A CNN-Transformer Deep Learning Model for Real-time Sleep Stage Classification in an Energy-Constrained Wireless Device

Zongyan Yao
University of Toronto
Toronto, Canada
zongyan.yao@mail.utoronto.ca

Xilin Liu
University of Toronto
Toronto, Canada
xilinliu@ece.utoronto.ca

***Abstract*—This paper proposes a deep learning (DL) model for automatic sleep stage classification based on single-channel EEG data. The DL model features a convolutional neural network (CNN) and transformers. The model was designed to run on energy and memory-constrained devices for real-time operation with local processing. The Fpz-Cz EEG signals from a publicly available Sleep-EDF dataset are used to train and test the model. Four convolutional filter layers were used to extract features and reduce the data dimension. Then, transformers were utilized to learn the time-variant features of the data. To improve performance, we also implemented a subject specific training before the inference (i.e., prediction) stage. With the subject specific training, the F1 score was 0.91, 0.37, 0.84, 0.877, and 0.73 for wake, N1- N3, and rapid eye movement (REM) stages, respectively. The performance of the model was comparable to the state-of-the-art works with significantly greater computational costs. We tested a reduced-sized version of the proposed model on a low-cost Arduino Nano 33 BLE board and it was fully functional and accurate. In the future, a fully integrated wireless EEG sensor with edge DL will be developed for sleep research in pre-clinical and clinical experiments, such as real-time sleep modulation.**

I. Introduction

Sleep quality and health are closely related; therefore, it is important to understand one's sleep quality to improve health condition. A measurement of sleep quality is the time spent in each sleep stage. There are five sleep stages, which are wake, N1, N2, N3, and REM, with each stage progressively deeper sleep. Most of the sleep occurs between stages N1 and N3 [1]. The clinical evaluation of sleep stages is performed by polysomnogram (PSG), a procedure that records one's electroencephalogram (EEG), electrooculogram (ECG), and other physiological features. Medical professionals will manually classify their sleep stages over time according to one or more of the features mentioned above.

With the development of electronic technology and machine intelligence, wearable devices, such as smartwatches, can measure user biosignals and potentially classify their sleep stages. However, the cost of these devices is high and the performance of sleep stage classification is limited. In addition, classification often requires the transmission of data to mobile phones or the cloud, raising concerns about cybersecurity [2].

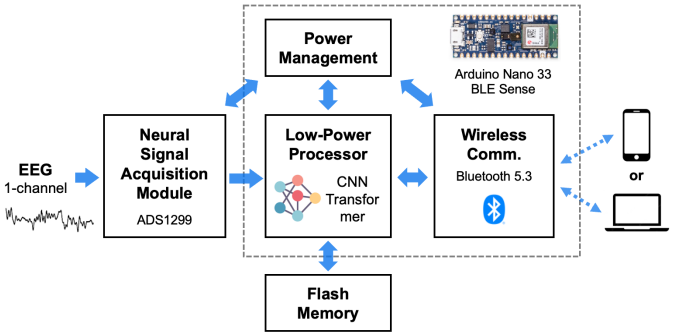


Fig. 1. An overview of the envisioned wireless device for real-time sleep stage classification using edge DL. This paper focuses on the development and deployment of the DL model.

High-quality real-time sleep classification and sleep modulation still needs to be performed in sleep laboratories. There is a need for low-cost at-home sleep monitoring devices that can perform sleep stage classification on device, and potentially use the sleep stage to generate auditory stimulation for treating sleep disorders or enhancing sleep quality [3].

In this paper, we develop a lightweight DL model for running on devices with restricted energy and memory, such as microcontrollers [4]. There are two main constraints to the development of the model for hardware. The first constraint is that the size of the model will be limited by the memory resources available on the hardware, including the non-volatile Flash memory for model storage and the random-access memory (RAM) for model computing. The second constraint is that the computational demand of the model will be limited by the clock rate, bit width, and computational capabilities (such as floating point or fixed point) of the device. Key trade-offs are between model performance and complexity. In this work, we developed a DL model that can run on a low-power wireless microcontroller, based on a low-cost Arduino Nano 33 BLE development board. Despite its small size, our model achieved performances comparable to the state-of-the-art during a validation using a publicly available sleep dataset.

Fig. 1 shows the overall block diagram of the envisioned

fully integrated sleep stage classification device featuring the developed DL model. The device will be miniature in size and fully self-contained. It can enable a wide range of sleep research in pre-clinical and clinical studies.

