Deep Neural Model for Automated Sleep Staging System using Single-Channel EEG Signal

Tanmay Rathod
Information and Communication Technology,
Pandit Deendayal Energy University, Gandhinagar, Gujarat
tanmayrathod777@gmail.com
https://orcid.org/0000-0000-0000-0001 0000-0000-0000-0001

Santosh Kumar Satapathy
Information and Communication Technology,
Pandit Deendayal Energy University, Gandhinagar, Gujarat
abc@google.com
https://orcid.org/0000-0000-0000-0002 0000-0000-0000-0002

Abstract

Accurate sleep staging is essential for diagnosing sleep disorders, traditionally relying on manual analysis of EEG signals. In this study, we utilize the Sleep-EDF dataset and preprocess the EEG signals by segmenting them into chunks and transforming them into PKL file format for efficient processing. The dataset undergoes thorough preprocessing, including noise removal and normalization, ensuring high-quality input for classification. We then evaluate the performance of multiple machine learning and deep learning models in automating sleep stage classification. We experiment with Random Forest, K-Nearest Neighbors, XGBoost, AdaBoost, and ensemble methods, along with deep learning architectures such as LSTM, BiLSTM, and RNN. Among machine learning models, XGBoost achieved the highest accuracy of 85.35%, while BiLSTM outperformed other deep learning models with an accuracy of 81.13%. We also assess validation and testing metrics to ensure model robustness. The results highlight the effectiveness of deep learning in sleep stage classification and suggest further improvements through model optimization and multi-modal data integration.

Keywords: Sleep Staging, EEG Signal Processing, Machine Learning, Deep Learning, BiLSTM, XGBoost, Sleep-EDF, PKL File, LSTM, BiLSTM, RNN.

I. Introduction

Sleep is a crucial part of the human life cycle. To maintain a healthy mind and proper brain function, it is recommended to get at least 7 to 9 hours of sleep each night. A lack of sleep can significantly reduce concentration and overall efficiency, affecting daily performance. Today's modern lifestyle and changes in dietary habits have led many people to experience sleep complications, which in turn disrupt work-life balance and may even contribute to serious conditions such as sleep apnea, insomnia, restless legs syndrome, and hypersomnia.

Table 1: Brainwave classifications and their characteristics.

Wave Type	Description
Alpha Waves (8–12 Hz)	Associated with relaxed, wakeful states; prominent during meditation or a calm mind.
Beta Waves (13–30 Hz)	Indicative of active, alert, and focused states; seen during problem-solving or engaging tasks.
Theta Waves (4–8 Hz)	Observed during light sleep or drowsiness; also present in deep relaxation or meditation.
Delta Waves (0.5–4 Hz)	Predominant during deep sleep; occasionally seen in certain brain disorders.
Gamma Waves (30–50 Hz)	Linked to high cognitive processing and attention; less relevant for sleep analysis.

Doctors and experts often recommend tests like EEG and other biosignal assessments to diagnose these issues. In a controlled environment, data is collected through methods such as EDF files, hypnograms, and sometimes video recordings. These recordings capture a variety of biosignals, including EEG, EMG, EOG, ECG, temperature, and rectal measurements. For our experiment, we used data from the Sleep-EDF dataset, where experts have already annotated the recordings by classifying sleep stages into N1, N2, N3, N4, wake, and REM.

Figure 1: Sleep signals representation.

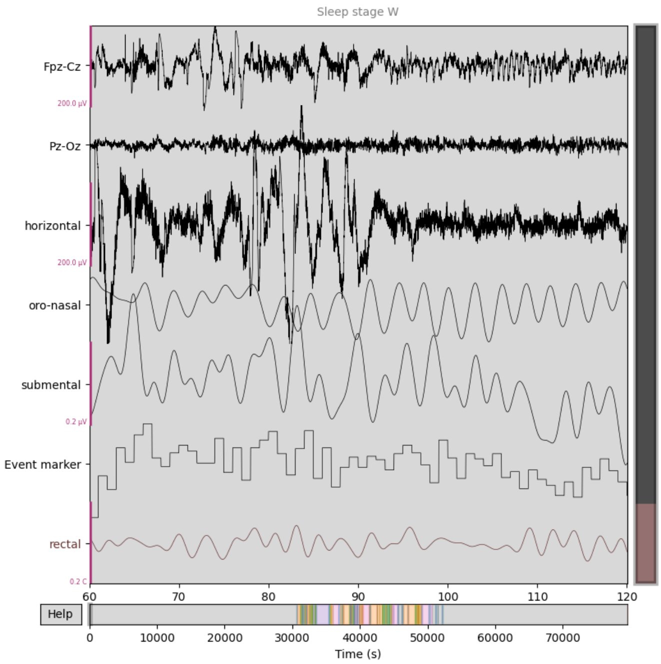
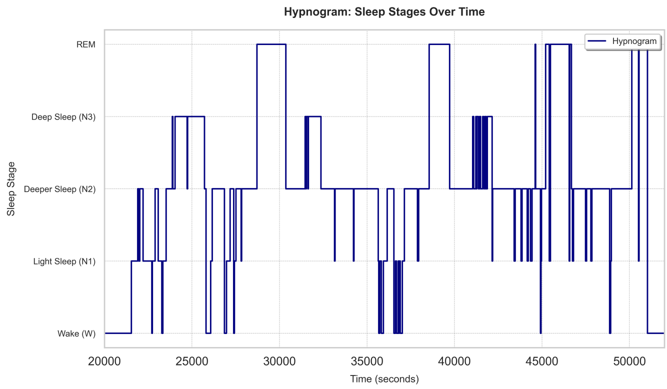


Figure 2: Hypnogram plot showing sleep stage transitions.



