CNN-GRU for Drowsiness Detection from Electrocardiogram Signal

Setiawan Hendratno[1]*; Nico Surantha[2]

[1-2]BINUS Graduate Program – Master of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
[2]Department of Electrical, Electronics and Communication Engineering, Faculty of Engineering,
Tokyo City University
Tokyo, Japan 158-8557
[1]setiawan.hendratno@binus.ac.id; [2]nico.surantha@binus.ac.id

Received: 19th December 2024/ **Revised:** 25th July 2025/ **Accepted:** 28th July 2025

How to Cite: Hendratno, S., & Surantha, N. (2025). CNN-GRU for drowsiness detection from electrocardiogram signal. *ComTech: Computer, Mathematics and Engineering Applications, 16*(2), 117−125. https://doi.org/10.21512/comtech.v16i2.12755

Abstract **-** Drowsiness is a problem that needs to be addressed to improve road safety. To minimize this safety issue, driving-monitoring systems have been implemented in current car models, and electrocardiography (ECG) is one of the most commonly used driving monitoring techniques. ECG data are modeled using a deep neural network, including a Bidirectional Gated Recurrent Unit (Bi-GRU). However, the accuracy for classifying Wake-Sleep is under 80% and Wake-NREM-REM reaches less than 68%. To address this issue, ECG data from the MESA and SHHS datasets are modeled using a combination of a Convolutional Neural Network (CNN) and a Bi-GRU, referred to as CNN-GRU. This model incorporated Batch Normalization and RMSProp to achieve improved accuracy in classifying drivers' conditions. It operates in two computing sectors: cloud computing (Google Colaboratory, also known as Colab) and edge computing (utilizing an AMD Ryzen 5 4600H processor laptop). Those computing sectors focused on a case where no internet connectivity occurred to process the classification. Those classifications achieved accuracy rates of 82.88% and 81.78% for Wake-Sleep classification in cloud- and edge-computing, respectively. Additionally, it achieved 71.01% (Colab) and 68.85% (edge-computing) accuracy in Wake-NREM-REM classification. This result indicates that CNN-GRU achieved better performance, surpassing the previous Bi-GRU model, which only achieved 80.42% (Colab) and 76.2% (edge-computing) for Wake-Sleep, and 68.85% (Colab) and 66.43% for Wake-NREM-REM.

Keywords: Internet of Things, deep learning, ECG, drowsiness detection, edge computing

I. INTRODUCTION

Traffic accidents are a top priority when implementing road safety measures. Traffic accidents occur due to various factors, one of which is drowsiness, which affects 633 drivers (1.6%) per year in the USA (Kirley et al., 2023). Furthermore, drowsiness contributed to 74.4% of crash incidents reported in Europe, with the Czech Republic accounting for the highest proportion of cases within the continent (84.5%) (van Schagen, 2021). The high number of accidents leads several automobile associations, research institutes, and manufacturers to implement safety systems.

Driver monitoring systems (DMS) represent one of the most prominent areas of current research, offering a practical approach to reducing the incidence of fatal road accidents. These systems primarily utilize facial landmarks and eye aspect ratios (EAR) as the two predominant technologies for assessing driver drowsiness (Florez et al., 2024). Both technologies utilize cameras or image-sensing devices to monitor the condition of potentially drowsy drivers, where the captured images are analyzed using deep learning methodologies, particularly Convolutional Neural Networks (CNN). This results in a classification accuracy of at least 90% (Albasrawi et al., 2022). However, facial landmarks and EAR do not detect drivers under nighttime or low illumination or capture images when the subject is not in a proper position (Shamrat et al., 2022; Guthikonda, 2023). Therefore, image recognition is not viable under unconditional scenarios.

One alternative method for measuring heart rate is electrocardiography (ECG), which demonstrates the

*Corresponding Author

highest accuracy (94%) and precision (97%) (Curtin et al., 2018; Mewada, 2023; Venkatesh et al., 2022). Nowadays, ECG is extracted from polysomnography (PSG), which is used to detect the electrostatic current generated by the pulse of blood vessels and measure peak-to-peak heart rate variability (Dutt et al., 2023; Morokuma et al., 2023). Some researchers recognize this benefit as a method for classifying drowsiness levels by modeling data using a deep learning process.