II. Methods

A. Dataset

Sleep-EDF Expanded Database (version 1, published in 2013) contains 197 whole-night Polysomnographic (PSG) sleep recordings [5], [6]. It contains two subsets, the Sleep Cassette Study (SC) and Sleep Telemetry Study (ST). Data from the Sleep Telemetry Study was obtained in 1994 to study the effects of temazepam on sleep. Since we are proposing a model to classify the stages of sleep of healthy people, we will not use data from this subset. Our experiments will be performed on the SC subset.

Two 20-hour PSG recordings were taken for 77 subjects between the age of 25 and 101. The first nights of subjects 36 and 52, and the second night of subject 13 were lost. The PSG recordings contain three channels of EEG signals, one channel of EOG and chin EMG signals, oronasal airflow, rectal body temperature, and event marker. To reduce our model's size, we only use Fpz-Cz EEG signals as input to our model. The EEG signals were sampled at 100 Hz. Each of the 30-second segments of the signals was labeled by well-trained sleep experts. There are eight stages, N1, N2, N3, N4, Wake, REM, MOVEMENT, UNKNOWN). To make our results consistent and comparable with previous studies [7]–[9], we preprocessed the data with the following methods:

1) Discarded the segments with UNKNOWN and MOVEMENT labels.
2) Combined N4 and N3 together as N3 stage.
3) Ignored wake epochs longer than 30 minutes outside of sleep periods.

TABLE I
DATA DISTRIBUTION

	Wake	N1	N2	N3	REM
of Segment	44752	15793	54682	12268	20976
Distribution	30.14%	10.64%	36.83%	8.26%	14.13%
Total Number	148471				

B. Performance Metrics

We evaluated our model's performance using per-class Precision (PR), per-class Recall (RE), per-class F1-score (F1), and overall accuracy (acc). Overall accuracy is the ratio between the number of correct predictions and the population. For a category prediction, there are four outcomes: true positive (TP), false positive (FP), true negative (TN), false negative (FP). Metrics are defined as:

$$PR = \frac{TP}{TP + FP} \tag{1}$$

$$RE = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2TP}{2TP + FN + FP} \tag{3}$$

Overall accuracy is commonly used to measure classification performance. However, for an imbalanced dataset, precision does not provide adequate information on classifiers, because it hardly reveals performance in minority groups [10]. Table I shows the distribution of the dataset. The dataset is highly imbalanced, so we introduced additional metrics, PR, RE, and F1, to correctly measure the performance of our model.

C. Proposed Model

Raw EEG data contains time-invariant and time-variant features. Each 30-second input data segment with 100-Hz sampling frequency, so the input shape is (3000,1). It is too large to feed it into a transformer unit. Fig. 2 shows our model's architecture. A convolutional neural network can extract time-invariant data and output smaller data. We implemented four sequential convolutional layers to output features with shape (19,128). Then we use a transformer unit to learn some time-variant information from the features. Its attention mechanism learned the contexts on all positions of the time series data. The two dense layers inside the transformer unit work as an encoder. The output of the encoder is then added to the input data for additional features. Finally, to correctly classify sleep scores, we used a dense layer with a softmax activation function to obtain the most possible categories. We also tried other models, such as recurrent neural network and auto-encoders. Experiments showed that our proposed model yielded the best performance.

III. Experiments

A. Data Preprocessing

Since our model was designed to be deployed on microcontrollers with limited memory and computation resources, we cannot design a complex data preprocessing method, so we implemented a simple standarization. The performance of deep-learning models is highly dependent on the statistical properties of the input data. If the input data is too small or too large, the models perform poorly. The EEG data in the Fpz-Cz channel is within the scope of 10^{-5}, which is too small for our model. Standarization transforms data to have a zero mean and a standard deviation of one. For each sample X, we standardized it to Z using

$$Z = \frac{X - M}{S}$$

where M indicates the mean of the samples and S indicates the standard deviation of the samples. After standarization, the input data was in the range of 10^{-1} to 10^{1}, and the model showed the best performance.

