuracy in five stage sleep classification in comparison to many state-of-the-art classifiers.

C. Inference Time Profiling

In addition to the evaluation metrics, this paper also explores the inference time for each of the three models. Two of the issues encountered while creating an inference time profile for a neural network are, asynchronous execution and GPU power-saving modes [43]. To minimize the effects of these two issues in the run-time profiling process, the *torch.cuda.synchronise()* function is used.

TABLE III
COMPARISON OF RESULTS WITH EXISTING METHODS

Method	Dataset	Classifier	Acc.
Cui et al. [17]	ISRUC-Sleep	1D CNN	0.922
Yildirim et al. [18]	Sleep-EDF	1D CNN	0.908
Aboalayon et al. [20]	Sleep-EDF	DT	0.933
		SVM	0.924
		NN	0.917
		KNN	0.894
Michielli et al. [21]	Sleep-EDF	LSTM	0.867
Supratak et al. [22]	Sleep-EDF	1D CNN + LSTM	0.820
Proposed Method	Sleep-EDF	**1D CNN + 2D CNN**	**0.943**

This function performs synchronization between the GPU and CPU to prevent asynchronous execution. Moreover, the GPU is "warmed up" before making the time measurements by running inference on a dummy input for 100 iterations. Taking into account the variance of the run-time, inference is made on all three models over the same mini-batch of 100 samples over 300 iterations. The graph comparing the inference time for the three models in each iteration is shown in Fig. 8.

VI. CONCLUSION

Firstly, this study formulates sleep stage scoring as a classification task encompassing both, the temporal and spectral features of single-channel raw EEG signals. Subsequently, it proposes a Deep Learning model called SpectroTemporalNet based on Stacked Generalization. This ensemble architecture consists of two CNN base-learners called SpectralNet and TemporalNet for capturing the spectral and temporal features of EEG signals respectively. Secondly, this study performs a quantitative evaluation of the aforementioned models on the PhysioNet Sleep-EDF dataset. SpectroTemporalNet achieves a classification accuracy of 94.31% and sensitivity of 94% on the test set. Based on the comparison of this study with the existing methods, SpectroTemporalNet achieves a better generalization performance.

The authors highly encourage future work to be done on improving certain areas in the proposed work:

- The low F1-score of class *N1* is due to imbalance in distribution of the sleep stages in the dataset used. To overcome the dataset imbalance problem, the Synthetic Minority Oversampling Technique (SMOTE) and Tomek Links Under sampling could be utilized [44].
- Sourcing the data from different laboratories increasing the number of subjects can lead to a more diverse dataset which can be useful for better training.

This study contributes to the research towards building an automated, efficient and convenient system for scoring sleep stages. This can aid in the detection of sleep disorders using PSG signals.

REFERENCES

[1] Iber, Conrad & Ancoli-Israel, Sonia & Chesson, A.L. & Quan, Stuart. (2007). The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. Westchester, IL: American Academy of Sleep Medicine.

[2] Baran AS, Chervin RD. Approach to the patient with sleep complaints. Semin Neurol. 2009 Sep;29(4):297-304. doi: 10.1055/s-0029-1237116. Epub 2009 Sep 9. PMID: 19742407.

[3] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation. 2000 Jun 13;101(23):E215-20. doi: 10.1161/01.cir.101.23.e215. PMID: 10851218.

[4] Wolpert EA. A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects. Arch Gen Psychiatry. 1969;20(2):246–247. doi:10.1001/archpsyc.1969.01740140118016.

[5] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[6] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[7] Kiranyaz, Serkan & Avci, Onur & Abdeljaber, Osama & Ince, Turker & Gabbouj, Moncef & Inman, Daniel. (2019). 1D Convolutional Neural Networks and Applications: A Survey.

[8] Yangdong He & Jiabao Zhao 2019 J. Phys.: Conf. Ser. 1213 042050.

[9] Vilamala, Albert & Madsen, Kristoffer & Hansen, Lars. (2017). Deep Convolutional Neural Networks for Interpretable Analysis of EEG Sleep Stage Scoring.

[10] Wolpert, David. (1992). Stacked Generalization. Neural Networks. 5. 241-259. 10.1016/S0893-6080(05)80023-1.

[11] oord, Aaron & Dieleman, Sander & Zen, Heiga & Simonyan, Karen & Vinyals, Oriol & Graves, Alex & Kalchbrenner, Nal & Senior, Andrew & Kavukcuoglu, Koray. (2016). WaveNet: A Generative Model for Raw Audio.

[12] Koudelková, Zuzana & Strmiska, Martin. (2018). Introduction to the identification of brain waves based on their frequency. MATEC Web of Conferences. 210. 05012. 10.1051/matecconf/201821005012.

[13] Craik, Alexander & He, Yongtian & Contreras-Vidal, José. (2019). Deep learning for Electroencephalogram (EEG) classification tasks: A review. Journal of Neural Engineering. 16. 10.1088/1741-2552/ab0ab5.

[14] Q. Liu et al., "Spectrum Analysis of EEG Signals Using CNN to Model Patient's Consciousness Level Based on Anesthesiologists' Experience," in IEEE Access, vol. 7, pp. 53731-53742, 2019, doi: 10.1109/ACCESS.2019.2912273.

[15] S. Nilufar, N. Ray, M. K. I. Molla and K. Hirose, "Spectrogram based features selection using multiple kernel learning for speech/music discrimination," 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012, pp. 501-504, doi: 10.1109/ICASSP.2012.6287926.