One of the proposed approaches is SleepECG, which applies a Bi-directional GRU (Bi-GRU) to process inputs from datasets such as the Multi-Ethnic Study of Atherosclerosis (MESA), and Sleep Heart Health Studies (SHHS) to classify human drowsiness (Blaha & DeFilippis, 2021; Chen et al., 2015; Quan et al., 1997; Ullah & Tamanna, 2023). The Bi-GRU processing input also incorporates the RMSProp optimizer to normalize the error and first-order data, thereby avoiding overfitting during processing (Elshamy et al., 2023; Brunner & Hofer, 2023). Incorporating the RMSProp optimizer eliminates the vanishing gradient problem and classifies driver drowsiness levels (Cho et al., 2014; Rama Devi et al., 2024; Zulqarnain et al., 2024).

However, this model does not exceed an accuracy level of 80% for Wake-Sleep and 67% for Wake-NREM-REM. To overcome this issue, a Convolutional Neural Network (CNN) is added to this model. This deep learning combination is suitable for increasing the accuracy of classifying drowsiness (Chollet, 2021). Hence, combining models of CNN and Bi-GRU is chosen as the proposed model.

In this research, a CNN and Bi-GRU are used as a one-packaged model to combine the beneficial advantages of accuracy and handling the vanishing gradient problem. The model runs on supercomputers, which are categorized as cloud-computing systems, using Google Colab. This cloud system enables the utilization of Nvidia A100 GPUs to run complex datasets and modeling (Choquette et al., 2021). After running in a cloud system, the model is also processed on a personal computer (PC) or laptop, which utilizes an AMD Ryzen 5 4600H processor to execute a deep learning model with complex computational tasks (Sulistiyono et al., 2024). This utilization also tests the model's complexities that edge-computing devices might handle. Additionally, this approach also evaluates the model's performance on edge-computing devices and allows comparison of cloud- and edge-computing performance, including a comparison with SleepECG (Brunner & Hofer, 2023).

II. METHODS

In this section, the methods are divided into two platforms: the computing and data processing platform. The computing platform focuses on the tools, platforms, or specifications that the system use. Meanwhile, Data processing emphasizes the flow of data being modeled from the input datasets to be evaluated and run to classify sleep stages.

For the computing platform, this study employs cloud computing through Google Colaboratory with TensorFlow and Keras APIs to develop deep learning programs. Colab also serves as a platform for monitoring CPU and GPU utilization, while the NVIDIA A100 GPU accelerates the modeling of sleep–wake and REM–NREM–wake data. The measurement parameters for both CPU and GPU include power consumption and RAM/ROM utilization. These parameters are measured using the Weights and Biases API, which records both power usage and memory consumption (Tornede et al., 2023).

Another platform used in this study is edge computing. This system employs a laptop, specifically an HP Pavilion Gaming 15 with an AMD Ryzen 5 4600H CPU, which has six cores and 12 threads and provides processing speeds of up to 4 GHz. The Ryzen 5 4600H also integrates a VGA processor with six cores and a frequency of 1500 MHz (Sulistiyono et al., 2024). These specifications are sufficient for processing deep learning models, as similar devices have been used for complex human activity recognition tasks that require large datasets and advanced modeling. In addition, edge computing offers the advantage of operating in locations without internet access (Wang et al., 2025). After defining the computational platform, the next step is to define the data processing workflow. This includes inputting data, implementing algorithms or models, and conducting evaluation after modeling, as illustrated in Figure 1.

For data input, the ECG modeling uses the Multi-Ethnic Study of Atherosclerosis (MESA) dataset (Blaha & DeFilippis, 2021; Chen et al., 2015) for training and the Sleep Heart Health Study (SHHS) database (Quan et al., 1997; Ullah & Tamanna, 2023) to predict the results after being modeled. These two databases must be requested by the National Sleep Research Resource (NSRR) and will be processed for two weeks. Once access is granted, the datasets are downloaded into a Python IDE for model execution. MESA and SHHS retrieval can be achieved using the SleepECG API. Data retrieval from MESA and SHHS is performed using the SleepECG API, where the commands sleepecg.read_mesa() and sleepecg.read_shhs() download and read the full datasets without filtering by file code (Brunner & Hofer, 2023).