B. Basic Training

We used 5-fold cross-validation to train and test our model. There are 77 subjects in total, and each fold contains 16 subjects' data (one fold has 13 subjects). In each iteration, we selected four folds as training and validating data, and

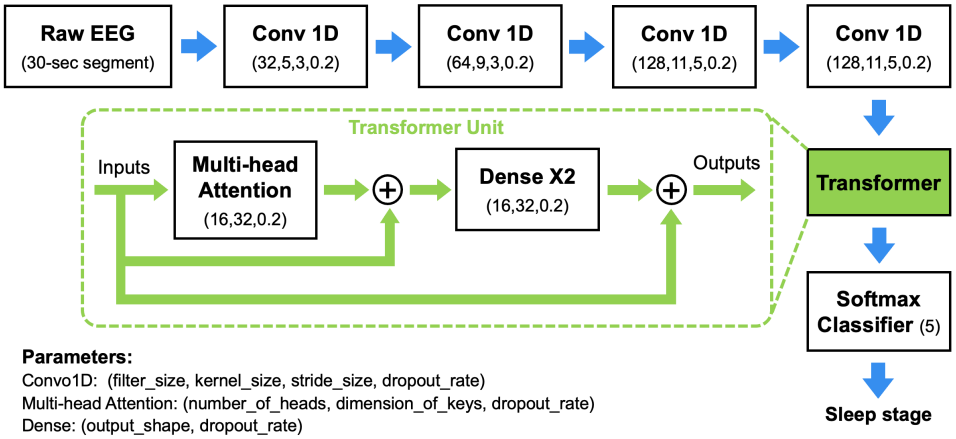


Fig. 2. Architecture of the proposed CNN-Transformer DL model for real-time sleep stage classification using single channel EEG signal.

one fold as testing data. Among the four folds of data, we randomly sampled 10% for validation. The remaining 90% of data was for training. We used Adam algorithm for optimization, which is a stochastic gradient descent method based on adaptive estimation of first-order and second-order moments. It computes efficiently and requires little memory [11]. We utilized categorical cross-entropy as the loss function and fed our model with a batch size of 64 samples.

C. Subject-Specific Training

We also performed subject-specific training in the testing stage to further adapt the patterns of each subject. We randomly selected 10% of the test data and fed them to our trained model. The remaining 90% of the test data were used to test our models after subject-specific training.

IV. Results

Table IV and Table III show the confusion table and the performance of our model, respectively. Since the dataset is imbalanced, the per-class performance in N1 and REM was expected to be poorer than the majority classes. The model performed well in the Wake and N2 classes. The state-of-the-art performances from literature are also similar. From Table VI, all models have an F1 score per class less than 0.5 in N1 and less than 0.8 in REM. To improve the performance of our model, we perform a subject-specific training to further adapt subject-wise patterns. Table IV and Table V show that performances improved after subject-specific training. Firstly, the overall accuracy increased from 0.775 to 0.795. Secondly, per-class precision, recall, and F1 score increased for all classes.

Table VI compares the F1 score of different models from the literature. Since our proposed model is lightweight and small, it cannot yield the best performance. However, it still had performance comparable to that of the state-of-the-art. For the wake stage, we yielded the highest F1-score of 0.91. For other classes, we were close to the highest. There is no model that could beat our performances in all classes. Some models performed better on certain classes.

TABLE II
CONFUSION MATRIX BEFORE SUBJECT-SPECIFIC TRAINING

Actual/Predict	Wake	N1	N2	N3	REM
Wake	0.90	0.04	0.01	0	0.04
N1	0.20	0.26	0.31	0.01	0.21
N2	0.01	0.04	0.82	0.07	0.06
N3	0	0	0.19	0.80	0
REM	0.05	0.08	0.12	0	0.74

TABLE III
PERFORMANCE BEFORE SUBJECT-SPECIFIC TRAINING

	Wake	N1	N2	N3	REM
Precision	0.90	0.26	0.72	0.80	0.74
Recall	0.89	0.42	0.81	0.70	0.64
F1-Score	0.90	0.32	0.82	0.75	0.69
Accuracy	0.775				

TABLE IV
CONFUSION MATRIX AFTER SUBJECT-SPECIFIC TRAINING

Actual/Predict	Wake	N1	N2	N3	REM
Wake	0.91	0.05	0.01	0	0.02
N1	0.21	0.31	0.29	0.01	0.18
N2	0.01	0.04	0.84	0.06	0.05
N3	0	0	0.20	0.80	0
REM	0.05	0.08	0.10	0	0.78

TABLE V
PERFORMANCE AFTER PATIENT-SPECIFIC TRAINING

	Wake	N1	N2	N3	REM
Precision	0.92	0.31	0.84	0.80	0.78
Recall	0.90	0.46	0.83	0.74	0.69
F1-Score	0.91	0.37	0.84	0.77	0.73
Accuracy	0.795				

