Mice Sleep Staging from EEG and EMG signals

Fazla Rabbi Mashrur[1] , Yixuan Luo[2] , Paul Berggren[3] , Yanlong Zhao[4]

Abstract— **Automatic sleep staging using electroencephalography (EEG) and electromyography (EMG) data is a critical component of sleep-related research. Current sleep staging approaches have two significant flaws: limited information exchanges across modalities and a lack of unified models capable of processing many input sources. To address these difficulties, we compare three methods to autonomous sleep staging: a CNN-based model, a transformer-based model, and an attention-based model. Experiments demonstrate that the CNN-based approach achieves an average accuracy of 85.81%. When a self-attention mechanism was added, the average accuracy was increased to 86.89%. The transformer and attention-based models also show promising results. Future research should focus on improving the classification performance for challenging sleep stages, such as REM, to enhance the overall utility of sleep-related research.**

I. CONTRIBUTION

Fazla- *Code*: Preprocesing, CNN arcitecture, Evaluation, *Report*: Material: B, C; Experiments: A, B(1)
YiXuan- *Code*: Transformer-based model, Evalutation *Report*: Introduction; Methods: C; Experiment: B(3);
Yanlong- *Code*: Literature replication *Report*: Materials: A, C, D
Paul- *Code*: Attention+CNN model *Report*: Methods: A, B; Experiment: B(2)

II. INTRODUCTION

Sleep, an essential animal behavior, plays a critical role in maintaining various physiological and cognitive functions. The precise classification of sleep stages—namely wakefulness, REM, and non-REM—is pivotal for advancing our understanding in sleep-related research [1]. The traditional methodologies in this domain, however, are exceedingly labor-intensive and necessitate a high level of specialized knowledge, strongly underscoring the pressing need for automated systems [2], [3]. Nonetheless, the automated approaches currently available encounter substantial obstacles. These include the necessity for improved temporal resolutions, with data granularity improving from intervals ranging between 4 to 10 seconds, and the integration challenges posed by the absence of comprehensive models capable of assimilating heterogeneous data sources effectively. Addressing these limitations is crucial not only for enhancing the effectiveness and efficiency of these systems but also for transforming how research is conducted in this field, thereby deepening our insight into the complexities of sleep dynamics.

In recent years, deep learning-based methods have demonstrated considerable potential in classification and prediction tasks. These advanced technologies, when applied to deep learning, achieve state-of-the-art results in various fields, including sleep staging [4], [5]. Most existing research utilizes electroencephalography (EEG) time series signals for predictive analysis. These signals are instrumental in classifying sleep into stages such as wakefulness, REM sleep, and non-REM sleep.

In this study, our aim is to harness the power of machine learning technologies to forecast sleeping stages using signal data from EGG and EMG. By employing advanced machine learning techniques, we anticipate achieving significantly enhanced accuracy compared to traditional methods.

III. MATERIALS

A. Literature Review

Previous studies have explored various automated sleep scoring methods for mice, introducing models such as those detailed by Schwabedal et al. [6], which focus on the automation of sleep stage classification and EEG artifact detection using deep neural networks. These approaches address the labor-intensive nature of manual sleep scoring in research settings. In this context, our work introduces SlumberNet, a novel approach leveraging a Residual Neural Network (ResNet) architecture tailored for the classification of sleep stages in mice through the analysis of EEG and EMG data [7]. Notably, SlumberNet not only advances the methodology in sleep stage classification but also establishes a relevant connection between the sleep patterns observed in mice and potential implications for human sleep studies. This linkage underscores the importance of our model in setting a foundation for future optimizations aimed at human sleep analysis. Additionally, the implementation of DeepSleepNet [8] aligns with the standards set forth in the AASM manual, further validating the effectiveness of deep learning models in capturing critical EEG patterns indicative of distinct sleep stages.

B. Data Collection

This section describes the processes used to obtain data for mouse sleep staging. The linked dataset is freely accessible and contains sleep data from mice [2]. The researchers manually labelled this dataset with sleep stages. All sleep

[1]F. R. Mashrur is with Biomedical Engineering, University of Rochester, USA
[2]Yixuan Luo is with Electrical and Computer Engineering, University of Rochester, USA
[3]Paul Berggren is with Electrical and Computer Engineering, University of Rochester, USA
[4]Yanlong Zhao is with Electrical and Computer Engineering, University of Rochester, USA

monitoring data were collected from C57BL/6 mice, including both male and female animals. The researchers gathered EEG and EMG data while the subjects. To improve the quality of the acquired EEG signals, scientists used filters such as a band-pass filter at 1 to100 Hz. For EMG signals, a 10 to 250 Hz bandpass filter is used. Additionally, a 50 Hz notch filter was used to remove power line noise. For a more detailed description of the dataset, please refer to V.