The relevant information is extracted from heart rate variability and R-R interval filtering after reading the data, a process referred to as feature extraction. The extracted data from MESA serves as the training dataset, while the SHHS extraction is used for testing and prediction. The training and testing data then enter the staging classification, which is divided into two types: Wake-Sleep and Wake-NREM-REM. The staged training data is used to train the model, which is subsequently tested with the staged testing data to predict driver classification.

After classification into stages is complete, the data are modeled using a CNN-GRU architecture adapted from the Bi-GRU model of SleepECG (Brunner & Hofer, 2023). The proposed CNN-GRU

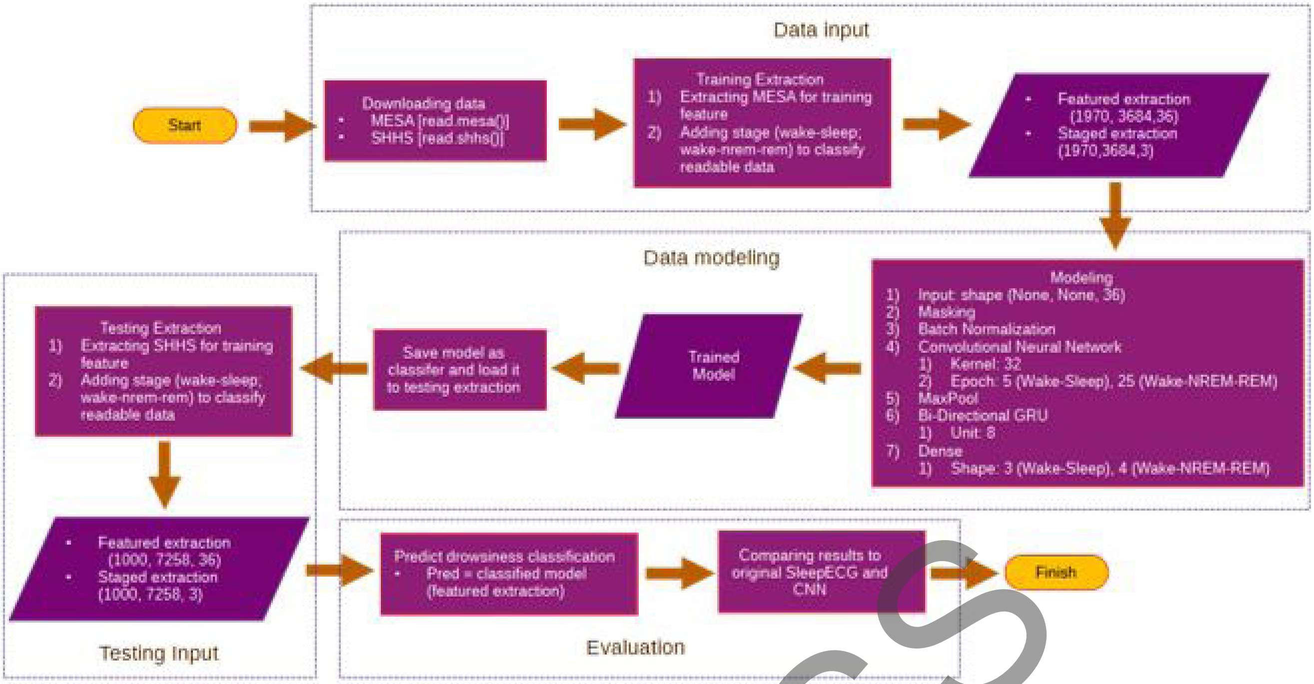

Figure 1 Research Workflow

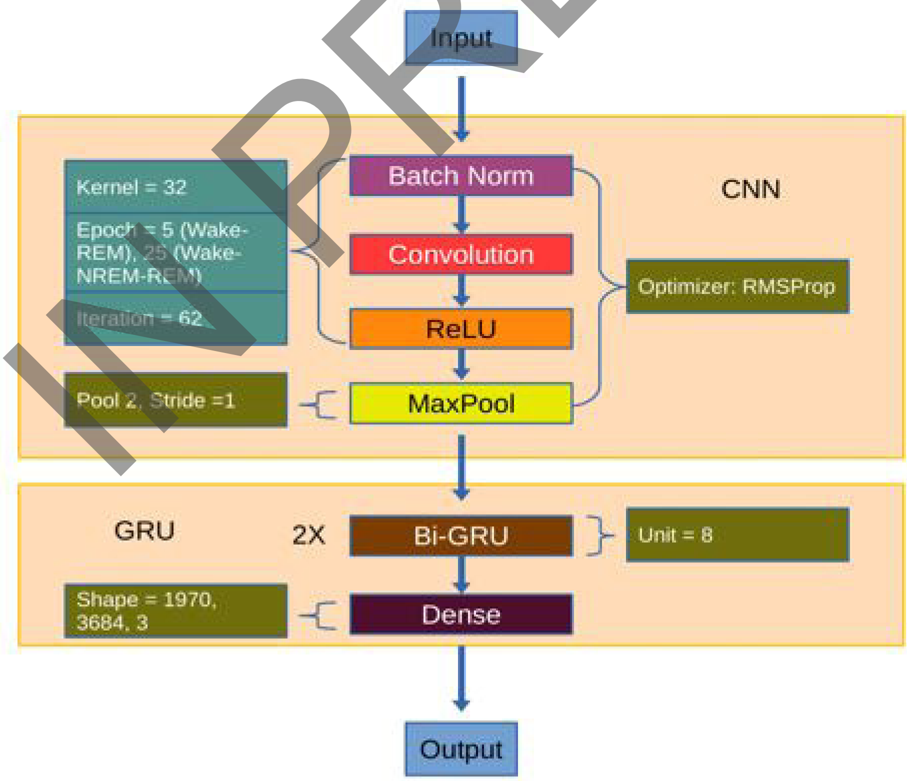

Figure 2 CNN-GRU Model

runs with five epochs for Wake-REM and 25 epochs for Wake-NREM-REM, using 62 iterations to process a total of 7,257,480 data cells (1970 × 3684) as training input. The architecture consists of convolution, max-pooling, batch normalization, and bidirectional GRU layers. The convolution layer applies 16 filters with a kernel size of two units and uses padding to maintain input dimensions, while the max-pooling layer employs a pool size of two units and a stride of one unit with padding so that pooling does not reduce the data dimensions.

The data are normalized and densely connected after pooling, after which the process continues with the GRU. The GRU is designed with two bidirectional layers consisting of 16 units, and its output is passed to a dense layer that is fully connected to the softmax function. To optimize the CNN-GRU, the RMSProp optimizer is applied together with categorical cross-entropy to evaluate and minimize loss. This optimization process limits weight updates, calculates momentum, and adjusts bias to improve model stability (Elshamy et al., 2023).