V. Discussion

Our lightweight model was designed for memory-constrained microcontroller units. It had around 300,000 pa-

TABLE VI
F1-SCORE COMPARISON WITH STATE-OF-ART

Reference	Year	Architecture	F1-Score					Accuracy
			Wake	N1	N2	N3	REM	
[12]	2022	DSSNet	0.86	0.20	0.87	0.87	0.71	0.82
[13]	2018	1-Max-CNN	0.77	0.33	0.87	0.86	0.76	0.80
[7]	2017	DeepSleepNet	0.88	0.37	0.83	0.77	0.80	0.80
[14]	2019	SleepEEGNet	0.91	0.44	0.82	0.73	0.76	0.80
[15]	2021	CNN	0.91	0.42	0.77	0.66	0.69	N/A
This work			0.91	0.37	0.84	0.77	0.73	0.80

rameters, and its size was 2 MB. We used an Arduino Nano 33 BLE board, which integrates a wireless microntroller (nRF52840, Nordic Semiconductor) with a 64 MHz 32-bit ARM CPU, 1 MB of flash memory, and 256 KB of SRAM [16]. The deployed model was stored in the flash memory, so the model was supposed to be less than 1 MB. We planned to add an external memory unit to the model in the future. To test the availability of our design in the current stage, we implemented a smaller version of the model and deployed it on the Arduino board. The model had an accuracy of 68%, but the microcontroller was fully functional for target classification. To reduce the size of our model, we also quantized the model in the deployment stage.

We randomly selected 10% of the test data set to perform subject-specific training. If a dataset is large, the randomly selected data should follow the distribution of the dataset. In our experiments, the distribution varied as the dataset was not large enough. The improvement of performance was highly dependent on the distribution of the selected data. Therefore, we will enforce the distribution of the subject-specific training data. This means that the number of selected data in each category is calculated and fixed based on the test dataset. This method would maximize the benefits of subject-specific training.

As mentioned previously, the dataset has an imbalanced class distribution, which significantly affects models' performance, especially on the minority categories. There are different methods to mitigate the effect of imbalance data, such as oversampling and undersampling. Oversampling is used to duplicate samples in minority classes, while undersampling is used to remove samples from the majority classes. There are two common ways in training imbalanced data. Another method we will try in the future is to add weights to the loss function. The loss function can be weighted differently for different classes, so that minority classes are learned more.

VI. CONCLUSION

In this work, we developed a lightweight DL model for real-time sleep stage classification using single-channel EEG data. The DL model features CNN and transformers. We validated the model using the Sleep-EDF dataset and tested it in a low-power microcontroller. The model achieved performance comparable to the state-of-the-art works. In the future, we plan to develop a fully integrated wireless EEG sensor using the model. The developed device can enable a wide range of sleep research.

REFERENCES

[1] A. K. Patel, V. Reddy, and J. F. Araujo, "Physiology, sleep stages," in *StatPearls [Internet]*. StatPearls Publishing, 2022.

[2] X. Liu, A. G. Richardson, and J. Van der Spiegel, "An energy-efficient compressed sensing-based encryption scheme for wireless neural recording," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 2, pp. 405–414, 2021.

[3] X. Liu and A. G. Richardson, "A system-on-chip for closed-loop optogenetic sleep modulation," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 5678–5681.

[4] X. Liu, B. Subei, M. Zhang, A. G. Richardson, T. H. Lucas, and J. Van der Spiegel, "The pennbmbi: A general purpose wireless brain-machine-brain interface system for unrestrained animals," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 650–653.

[5] B. Kemp, A. Zwinderman, B. Tuk, H. Kamphuisen, and J. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[6] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[7] 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, pp. 1998–2008, 2017.

[8] W. Qu, Z. Wang, H. Hong, Z. Chi, D. D. Feng, R. Grunstein, and C. Gordon, "A residual based attention model for eeg based sleep staging," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 2833–2843, 2020.

[9] N. Goshtasbi, R. Boostani, and S. Sanei, "Sleepfcn: A fully convolutional deep learning framework for sleep stage classification using single-channel electroencephalograms," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2088–2096, 2022.

[10] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[11] D. P. Kingma, "Adam: A method for stochastic optimization/diederik p," *Kingma, Jimmy Ba, URL: https://arxiv. org/abs/1412.6980*.