[16] Ruffini, Giulio & David, Ibanez-Soria & Castellano, Marta & Dubreuil Vall, Laura & Soria-Frisch, Aureli & Postuma, Ron & Gagnon, Jean-François & Montplaisir, Jacques. (2019). Deep Learning With EEG Spectrograms in Rapid Eye Movement Behavior Disorder. Frontiers in Neurology. 10. 806. 10.3389/fneur.2019.00806.

[17] Cui, Zhihong & Zheng, Xiangwei & Shao, Xuexiao & Cui, Lizhen & Chen, Jingbo & Dinero, Juan. (2017). Automatic Sleep Stage Classification Based on Convolutional Neural Network and Fine-Grained Segments. Complexity. 2018. 10.1155/2018/9248410.

[18] Yildirim O, Baloglu UB, Acharya UR. A Deep Learning Model for Automated Sleep Stages Classification Using PSG Signals. Int J Environ Res Public Health. 2019 Feb 19;16(4):599. doi: 10.3390/ijerph16040599. PMID: 30791379; PMCID: PMC6406978.

[19] Kemp, Bob & Zwinderman, Aeilko & Tuk, Bert & Kamphuisen, Hilbert & Oberyé, Josefien. (2000). Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG. Biomedical Engineering, IEEE Transactions on. 47. 1185 - 1194. 10.1109/10.867928.

[20] Aboalayon, Khald & Faezipour, Miad & Almuhammadi, Wafaa & Moslehpour, Saeid. (2016). Sleep Stage Classification Using EEG Signal Analysis: A Comprehensive Survey and New Investigation. Entropy. 18. 10.3390/e18090272.

[21] Michielli, Nicola & Acharya, U Rajendra & Molinari, Filippo. (2019). Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals. Computers in Biology and Medicine. 106. 10.1016/j.compbiomed.2019.01.013.

[22] Supratak, Akara & Dong, Hao & Wu, Chao & Guo, Yike. (2017). DeepSleepNet: a Model for Automatic Sleep Stage Scoring based on Raw Single-Channel EEG. IEEE Transactions on Neural Systems and Rehabilitation Engineering. PP. 10.1109/TNSRE.2017.2721116.

[23] Tsinalis, Orestis & Matthews, Paul & Guo, Yike & Zafeiriou, Stefanos. (2016). Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks.

[24] Chambon, Stanislas & Galtier, Mathieu & Arnal, Pierrick & Wainrib, Gilles & Gramfort, Alexandre. (2017). A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series. IEEE Transactions on Neural Systems and Rehabilitation Engineering. PP. 10.1109/TNSRE.2018.2813138.

[25] Acharya UR, Chua EC, Chua KC, Min LC, Tamura T. Analysis and automatic identification of sleep stages using higher order spectra. Int J Neural Syst. 2010 Dec;20(6):509-21. doi: 10.1142/S0129065710002589. PMID: 21117273.

[26] Bai, Shaojie & Kolter, J. & Koltun, Vladlen. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.

[27] Laan, Mark & Polley, Eric & Hubbard, Alan. (2007). Super Learner. Statistical applications in genetics and molecular biology. 6. Article25. 10.2202/1544-6115.1309.

[28] Breiman, L. Stacked Regressions. Machine Learning 24, 49–64 (1996). doi: 10.1023/A:1018046112532.

[29] Smyth, P., Wolpert, D. Linearly Combining Density Estimators via Stacking. Machine Learning 36, 59–83 (1999). doi: 10.1023/A:1007511322260.

[30] Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 15. 1929-1958.

[31] Lu, Lu & Shin, Yeonjong & Su, Yanhui & Karniadakis, George. (2019). Dying ReLU and Initialization: Theory and Numerical Examples.

[32] Xu, Bing & Wang, Naiyan & Chen, Tianqi & Li, Mu. (2015). Empirical Evaluation of Rectified Activations in Convolutional Network.

[33] Ioffe, Sergey & Szegedy, Christian. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

[34] Boilard, Jonathan & Gournay, Philippe & Lefebvre, R. (2019). A Literature Review of WaveNet: Theory, Application and Optimization.

[35] Bai, Shaojie & Kolter, J. & Koltun, Vladlen. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.

[36] Dilated Convolutions and Kronecker Factored Convolutions, May 2016. [Online]. Available: https://www.inference.vc/dilated-convolutions-and-kronecker-factorisation]

[37] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[38] Yu, Fisher & Koltun, Vladlen. (2016). Multi-Scale Context Aggregation by Dilated Convolutions.

[39] Lin, M., Chen, Q., & Yan, S. (2014). Network In Network. CoRR, abs/1312.4400.

[40] Mousavi, Sajad & Afghah, Fatemeh & Acharya, U Rajendra. (2019). SleepEEGNet: Automated Sleep Stage Scoring with Sequence to Sequence Deep Learning Approach.

[41] Glorot, Xavier & Bengio, Y.. (2010). Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research - Proceedings Track. 9. 249-256.

[42] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[43] Geifman, A., 2020. The Correct Way to Measure Inference Time of Deep Neural Networks. [online] Deci.ai. Available at: https://deci.ai/the-correct-way-to-measure-inference-time-of-deep-neural-networks. [Accessed 25 February 2021].

[44] Batista, Gustavo & Prati, Ronaldo & Monard, Maria-Carolina. (2004). A Study of the Behavior of Several Methods for Balancing machine Learning Training Data. SIGKDD Explorations. 6. 20-29. 10.1145/1007730.1007735.