To summarize the model, Figure 2 presents its structure, including the optimizer. The SHHS data are also incorporated into the CNN-GRU model, with the same dimensions as the MESA dataset (1970 × 3684), and are used for testing the proposed model. The processed data generate a confusion matrix that reports accuracy, Cohen's Kappa, F1-score, and precision.

To evaluate the model, the data processed by the CNN–GRU algorithm are compared with SleepECG and other studies using accuracy results obtained from the Keras metrics. The accuracy is calculated using the following formula in Equation (1), where *TP* represents the confusion matrix value at the top left, *TN* the value at the bottom right, *FP* the value at the bottom left, and *FN* the value at the top right. In addition to accuracy, memory usage is also assessed by comparing the GRU model of Brunner and Hofer with the proposed CNN–GRU model. This comparison determines the RAM capacity required for processing each model after extracting data from the MESA dataset.

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN} \tag{1}$$

Following the assessment of data accuracy and precision, a comparative analysis is conducted between the CNN-GRU-processed data and the SleepECG model. This evaluation encompasses three primary aspects: the accuracy metrics of each implemented algorithm, kappa values (Warren, 2015), and the computational demands of data processing. Accuracy metrics are derived from the findings reported by Brunner and Hofer, utilizing datasets from MESA and SHHS. The assessment of computational resource consumption focuses on RAM utilization during data processing with the three algorithms.