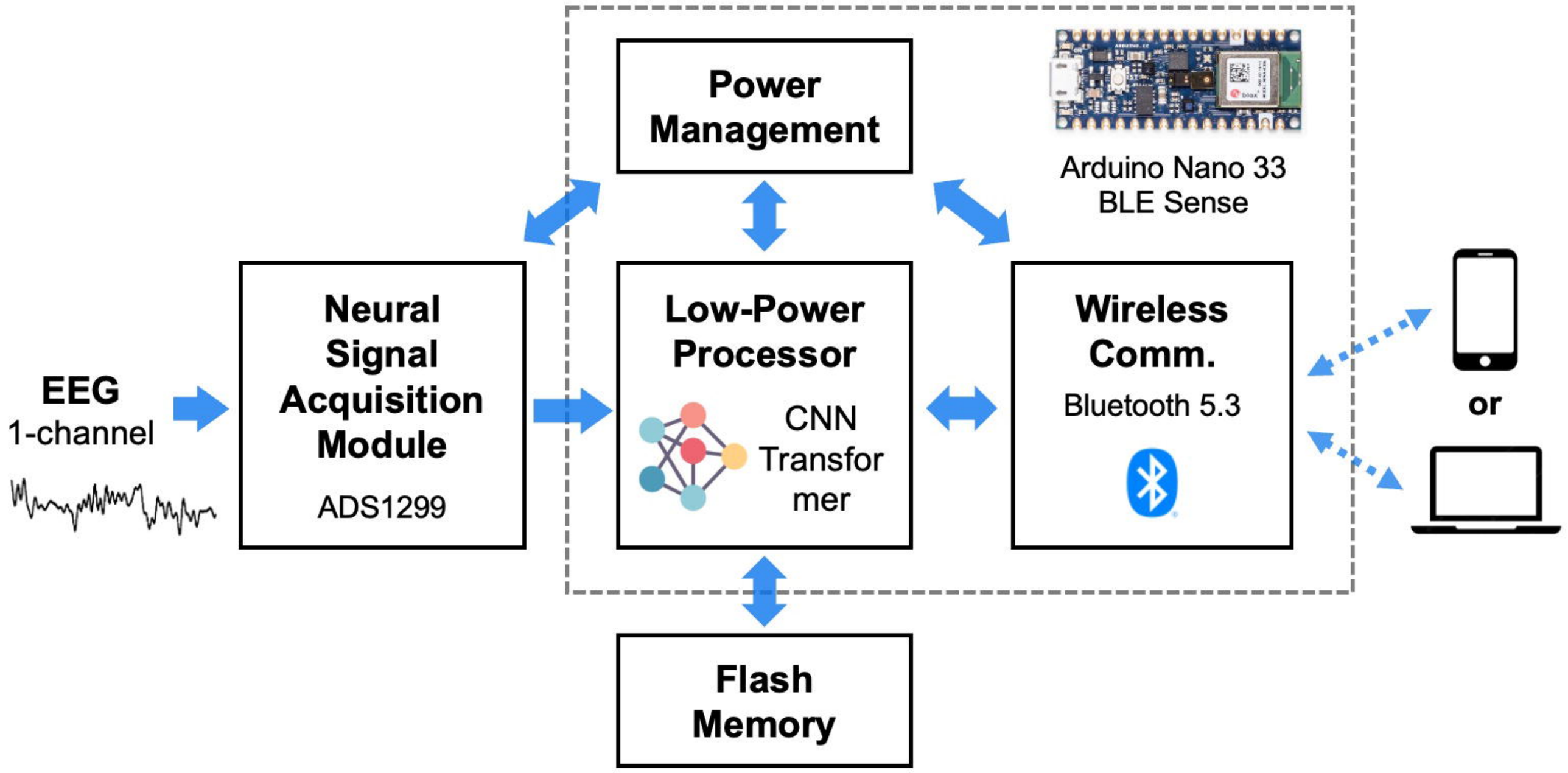
[12] S. Chang, Z. Yang, Y. You, and X. Guo, "Dssnet: A deep sequential sleep network for self-supervised representation learning based on single-channel eeg," *IEEE Signal Processing Letters*, vol. 29, pp. 2143–2147, 2022.