C. dataset in SlumberNet

The dataset utilized in this study comprises electroencephalogram (EEG) and electromyogram (EMG) signals collected from mice. The age range of the mice was 9-11 weeks, and the data was recorded over a period of 6-7 days at a sampling rate of 500 Hz. The collection phases included baseline sleep, sleep deprivation, and recovery sleep, ensuring a comprehensive dataset that reflects a wide range of sleep behaviors.

Data preparation was meticulous, involving the selection of files containing paired raw voltage data alongside identified sleep stages. The preprocessing steps were designed to enhance the quality and usability of the data for model training. This included the removal of artifacts and the downsampling of the signal to manage the dataset size and improve computational efficiency. The cleaned data was then segmented into 4-second epochs and categorized into one of three sleep stages: Wake, NREM, and REM. Epochs labeled as artifacts were systematically excluded from the training process to maintain the integrity and accuracy of the model training.

The extensive nature of this dataset provides a robust foundation for developing and optimizing SlumberNet, potentially enhancing its predictive accuracy and generalizability across different sleep stages. Also, may help us to expand our dataset.

D. Implement of SlumberNet

We have implemented the SlumberNet model as described in the referenced paper. SlumberNet is a residual neural network (ResNet) designed for classifying sleep stages using electroencephalogram (EEG) and electromyogram (EMG) signals. Originally developed to analyze sleep conditions in mice, this model has demonstrated high accuracy and efficiency in sleep stage classification, significantly reducing the time required for manual analysis. We have adopted SlumberNet as the baseline model for our comparative analysis due to its robust performance and advanced deep learning architecture.

The training process are shown as the1

Results demonstrate that DeepSleepNet can automatically learn features for sleep stage scoring from different raw single-channel EEG datasets with varying properties and scoring standards, achieving comparable performance to state-of-the-art methods without altering the model architecture or training algorithm. Based on the confusion matrix in Fig2, we get:

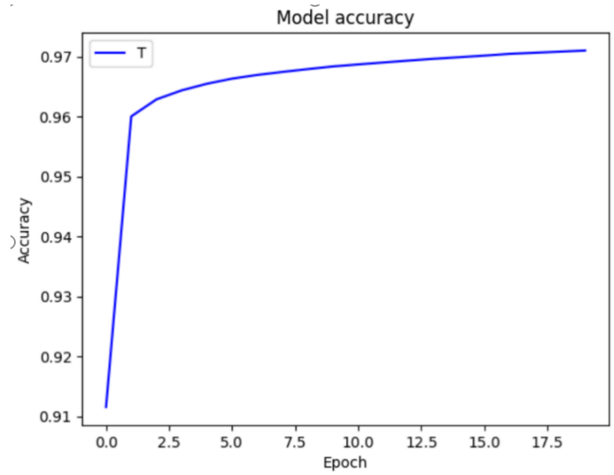


Fig. 1. Training Accuracy vs. epochs

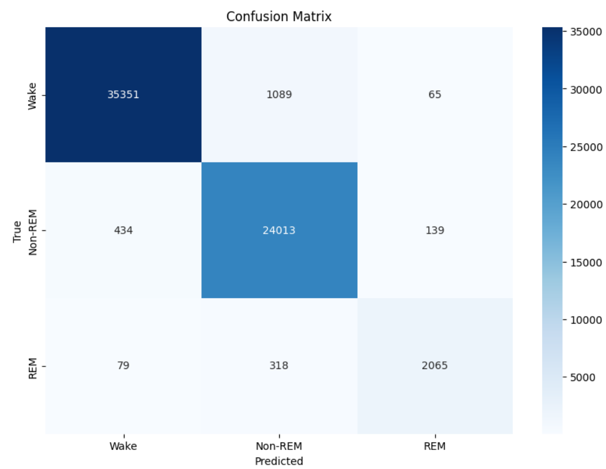


Fig. 2. Confusion Matrix Based on SlumberNet

- **Wake Stage:** Highly accurate with few misclassifications.
- **Non-REM Stage:** Reasonably accurate but shows some confusion with the REM stage.
- **REM Stage:** The model struggles here, with notable misclassifications as Non-REM.