In addition to comparing Brunner and Hofer's GRU model, the CNN-GRU model is also evaluated against 1-dimensional CNN models inspired by Ellis et al.'s work. The original Ellis model consists of an 8-layer 1-dimensional convolutional network, with six layers forming the initial structure of the deep learning network (phase i) and two additional layers positioned in the middle (phase ii) before the training and testing data are processed in the fully connected layer (Ellis et al., 2021). In this research, however, only phase (ii) of the CNN model is used, with a simplified design that includes a single convolutional layer, no dropout, and one fully connected dense layer. This configuration applies 32 convolutional kernels along with a pooling size of two and a stride of one.

III. RESULTS AND DISCUSSIONS

The experimental results are presented in two parts. The first part explains the accuracy and loss of data processing. The second part focuses on the results of model accuracy and Cohen's kappa coefficient (Warrens, 2015; Zhang et al., 2021) for the previous and proposed models.

The first part of the experimental results explains the accuracy and loss of data processing by comparing CNN-GRU, Bi-GRU, and CNN. For training, the MESA dataset is loaded and modeled using a CNN Bi-GRU. RMSProp and categorical cross-entropy are applied to correct and minimize data errors. The model runs on both Google Colaboratory and edge-computing systems. The data are then processed with CNN and Bi-GRU deep learning models, and the outputs of these processes are summarized in Table 1.

The CNN-GRU model demonstrates better processing time than the Bi-GRU model, requiring 1 hour 40 seconds (3,640 seconds) for Wake-REM and 5 hours 42 minutes 39 seconds (20,559 seconds) for Wake-NREM-REM when using a CPU. The processing time improves further with the NVIDIA A100, decreasing to 27 minutes 9 seconds (1,629 seconds) for Wake-REM and 4 hours 59 minutes 48 seconds (17,988 seconds) for Wake-NREM-REM. Additionally, the model achieves its highest accuracy of 93.46% when trained with the NVIDIA A100. However, RAM consumption remains relatively high, reaching 8.6 GB with the A100 for Wake-REM, compared to only 5.69 GB when using the CPU.

However, when trained with the Wake-NREM-REM stage, the RAM capacity used is 9.11 GB, a 0.39 GB difference from the 8.72 GB used in the CPU. These results also happened to the edge computing model, which used a capacity of 2.9 GB for classifying Wake-Sleep and 4.05 GB for Wake-NREM-REM. These results indicate that CNN-GRU has a higher accuracy value and lower loss compared to CNN and Bi-GRU separately. However, the use of higher RAM with an Nvidia A100 GPU in the CNN-GRU model, particularly in the wake-REM stage, is a special concern.

After training the model with MESA, it is tested using the SHHS dataset. The test measures model accuracy and the kappa coefficient for classifying sleepiness levels. In addition, precision, F1 score,

Table 1 Comparison After Training Classification of Deep Learning

CLASSIFIER	MODEL	SYSTEM	ACC (%)	PROC TIME (S)	RAM (GB)
Wake-REM	CNN	CPU (Xeon®)	91.8	95	5.47
	Bi-GRU		79.7	4522	6.26
	CNN-GRU		91.1	3640	5.69
	CNN	GPU (A100)	91.85	21	6.82
	Bi-GRU		80.18	3.303	7.56
	CNN-GRU		93.46	1.629	8.6
	CNN	Edge Comp	91.8	78	1.98
	Bi-GRU		76.3	3	2.57
	CNN-GRU		91.99	2	2.9
Wake-NREM-REM	CNN	CPU (Xeon®)	88.98	542	7.69
	Bi-GRU		74.3	27.425	9.75
	CNN-GRU		89.96	20.559	8.72
	CNN	GPU (A100)	89.2	367	8.34
	Bi-GRU		74.98	24.156	10.2
	CNN-GRU		89.98	17.988	9.11
	CNN	Edge Comp	89.03	472	2.81
	Bi-GRU		75.6	23,927	4.75
	CNN-GRU		89.74	20.173	4.05