We briefly discuss the preprocessing steps in this study, where these signals are transformed into classified sleep stages. This approach has the potential to automate the manual and tedious work traditionally performed by experts. In our research, we explore various machine learning methods—such as neural networks, LSTM, BiLSTM, and RNN—to train models that accurately classify sleep stages and robustly perform on unseen data.

II. Contributions

we emphasize automating sleep stage categorization with deep learning, in particular for utilizing EEG signals from the Sleep-EDF dataset. Conventionally, sleep staging involves experts manually examining EEG recordings and labeling various sleep stages—a laborsome and time-consuming task. We aim to mechanize this task using machine learning and deep neural networks, resulting in faster and improved sleep staging.

To do this, we begin with large data preprocessing. We break up the raw EEG signals into 30-second windows, so we maintain consistency in how the data is being represented. Because actual sleep data is usually imbalanced (some of the sleep stages are more prevalent than others), we apply the Synthetic Minority Over-sampling Technique (SMOTE) to balance the data. This process aids the model in classifying all the sleep stages more effectively.

We try various machine learning models, such as Random Forest, XGBoost, and AdaBoost, and deep learning models like LSTM, BiLSTM, and RNN. Our model based on BiLSTM is unique because it has the ability to read EEG patterns both in forward and backward directions, thus picking crucial transitions between the sleep stages. The model is composed of recurrent layers for learning temporal patterns, dropout layers for preventing overfitting, and optimal activation functions for better performance.

To accelerate the whole process, we use parallel processing so that we can process large-scale EEG data efficiently.We divide the dataset into pretext, training, and testing sets to ensure a balanced evaluation. We train our model for the over 50 epochs, and we evaluate its performance based on accuracy metrics, loss curves, and confusion matrices.

The results show that our deep learning approach is significantly reduces the need for human annotation of sleep stages is offering a more fast and reliable alternative. This has the potential to assist sleep clinicians and researchers in diagnosing sleep disorders like insomnia and sleep apnea more efficiently. Our research demonstrates how deep neural networks can revolutionize EEG-based sleep staging and paves the way for future improvements, including deeper architectures and multi-modal data fusion.

III. Optimizing Research: Tools, Datasets, and System Requirements

To ensure efficient implementation and reproducibility, this research follows a structured computational setup, version control, and essential libraries for data processing and deep learning.

A. *System Configuration*

The experiments are conducted on the following hardware and software setup:

- **Operating System:** Windows/Linux
- **Processor:** Intel Core i5 11th Gen (Iris Xe Graphics, 3.9 GHz)
- **Python Version:** 3.13.1
- **Package Manager:** pip

B. *Dataset*

The study utilizes the Sleep-EDF dataset, which provides EEG recordings and hypnograms for sleep stage classification. The dataset is preprocessed into fixed-size segments before training the models. The dataset can be accessed at: https://www.physionet.org/content/sleep-edf/1.0.0/

C. *Version Control*

The project is managed using GitHub for efficient version control and collaboration. The source code and implementation details are available at: https://github.com/tanmay007thor/ContraWR

D. *Key Libraries and Versions*

The following libraries are used for data preprocessing, model training, and evaluation:

- NumPy (2.2.2) - Numerical computations
- scikit-learn (1.6.1) - Machine learning utilities
- imbalanced-learn (0.13.0) - Handling class imbalance (SMOTE)
- MNE (1.9.0) - EEG signal processing
- TensorFlow/Keras (2.16.1) - Deep learning model implementation
- Matplotlib (3.10.0) Seaborn (0.13.2) - Data visualization

This setup ensures smooth execution of machine learning and deep learning models, including BiLSTM and other architectures, for automating sleep stage classification.