E. Data Prepossessing

The dataset primarily comprises EEG and EMG signal recordings from sleeping mice, accompanied by sleep stage labels assigned to every second. However, these raw data exhibit data. To address these issues, data preprocessing techniques are employed such as normalization, and segmentation. Throughout the entirety of mice sleep, there are time periods where experts are uncertain about the corresponding sleep stage. These unlabeled periods introduce data irregularities. However, since missing values occur in only a small percentage of the total data, we do not consider these data for our work. Lastly, to achieve input data with a fixed time span, we perform temporal slicing. Since our model aims to provide accurate predictions at the second level, we slice the input signals into per-second segments. Both EEG and EMG signals have an equal sampling frequency, which we denote as T.

IV. Methods

A. Convolutional Neural Network

In this section, we describe the settings and architecture of our multi-modal 1D CNN model. The model comprises four separate 1D convolutional branches, each processing a different type of signal: EEG and EMG, as well as their FFT-transformed counterparts.

Each branch is constructed using several 1D convolutional layers combined with activation functions, normalization, and pooling layers. The branches share a common convolutional block structure, defined as follows:

1D Convolutional Layer: Applies a 1D convolution to capture temporal features. PReLU Activation: Introduces non-linearity with a parametric ReLU activation function. Batch Normalization: Normalizes the feature maps to improve training stability. Dropout (Optional): Adds regularization to prevent overfitting if specified.

Each branch contains several of these convolutional blocks, which are followed by average pooling, flattening, dropouts, linear layers, and lastly a normalization of the activation layer.The classifier combines outputs from all four branches using a fully connected layer to yield the final classification. This classifier includes a pair of linear layer+ activation layer blocks, resulting in the final output.

B. Attention-based Network

In the attention-based model, we expanded on the CNN architecture described in the earlier section. This means that all of the layer order and parameters were not changed, apart from the newly added layers. A self-attention approach was taken, meaning that the signal is used as both the data and the context. Furthermore, we applied a multi-headed approach in order to allow the model to learn from multiple different feature spaces at the same time. The number of heads and embeddings per head was tweaked during testing for the EEG and EMG signals as well as their FFT counterparts. The attention algorithm that was implemented was quite standard, with no filtering applied. This was because it was okay for the preceding and following stages to influence the current stage. The attention modules were applied for each of these signals, with the hope that a more contextualized understanding would allow for better performance. Specifically, the attention modules were placed after each of the 1-dimensional convolution layers. This was done in order to allow the influences to be interconnected, providing a more global context window for the model.

C. Transformer-based Network

Here, we designed a transform-based model to harness the capabilities of the widely acclaimed transformer architecture in machine learning. Our model employs two 6-layer transformer encoder blocks, one for EEG signals and the other for EMG signals. These encoders are designed to extract

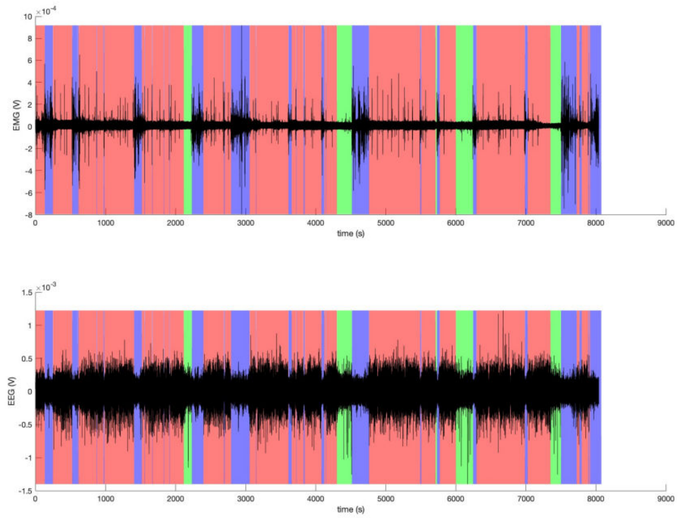
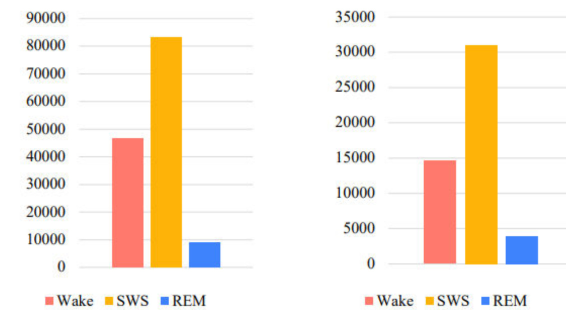