Table 2 Comparison After Testing Deep Learning Model for Classifications (CPU vs A100 vs Edge) for Accuracy, Cohen's Kappa, Precision, and F1 Score

CLASSIFIER	MODEL	PROCESS TYPE	ACC (%)	*COHEN'S KAPPA*	PRECISION (%)			*F1 SCORE* (%)		
					REM	NREM	Wake	REM	NREM	Wake
Wake-REM	Bi-GRU	CPU	79.84	0.5459	88		65	85		77
	CNN-GRU		83.06	0.6013	88		73	88		72
	CNN		73.34	0.4262	86		55	84		70
	Bi-GRU	A100	80.42	0.5416	88		65	87		72
	CNN-GRU		82.88	0.6014	88		72	88		73
	CNN		80.64	0.5059	82		77	87		63
	Bi-GRU	Edge computing	76.20	0.4749	87		59	82		75
	CNN-GRU		81.78	0.594	87		74	88		71
	CNN		72.21	0.4187	86		50	83		71
Wake-NREM-REM	Bi-GRU	CPU	67.36	0.4973	31	83	67	43	63	69
	CNN-GRU		69.35	0.5189	39	84	79	52	74	74
	CNN		65.51	0.3743	46	71	58	9	75	63
	Bi-GRU	A100	68.84	0.5032	37	83	64	38	65	71
	CNN-GRU		71.01	0.5214	39	85	78	52	74	75
	CNN		68.34	0.4437	47	70	63	11	72	59
	Bi-GRU	Edge computing	66.43	0.4972	37	83	64	38	65	71
	CNN-GRU		68.85	0.5112	38	85	78	51	74	72
	CNN		61.93	0.3378	46	72	55	7	73	67

and testing time are evaluated to assess the model's classification performance. The results of these measurements for each model are presented in Table 2.

As shown in Table 2, the CNN-GRU model achieves higher accuracy and Kappa values than the CNN and Bi-GRU models when evaluated separately. Specifically, it reaches 83.06% accuracy with a Kappa of 0.6013 on the Intel Xeon CPU, and 82.88% accuracy with a Kappa of 0.6014 on the A100 GPU for Wake-REM classification. For Wake-NREM-REM classification, the model obtains 69.35% accuracy with a Kappa of 0.5189 on the CPU, and 71.01% accuracy with a Kappa of 0.5214 on the A100 GPU.

Meanwhile, the accuracy and Kappa value of CNN-GRU in edge computing reached 81.78% and 0.594, compared to the accuracy and Cohen's kappa values of the GRU, which reached 76.20% and 0.4749, respectively. For Wake-NREM-REM, the accuracy and Cohen's Kappa values of the CNN-GRU are 68.85% and 0.5112, respectively. These values are higher than those of Brunner and Hofer's GRU and Ellis et al. 's CNN (Ellis et al., 2021). Thus, the CNN-GRU had a higher value than the two previously designed models.

After evaluating the model for sleep stage classification, the deep learning model's processing is analyzed in terms of device utilization, including RAM, GPU, power, and CPU usage. The device is measured using the Weight and Biases API to capture metrics such as RAM capacity, ROM, GPU power, processor capability, and GPU memory. These measurements are then summarized and presented in Table 3.

The RAM and CPU usage of the Bi-GRU, CNN, and CNN-GRU models increase significantly when the NVIDIA A100 GPU is activated, as shown in Table 3. In the CNN model, RAM usage rises from 5.47 GB to 6.82 GB. Similarly, the GRU and CNN-GRU models require 7.56 GB and 8.12 GB of RAM, respectively, when running with the A100. By contrast, when the GPU is not activated, RAM consumption is reduced to 6.26 GB for GRU and 5.69 GB for CNN-GRU in processing the driver/patient drowsiness classification task.

CNN-GRU RAM usage can be considered the largest in this study. This extensive RAM usage is due to the GPU's processing only utilizing 0.44 GB of Video RAM (V-RAM) when it is activated. Consequently, the Intel Xeon CPU must process the CNN-GRU model using 108 system/processor threads. This enables the CNN-GRU to be the largest RAM and CPU user when the NVIDIA A100 GPU is enabled.