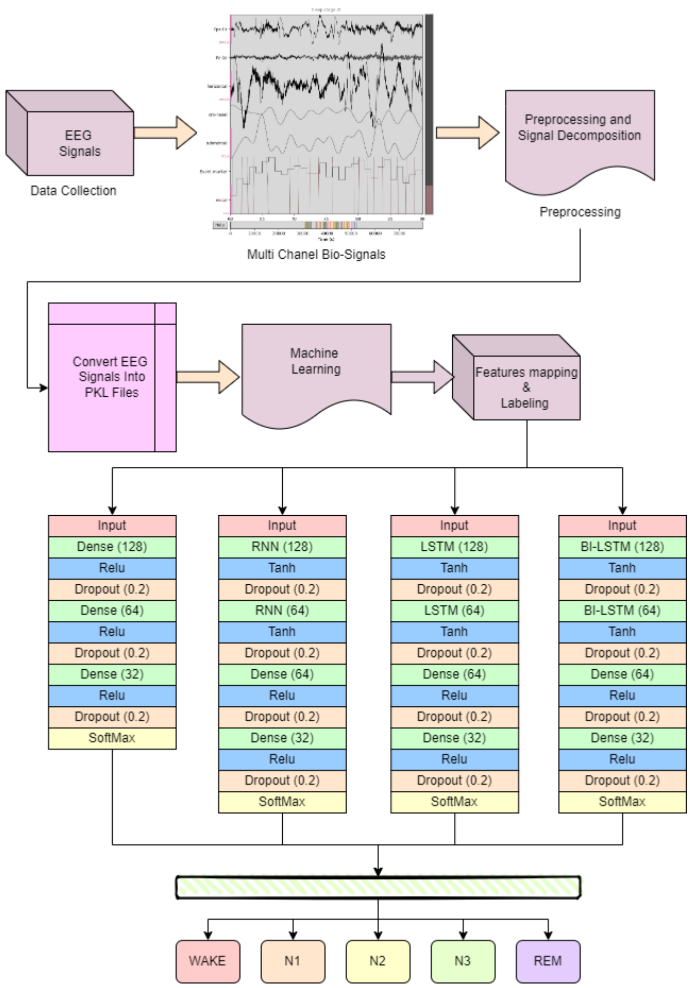
IV. Methodology

we employed the Sleep-EDF dataset wherein it has two file types: PSG in European Data Format (EDF) and Hypnogram files. The EDF contains several bio-signal channels, whereas Hypnogram files contain corresponding sleep stage labels for particular time durations. For preprocessing, we standardized the data to 100 Hz sampling rate and divided the data into 30-second epochs, achieving 3000 time steps per sample. These samples were subsequently stored as 48 KB chunks in the PKL format, making the data handling efficient and enabling models that are light enough to work on edge devices.

The data was organized into input (X) and output (Y) parts. The input (X) contained two EEG channels, namely FPZ-CZ and PZ-OZ, whereas the labels (Y) indicated the sleep stages and were labeled as Wake, N1, N2, N3, and REM. To ease processing, we also mapped the raw labels to a predefined class mapping. The dataset was further split into training, validation, and test sets in the ratio 90:5:5. The dataset was class-imbalanced, so we used the

SMOTE (Synthetic Minority Over-sampling Technique) technique in order to balance the dataset. We finally used several machine learning and deep learning models, which will be explained in the following sections.

Figure 3: Proposed Deep Learning Architecture for Sleep Stage Classification



1. Deep Neural Network (DNN)

Deep Neural Network (DNN) is a dense feedforward neural network made up of several dense layers. Here, the input layer is preceded by three dense layers of 128, 64, and 32 neurons, respectively, each activated through the ReLU function.

Table 2: Simple Neural Network Architecture

Layer Type	**Neurons/Units**	**Activation Function**
Input Layer	`X_train`	-
Dense Layer	128	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	64	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	32	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	Number of classes (e.g., 5)	Softmax

For avoiding overfitting, dropout layers with 0.2 probability are included between the dense layers. The last output layer applies a SoftMax activation function to classify EEG sleep stages into various stages (Wake, N1, N2, N3, REM).

DNNs are suitable for learning hierarchical representations of EEG features but may not be able to learn temporal dependencies in sequential data such as EEG signals.

2. Forward Propagation

During forward propagation, activations are computed layer by layer:

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)} \tag{1}$$
$$a^{(l)} = \sigma(z^{(l)}) \tag{2}$$

where:

- $z^{(l)}$ is the pre-activation output of layer l.
- $a^{(l)}$ is the activation at layer l.
- $W^{(l)}, b^{(l)}$ are weights and biases.

3. Backward Propagation

Backpropagation computes gradients using the chain rule:

$$\delta^{(L)} = \frac{\partial L}{\partial a^{(L)}} \odot \sigma'(z^{(L)}) \tag{3}$$
$$\delta^{(l)} = (W^{(l+1)})^T \delta^{(l+1)} \odot \sigma'(z^{(l)}) \tag{4}$$
$$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)}(a^{(l-1)})^T \tag{5}$$
$$\frac{\partial L}{\partial b^{(l)}} = \sum \delta^{(l)} \tag{6}$$

where:

- $\delta^{(l)}$ is the error term for layer l.
- $\sigma'(z^{(l)})$ is the derivative of the activation function.
- $\frac{\partial L}{\partial W^{(l)}}$ and $\frac{\partial L}{\partial b^{(l)}}$ are weight and bias gradients.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{7}$$

where:

- z_i represents the input to the i^{th} neuron.
- N is the total number of classes.
- The denominator ensures that the outputs sum to 1.

The Rectified Linear Unit (ReLU) activation function is used in the dense layers:

$$f(x) = \max(0, x) \tag{8}$$

where:

- x is the input to the neuron.
- If $x > 0$, the function returns x, otherwise it returns 0.

4. Recurrent Neural Network (RNN)

The Recurrent Neural Network (RNN) is specifically designed to work on sequential data by using recurrent layers which have memory of previously seen information. In this model, an RNN layer with 128 units is used, followed by another recurrent layer with 64 units, both utilizing the Tanh activation function.

Table 3: Simple RNN Neural Network Architecture for Sleep Disorder

Layer Type	**Neurons/Units**	**Activation Function**
Input Layer	`(X_trains)`	-
RNN Layer 1	128	Tanh
Dropout Layer	-	0.2 (Dropout Rate)
RNN Layer 2	64	Tanh
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	64	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	32	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	Number of classes	Softmax

Dropout layers (0.2) are used to enhance generalization. After the recurrent layers, dense layers (64 and 32 neurons) with ReLU activation further refine the learned representations before classification through the SoftMax output layer. Though RNNs can effectively model temporal dependencies in EEG signals, they are susceptible to vanishing gradient, which restricts the capacity to learn long-term dependencies.