Fig. 3. Example of collected raw data: Two graphs illustrating EMG and EEG data, respectively. Colored backgrounds denote sleep stage labels



(a) Bar plot of the training set (b) Bar plot of the test set

Fig. 4. Illustration of class distribution for one LOSO a. traning b.testing

features—or representations—from each input signal. The extracted features from both signals are then concatenated into a single vector that represents the overall feature landscape. On the decoder side, we opted for a simple Multilayer Perceptron (MLP) due to the computation limitations. This MLP processes the concatenated feature vector and outputs a logit vector, which serves as the prediction feedback. The final prediction is determined by selecting the entry with the highest value from the logit vector, which represents our predicted sleep stage.

V. Experiments

A. Setup

The collection includes EEG and EMG data from eight mouse, each lasting around four hours. Given that the EEG and EMG signals are captured at 512 Hz, each mouse record might include around 5 million EEG/EMG data points. To match the expert labelling span, we set the epoch window to one second, yielding a total of 11,0000 epoch. For model evaluation, we employed a leave-one-out (LOSO) criteria, using records from 7 mice subjects used for training and 1

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
Awake	0.8414	0.1586	0.9321	0.0679	0.8844
NREM	0.9177	0.0823	0.8518	0.1482	0.8835
REM	0.4629	0.5371	0.5413	0.4587	0.4990

Fig. 5. Detailed results for CNN-based approach.

TARGET / OUTPUT	Awake	NREM	REM	SUM
Awake	32895 29.57%	6073 5.46%	127 0.11%	39095 84.14% 15.86%
NREM	2258 2.03%	58863 52.91%	3019 2.71%	64140 91.77% 8.23%
REM	138 0.12%	4169 3.75%	3712 3.34%	8019 46.29% 53.71%
SUM	35291 93.21% 6.79%	69105 85.18% 14.82%	6858 54.13% 45.87%	95470 / 111254 85.81% 14.19%

Fig. 6. Confusion matrix for CNN-based approach.

mouse subject used for testing. The distribution of stages in the training and test sets may be seen in 4, respectively.

B. Implementation & Results

1) CNN-based approach: The experiments were conducted using the PyTorch deep learning framework on an NVIDIA Tesla K80 GPU. The model was trained with a batch size of 64, using the Adam optimizer with a learning rate of 0.01, and a OneCycleLR learning rate scheduler. The training process consisted of 30 epochs, utilizing CrossEntropyLoss as the loss function. Performance metrics such as accuracy, F1 score, precision, recall, and the misclassification rate were computed. Additionally, a confusion matrix was employed to assess classification performance, and the best model was selected based on the highest F1 score. The CNN-based model achieved an average accuracy of 85.5%. Detailed results and confusion matix for all the subject for each class, along with aggregate metrics, are presented in Fig and Fig 6 repectively.

2) Attention+CNN Approach: Similarly to the prior section, all of the results gathered were from the Pytorch deep learning framework run on an NVIDIA A100 GPU. In order to aid comparisons between the two approaches, the batch size and learning rate were kept the same as well. The model performance was monitored with metrics such as accuracy, F1 score, precision, recall, and the misclassification rate, derived from a confusion matrix. The highest performing version of this model achieved an accuracy of 84.65%, which

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
Awake	0.7725	0.2275	0.8558	0.1442	0.8120
NREM	0.9064	0.0936	0.8087	0.1913	0.8548
REM	0.2667	0.7333	0.4084	0.5916	0.3227

Fig. 7. Detailed results for Attention+CNN-based approach.