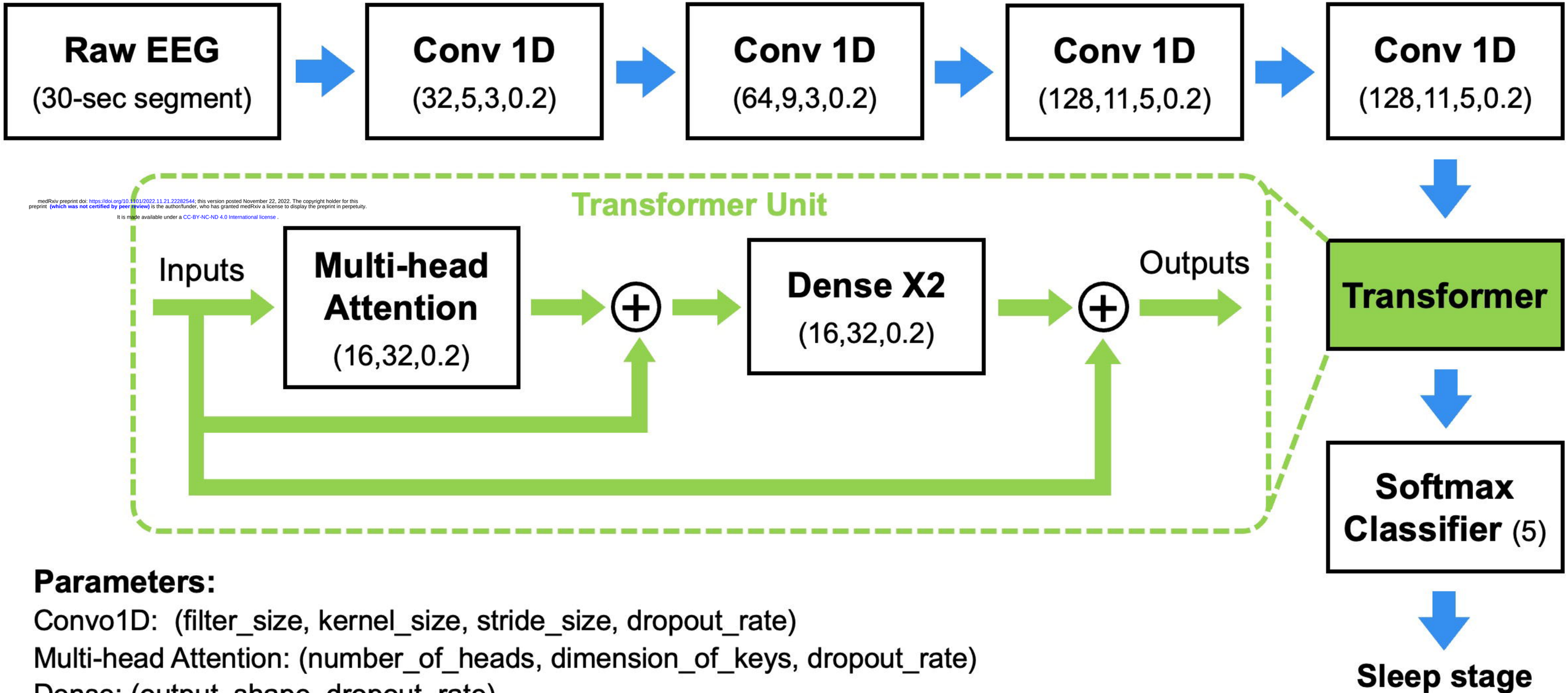
[13] H. Phan, F. Andreotti, N. Cooray, O. Y. Chèn, and M. De Vos, "Dnn filter bank improves 1-max pooling cnn for single-channel eeg automatic sleep stage classification," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 453–456.

[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, p. e0216456, 2019.

[15] C. A. Ellis, R. L. Miller, and V. D. Calhoun, "A novel local explainability approach for spectral insight into raw eeg-based deep learning classifiers," in *2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE)*, 2021, pp. 1–6.

[16] A. Kurniawan, "Iot projects with arduino nano 33 ble sense," *Berkeley: Apress*, vol. 129, 2021.


Power Management
Arduino Nano 33 BLE Sense
EEG
1-channel
Neural Signal Acquisition Module
ADS1299
Low-Power Processor
CNN Transformer
Wireless Comm.
Bluetooth 5.3
or
Flash Memory



Parameters:

Convo1D: (filter_size, kernel_size, stride_size, dropout_rate)

Multi-head Attention: (number_of_heads, dimension_of_keys, dropout_rate)

Dense: (output_shape, dropout_rate)