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h) \tag{9}$$

where:

- h_t is the hidden state at time t.
- x_t is the input at time t.
- W_h and W_x are weight matrices.
- b_h is the bias term.
- tanh is the activation function.

5. Long Short-Term Memory (LSTM)

LSTM networks are an advancement over RNNs with the inclusion of purposeful memory cells that enable the preservation of long-term dependencies in sequential data.

Table 4: LSTM Neural Network Architecture

Layer Type	**Neurons/Units**	**Activation Function**
Input Layer	`(X_train)`	-
LSTM Layer 1	128	Tanh
Dropout Layer	-	0.2 (Dropout Rate)
LSTM Layer 2	64	Tanh
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	64	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	32	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	Number of classes (e.g., 5)	Softmax

The structure is built with two LSTM layers (128 and 64 units) using Tanh activation, followed by two dropout layers (0.2) to avoid overfitting. The deep layers (64 and 32 neurons) employ ReLU activation prior to the SoftMax classifier. LSTM networks are especially beneficial in EEG-based sleep stage classification since they extract long-range dependencies in sleep patterns, hence performing better than conventional RNNs in processing complicated temporal sequences.

LSTMs improve RNNs by introducing gating mechanisms to control memory flow. The key equations are:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad \text{(Forget Gate)} \tag{10}$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad \text{(Input Gate)} \tag{11}$$
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad \text{(Candidate Cell State)} \tag{12}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad \text{(Cell State Update)} \tag{13}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad \text{(Output Gate)} \tag{14}$$
$$h_t = o_t \odot \tanh(c_t) \quad \text{(Hidden State Update)} \tag{15}$$

where:

- f_t, i_t, o_t are the forget, input, and output gate activations.
- c_t is the memory cell state.
- $\tilde{c}_t$ is the candidate memory cell state.
- $\odot$ represents element-wise multiplication.
- W, U, b are weight matrices and biases.

6. Bidirectional Long Short-Term Memory (Bi-LSTM)

The Bi-LSTM model is extends LSTMs model by processing it's input into sequences in both directions (forward and backward) so that it can learn better in context .

Table 5: Bidirectional LSTM Neural Network Architecture for Sleep Disorder

Layer Type	**Neurons/Units**	**Activation Function**
Input Layer	`(X_train)`	-
Bi-LSTM Layer 1	128	Tanh
Dropout Layer	-	0.2 (Dropout Rate)
Bi-LSTM Layer 2	64	Tanh
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	64	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	32	ReLU
Dropout Layer	-	0.2 (Dropout Rate)
Dense Layer	Number of classes	Softmax

This architecture includes two bidirectional LSTM layers with Tanh activation (128 and 64 units), followed by dropout layers (0.2) to prevent overfitting. The following dense layers (64 and 32 neurons) using ReLU activation facilitate feature extraction prior to classification. Bi-LSTM models work well in the sleep stage classification since they extend the model capacity to learn both past and future EEG signals, leading to improved accuracy in modeling sleep patterns.

Bi-LSTMs extend LSTMs by processing sequences in both forward and backward directions. They maintain two hidden states:

$$\overrightarrow{h}_t = LSTM(x_t, \overrightarrow{h}_{t-1}) \tag{16}$$
$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1}) \tag{17}$$
$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t] \tag{18}$$

where:

- $\overrightarrow{h}_t$ is the forward LSTM hidden state.
- $\overleftarrow{h}_t$ is the backward LSTM hidden state.
- h_t is the concatenated hidden state for final prediction.

V. Results and Analysis

The evaluation of different machine learning models for classification reveals varying levels of effectiveness. Random Forest achieved an accuracy of 84.29%, with a strong precision of 85.21% and recall of 84.29%, demonstrating robust performance, particularly in distinguishing the Wake stage. XGBoost with PCA slightly outperformed Random Forest, achieving an accuracy of 85.35%, with balanced precision (85.41%) and recall (85.35%), indicating its effectiveness in handling imbalanced data. In contrast, K-Nearest Neighbors (KNN) with PCA performed the worst, with an accuracy of only 19.44%, suggesting it struggled to differentiate between sleep stages effectively. AdaBoost with PCA also showed weak performance, reaching an accuracy of 31.23%, with significantly lower recall, highlighting its difficulty in learning complex patterns. Neural Network models demonstrated moderate performance, with an accuracy of 77.73%, but a relatively lower macro F1-score, indicating potential struggles in classifying minority classes. Among recurrent models, LSTM outperformed RNN and Bi-LSTM, achieving an accuracy of 79.97% compared to 70.60% (RNN) and 81.13% (Bi-LSTM). Bi-LSTM had the highest accuracy among deep learning models, with a weighted precision of 79.68%, suggesting its ability to capture sequential dependencies better. Overall, XGBoost and Random Forest emerged as the best-performing models, whereas KNN and AdaBoost failed to generalize well.