Training Set				
TARGET / OUTPUT	Awake	NREM	REM	SUM
Awake	30202 27.14%	7219 6.49%	1674 1.50%	39095 77.25% 22.75%
NREM	3389 3.05%	55895 50.24%	2383 2.14%	61667 90.64% 9.36%
REM	1700 1.53%	6001 5.39%	2801 2.52%	10502 26.67% 73.33%
SUM	35291 85.58% 14.42%	69115 80.87% 19.13%	6858 40.84% 59.16%	88898 / 111264 79.90% 20.10%

Fig. 8. Confusion matrix for Attention+CNN-based approach.

shows a slight decrease from the pure CNN model. The confusion matrix shows how the precision on the REM staging decreased significantly. This is most likely the cause behind the decrease in accuracy, as the model is worse at generalizing due to the class-imbalance and small amount of data. The exact metrics are presented in Fig and Fig 8 respectively.

3) Transformer-based Network: All experiments were conducted using the PyTorch deep learning framework on an NVIDIA A100 GPU. The model was trained with a batch size of 512, employing the SGD optimizer with a learning rate of 0.1, complemented by a OneCycleLR learning rate scheduler. The training regimen spanned 30 epochs, with CrossEntropyLoss serving as the loss function. Performance metrics including accuracy, precision, recall, F1 score, and specificity were calculated to evaluate the model. Additionally, a confusion matrix was utilized to assess classification performance thoroughly. The model achieving the highest F1 score was selected as the best model.

The Transformer-based model reached an average accuracy of 76.05%, slightly below the traditional CNN-based model discussed earlier. This lower performance could be attributed to the limited number of data samples available. As a data-intensive model, Transformers require a substantial amount of data to effectively train their feature extraction patterns.

TABLE I

ACCURACY(%) OF THE IMPLEMENTED MODELS FOR EACH MOUSE IN LOSO EVALUATION

Mice	CNN	CNN + Attention	Transformer
1	90.92	88.60	77.88
2	82.3	80.23	85.20
3	87.97	86.92	62.25
4	83.23	83.02	51.42
5	89.83	88.95	80.37
6	91.2	90.23	84.29
7	71.73	78.94	82.27
8	84.79	80.32	84.76
Avg	85.5	84.65	76.05

VI. CONCLUSION

Overall, this paper detailed a few different approaches to sleep staging in mice, including CNN, CNN + attention, and transformer-based models. These models aim to alleviate the high labor cost of manually labeling the sleep stages of mice, specifically at a granularity of 1 second. The explored CNN model preformed the best out of those explored, achieving an accuracy of 85.5%. A limitation which most likely held back the attention and transformer-based models was the dataset size, as these approaches tend to be very data hungry. Thus, a future work may simply include training the model on a larger dataset of mice, while also synthetically generating REM sleep data to overcome the class-imbalance issue.

REFERENCES

[1] J. B. Stephansen, A. N. Olesen, M. Olsen, A. Ambati, E. B. Leary, H. E. Moore, O. Carrillo, L. Lin, F. Han, H. Yan *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature communications*, vol. 9, no. 1, p. 5229, 2018.

[2] C. Kjaerby, M. Andersen, N. Hauglund, V. Untiet, C. Dall, B. Sigurdsson, F. Ding, J. Feng, Y. Li, P. Weikop *et al.*, "Memory-enhancing properties of sleep depend on the oscillatory amplitude of norepinephrine," *Nature neuroscience*, vol. 25, no. 8, pp. 1059–1070, 2022.

[3] J. Chen, Y. Yao, M. Anderson, N. Hauglund, C. Kjaerby, V. Untiet, M. Nedergaard, and J. Luo, "sdreamer: Self-distilled mixture-of-modality-experts transformer for automatic sleep staging," in *2023 IEEE International Conference on Digital Health (ICDH)*, 2023, pp. 131–142.

[4] S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. B. Westover, M. T. Bianchi, and J. Sun, "Sleepnet: automated sleep staging system via deep learning," *arXiv preprint arXiv:1707.08262*, 2017.

[5] J. Huang, L. Ren, X. Zhou, and K. Yan, "An improved neural network based on senet for sleep stage classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 10, pp. 4948–4956, 2022.

[6] J. T. Schwabedal, D. Sippel, M. D. Brandt, and S. Bialonski, "Automated classification of sleep stages and eeg artifacts in mice with deep learning," *arXiv preprint arXiv:1809.08443*, 2018.

[7] P. K. Jha, U. K. Valekunja, and A. B. Reddy, "Slumbernet: deep learning classification of sleep stages using residual neural networks," *Scientific Reports*, vol. 14, no. 1, p. 4797, 2024.

[8] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, p. 1998–2008, Nov. 2017. [Online]. Available: http://dx.doi.org/10.1109/TNSRE.2017.2721116