Table 6: Precision, Recall, and F1-Score for Different Models

Model	**Precision**	**Recall**	**F1-Score**
Random Forest	0.8521	0.8429	0.8229
KNN (with PCA)	0.2995	0.1945	0.1889
XGBoost (with PCA)	0.8541	0.8535	0.8433
AdaBoost (with PCA)	0.6059	0.3123	0.3529
Neural Network	0.7444	0.7773	0.7499
LSTM	0.7803	0.7997	0.7851
RNN	0.6979	0.7060	0.6898
Bi-LSTM	0.7968	0.8113	0.8030

A. *Deep Neural Network (DNN)*

Figure 4: Confusion Matrix of the Deep Neural Network (DNN)

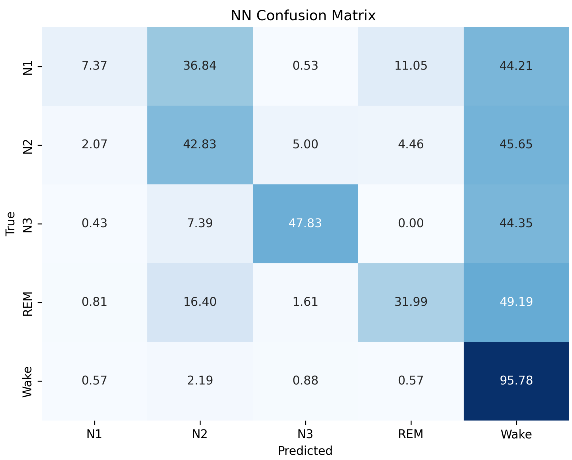
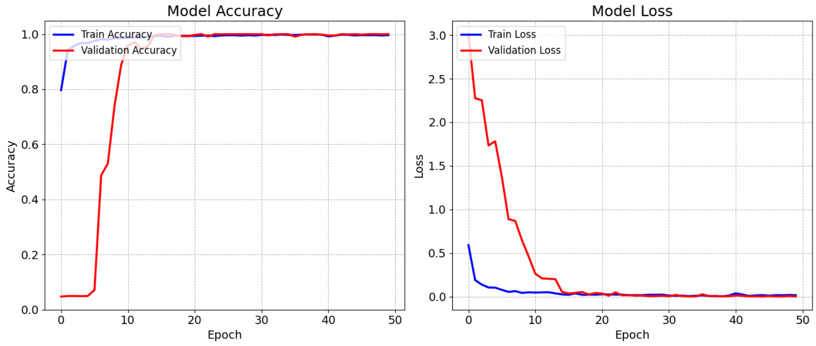


Figure 5: Training and validation curve of the Deep Neural Network (DNN)



B. *Recurrent Neural Network (RNN)*

Figure 6: Confusion Matrix of the Recurrent Neural Network (RNN)

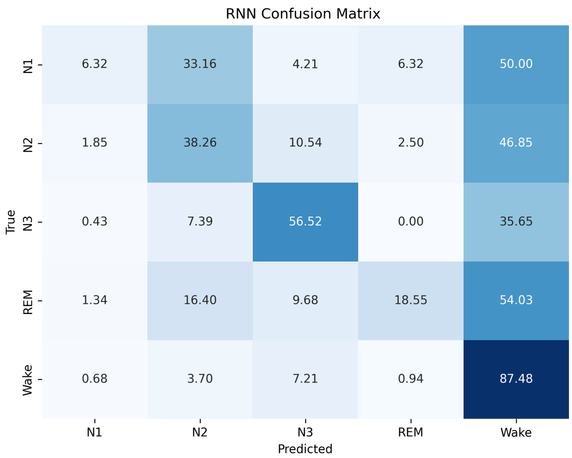
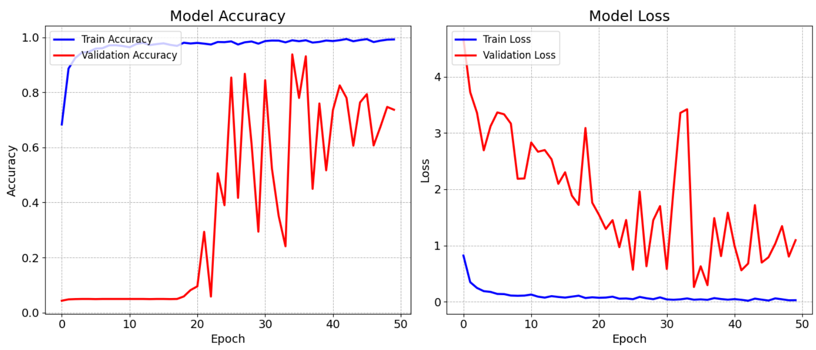


Figure 7: Training and validation Recurrent Neural Network (RNN)



C. *Long Short-Term Memory (LSTM)*

Figure 8: Confusion Matrix Long Short-Term Memory (LSTM)

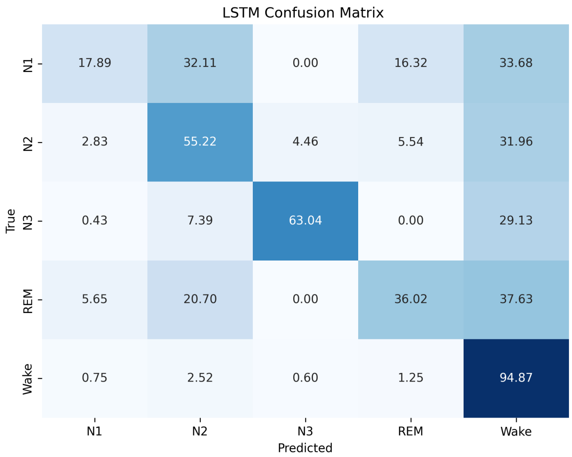
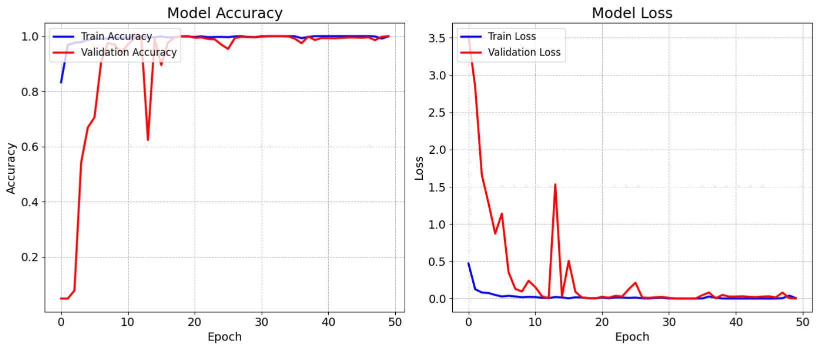


Figure 9: Training and validation curve Long Short-Term Memory (LSTM)



D. *Bidirectional LSTM (Bi-LSTM)*

Figure 10: Confusion Matrix of the Bidirectional LSTM (Bi-LSTM)

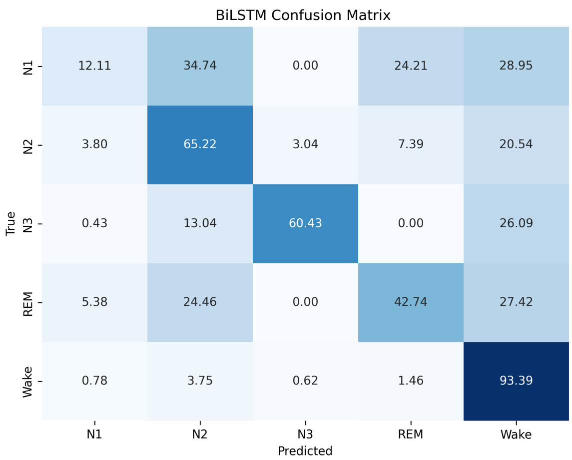
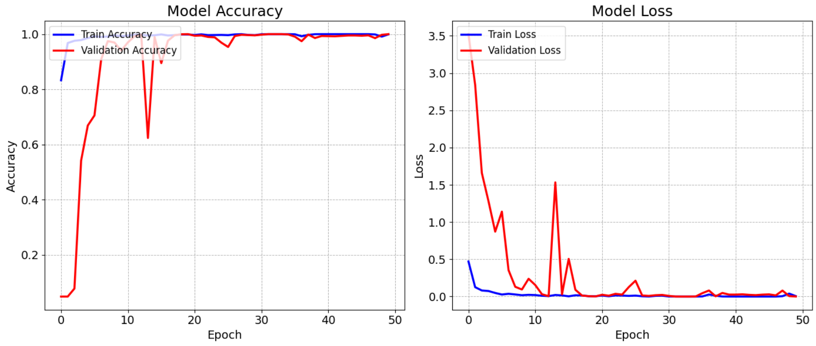
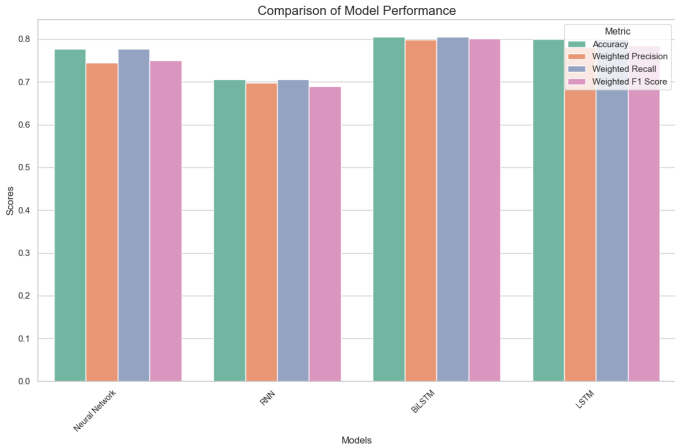


Figure 11: Training and validation curve(Bi-LSTM)



E. *Neural Network Comparison Plots*

Figure 12: Neural Network Comparison Plots Precision , Recall , F1 Score



VI. Discussion

The results of our study highlight the effectiveness of various machine learning and deep learning models in sleep stage classification using EEG signals from the Sleep-EDF dataset. Our experiments indicate that deep learning models, particularly LSTM and Bi-LSTM networks, outperform traditional machine learning models due to their ability to capture temporal dependencies in sequential EEG data.

A. *Impact of Preprocessing and Data Handling*

Standardizing the dataset to a 100 Hz sampling rate and segmenting it into 30-second epochs enabled efficient data representation while preserving relevant temporal features. The use of two EEG channels (FPZ-CZ and PZ-OZ) ensured that critical sleep-related patterns were captured. Furthermore, addressing class imbalance with SMOTE was essential, as sleep stage classification suffers from skewed distributions, particularly for N1 and REM stages. Without this correction, models would likely bias toward majority classes, degrading classification performance.

B. *Performance of Machine Learning vs. Deep Learning Models*

Among the tested machine learning models, **XGBoost** performed the best, achieving an **F1-score of 0.8433**, which is comparable to deep learning approaches. However, other traditional models such as **KNN and AdaBoost** exhibited significantly lower performance, indicating that simple distance-based or boosting methods struggle with complex EEG signal representation.

On the other hand, deep learning models demonstrated superior performance, with **Bi-LSTM achieving the highest F1-score of 0.8030**, followed closely by LSTM (**0.7851**) and a standard neural network (**0.7499**). The recurrent nature of LSTM and Bi-LSTM networks allowed them to better model the sequential nature of EEG signals, making them more effective at distinguishing sleep stages.

C. *Effect of Model Architecture*

The **DNN model**, despite capturing hierarchical representations, lacked the ability to model long-term dependencies, leading to lower performance compared to recurrent architectures. The **RNN model**, while improving upon DNN by incorporating memory, suffered from vanishing gradient issues, limiting its ability to process long sequences. The **LSTM model**, with its memory gating mechanisms, successfully overcame these limitations and captured long-term sleep patterns. Furthermore, the **Bi-LSTM model**, which processes information in both forward and backward directions, demonstrated even better classification performance, reinforcing the importance of bidirectional processing in sleep EEG analysis.

D. *Challenges and Future Directions*

Despite promising results, several challenges remain. First, the presence of overlapping spectral components across different sleep stages makes classification inherently difficult, even for deep models. Additionally, class imbalance, while mitigated using SMOTE, still poses challenges in achieving high recall for underrepresented sleep stages. Future work can explore hybrid models that integrate convolutional layers with LSTMs to improve feature extraction. Another potential direction is incorporating attention mechanisms to enhance the model's ability to focus on relevant EEG signal segments.

Overall, our findings suggest that while traditional machine learning models can provide reasonable performance, **deep learning, particularly LSTM-based architectures, is best suited for sleep stage classification due to its ability to model sequential dependencies and extract temporal features effectively**.

VII. Evaluation Against External Results: Benchmarking and Validation

Table 7: Comparison of Sleep Stage Classification Performance

Model	Architecture	Dataset	Precision	Recall	F1-Score
Random Forest (Our Study)	DNN-based	Sleep-EDF	0.8521	0.8429	0.8229
XGBoost (with PCA, Our Study)	Ensemble ML	Sleep-EDF	0.8541	0.8535	0.8433
Neural Network (Our Study)	Fully-Connected DNN	Sleep-EDF	0.7444	0.7773	0.7499
LSTM (Our Study)	LSTM RNN	Sleep-EDF	0.7803	0.7997	0.7851
RNN (Our Study)	RNN	Sleep-EDF	0.6979	0.7060	0.6898
Bi-LSTM (Our Study)	Bidirectional LSTM	Sleep-EDF	0.7968	0.8113	0.8030
SleepEEGNet [14]	CNN + Seq2Seq LSTM	Sleep-EDF	0.8000*	0.8000*	0.7966*
CNN-Transformer [2]	CNN + Transformer	Sleep-EDF	0.7450*	0.7450*	0.7450*
TinySleepNet [3]	Efficient CNN-LSTM	Sleep-EDF	0.8000*	0.8000*	0.8000*
XSleepNet [4]	CNN-LSTM	Sleep-EDF	0.8700*	0.8700*	0.8700*
RobustSleepNet [5]	Transfer Learning	Sleep-EDF	0.9000*	0.9000*	0.9000*

VIII. Conclusion

Within this paper, we introduced a fully integrated framework to classify sleep stages from the Sleep-EDF dataset. We used preprocessing to convert the EEG signals to an equal 100 Hz sampling frequency, segmented into 30-second epochs, and extracted features both for conventional machine learning and for deep learning model inputs. We compared various classifiers namely Random Forest, XGBoost, a fully-connected neural network, LSTM, RNN, and Bi-LSTM showing that our deep learning methods are able to extract the temporal relations inherent in EEG signals properly. Our experimental findings show that conventional machine learning approaches, such as Random Forest and XGBoost, obtained F1-scores of 0.8229 to 0.8433, whereas our deep learning models, especially the Bi-LSTM design, obtained an F1-score of 0.8030. Comparing our findings with recent state-of-the-art approaches such as SleepEEGNet, CNN-Transformer, TinySleepNet, XSleepNet, and RobustSleepNet, our results compare favorably and are even better in a few instances. Such comparisons highlight the viability of our framework in accomplishing strong sleep stage classification with little computational burden. In general, our results illustrate that an end-to-end deep learning system—properly optimized and crafted—can efficiently enhance sleep stage automation and assist with more streamlined clinical diagnosis and monitoring of sleep disease. The direction of future work will include better classifying problematic stages (e.g., N1) and extending the method to process multi-modal signals in order to facilitate more generalized application to diverse populations of patients.

IX. Reference

References

[1] S. Mousavi, F. Afghah, and U. R. Acharya, "SleepEEGNet: Automated Sleep Stage Scoring with Sequence to Sequence Deep Learning Approach," *PLOS ONE*, vol. 14, no. 5, e0216456, 2019.

[2] Z. Yao and X. Liu, "A CNN-Transformer Deep Learning Model for Real-time Sleep Stage Classification in an Energy-Constrained Wireless Device," arXiv:2211.13005, 2022.

[3] A. Supratak, H. Dong, C. Wu, and Y. Guo, "TinySleepNet: An Efficient Deep Learning Model for Sleep Stage Scoring Based on Raw Single-Channel EEG," in *Proc. IEEE Engineering in Medicine and Biology Society (EMBC)*, 2020, pp. 641–644.

[4] H. Phan, O. Y. Chén, P. Koch, A. Mertins, and M. De Vos, "XSleepNet: Multi-View Sequential Model for Automatic Sleep Staging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 473–485, 2021.

[5] A. Guillot and V. Thorey, "RobustSleepNet: Transfer Learning for Automated Sleep Staging at Scale," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 1441–1451, 2021.

[6] H. Alsolai, S. Qureshi, S. M. Z. Iqbal, et al., "A Systematic Review of Literature on Automated Sleep Scoring," *IEEE Access*, vol. 10, pp. 79419–79443, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.319414510.1109/ACCESS.2022.3194145.

[7] O. Faust, H. Razaghi, R. Barika, et al., "A review of automated sleep stage scoring based on physiological signals for the new millennia," *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 81–91, 2019. DOI: https://doi.org/10.1016/j.cmpb.2019.04.03210.1016/j.cmpb.2019.04.032.

[8] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders," *Annals of Biomedical Engineering*, vol. 44, no. 5, pp. 1587–1597, 2016. DOI: https://doi.org/10.1007/s10439-015-1444-y10.1007/s10439-015-1444-y.

[9] M. Ronzhina, O. Janoušek, J. Kolářová, et al., "Sleep scoring using artificial neural networks," *Sleep Medicine Reviews*, vol. 16, no. 3, pp. 251–263, 2012. DOI: https://doi.org/10.1016/j.smrv.2011.06.00310.1016/j.smrv.2011.06.003.

[10] S. Biswal, H. Sun, B. Goparaju, et al., "Expert-level sleep scoring with deep neural networks," *Journal of the American Medical Informatics Association*, vol. 25, no. 12, pp. 1643–1650, 2018. DOI: https://doi.org/10.1093/jamia/ocy13110.1093/jamia/ocy131.

[11] M. Dutt, S. Redhu, M. Goodwin, and C. W. Omlin, "SleepXAI: An explainable deep learning approach for multi-class sleep stage identification," *Applied Intelligence*, vol. 53, no. 13, pp. 16830–16843, 2023. DOI: https://doi.org/10.1007/s10489-022-04357-810.1007/s10489-022-04357-8.

[12] H. Han, M. J. Seong, J. Hyeon, et al., "Classification and automatic scoring of arousal intensity during sleep stages using machine learning," *Scientific Reports*, vol. 14, no. 1, 2024. DOI: https://doi.org/10.1038/s41598-023-50653-910.1038/s41598-023-50653-9.

[13] M. Gaiduk, Á. Serrano Alarcón, R. Seepold, et al., "Current status and prospects of automatic sleep stages scoring: Review," *Biomedical Engineering Letters*, vol. 13, no. 3, pp. 247–272, 2023. DOI: https://doi.org/10.1007/s13534-023-00299-310.1007/s13534-023-00299-3.

[14] S. Mousavi, F. Afghah, and U. R. Acharya, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLOS ONE*, vol. 14, no. 5, 2019. DOI: https://doi.org/10.1371/JOURNAL.PONE.021645610.1371/JOURNAL.PONE.0216456.

[15] C. Zhao, W. Wu, H. Zhang, et al., "Sleep Stage Classification Via Multi-View Based Self-Supervised Contrastive Learning of EEG," *IEEE Journal of Biomedical and Health Informatics*, 2024. DOI: https://doi.org/10.1109/JBHI.2024.343263310.1109/JBHI.2024.3432633.

[16] Y. Chen, Y. Lv, X. Sun, et al., "ESSN: An Efficient Sleep Sequence Network for Automatic Sleep Staging," *IEEE Journal of Biomedical and Health Informatics*, 2024. DOI: https://doi.org/10.1109/JBHI.2024.344334010.1109/JBHI.2024.3443340.

[17] C. Jiang, W. Xie, J. Zheng, et al., "MLS-Net: An Automatic Sleep Stage Classifier Utilizing Multimodal Physiological Signals in Mice," *Biosensors*, vol. 14, no. 8, 2024. DOI: https://doi.org/10.3390/bios1408040610.3390/bios14080406.

[18] M. He, M. Tang, L. Meng, and Z. Liang, "TBSTSleepNet: Three-branch spectro-temporal bidirectional LSTM based attention model for EEG sleep staging," *Biomedical Signal Processing and Control*, vol. 97, 2024. DOI: https://doi.org/10.1016/j.bspc.2024.10669510.1016/j.bspc.2024.106695.

[19] W. Zhang, S. Zhang, Y. Wang, et al., "A CNN-Transformer-ConvLSTM-CRF Hybrid Network for Sleep Stage Classification," *IEEE Sensors Journal*, 2024. DOI: https://doi.org/10.1109/JSEN.2024.343440410.1109/JSEN.2024.3434404.

[20] A. Ito and T. Tanaka, "SleepSatelightFTC: A Lightweight and Interpretable Deep Learning Model for Single-Channel EEG-Based Sleep Stage Classification," 2024. DOI: https://doi.org/10.1101/2024.08.02.60630110.1101/2024.08.02.606301.