Meanwhile, when the model is only processed through a CPU with a 16 GB RAM capacity, the CNN-GRU model requires only 5.69 GB of RAM. The GRU model required the most RAM, with a usage of 6.26 GB. The use of ROM and CPU threads requires only 35 systems/processor threads and 38.08 GB. The CNN model utilizes the least RAM, ROM, and CPU threads, with figures of 5.47 GB, 37, and 30 system/processor threads, respectively. Instead, while the model is processed in edge computing, the three models are assisted by the internal GPU of the CPU, which has a video RAM capacity of 0.155 GB. The CNN-GRU model processing obtains the lowest ROM usage, at 1.25 GB.

Meanwhile, the two models require different amounts of RAM, specifically 30.28 GB for GRU and 3.01 GB for CNN. Meanwhile, the CNN model utilizes the fewest CPU threads, specifically 71 system/processor threads. The Bi-GRU model had the highest thread usage, with a value of 75 systems per thread. This is undoubtedly one of the factors contributing to accuracy and the highest Kappa value compared to the two separate models.

Table 3 Comparison After Testing Deep Learning Model for Classifications (CPU vs A100 vs Edge)

UTILIZATION	SYSTEM	WAKE-REM			WAKE-NREM-REM		
		GRU	CNN-GRU	CNN	GRU	CNN-GRU	CNN
RAM (GB)	*CPU Only (Colab)*	6.26	5.69	5.47	8.73	6.93	7.32
	CPU+With A100	7558	8118	6.82	9.07	7583	7632
	Edge computing	2.57	2.40	1.98	3.07	2.87	1.98
ROM(GB)	*CPU Only (Colab)*	38.08	38.07	37	40.02	40	38.1
	CPU+With A100	38.17	39.85	39.75	40.11	40.53	40.23
	Edge computing	30.28	1.25	3.01	32.14	2.35	3.01
CPU Thread (System/ Processor)	*CPU Only (Colab)*	35	36	30	41	37	37
	CPU+With A100	62.21	45.41	53.2	67.47	47.81	55.1
	Edge computing	75	72	71	78	75	73
VRAM (GB)	*CPU+With A100*	2874	0.441	0.85	2961	0.892	0.88
	Edge computing	0.155	0.155	0.155	0.155	0.155	0.155

The second part of the experimental results involves comparing CNN-GRU and other deep learning models. CNN-GRU outperforms the previous Bi-GRU by achieving 83.06% accuracy and 0.6014 in cloud computing, as well as 81.78% accuracy and 0.594 in edge computing. Nevertheless, this model can be improved by comparing it with other models that combine multiple deep learning techniques into a more comprehensive, single model. Some models use a combination of feed-forward and back-propagation methods, while others employ a simple convolutional layer. The models mentioned are listed in Table 4.

In Table 4, the Bi-GRU and CNN-DA models for wake-sleep classification show varying accuracy and Cohen's Kappa values compared to the CNN-GRU model. The CNN model studied by Tang et al. (2022) achieves only 65.1% accuracy with the SHHS dataset, which is lower than the accuracy of the CNN-GRU model for Wake-Sleep classification. However, Tang's model classifies the Wake-N1-N2-Sleep category with slightly lower accuracy than the Wake-NREM-REM CNN-GRU, which achieves the highest score of 71.01%. Meanwhile, the CNN-DA model exceeds 83% accuracy and achieves a Cohen's Kappa coefficient of 0.749, which is the highest value reported so far. This improvement is possible because the CNN-DA model combines data points from both training and test datasets before encoding, and also employs Leaky ReLU to maintain the gradient of negative inputs, thereby preventing severe over- or under-correction (Maniatopoulos & Mitianoudis, 2021).

In comparison, the Bi-GRU model developed by Brunner and Hofer (2023) achieved a Kappa value of 0.5459. This performance is attributed to the GRU's capability to ignore unreadable or corrupted values and update them with newly processed outputs. As a result, the model yielded relatively high precision and F1 scores, with 88% and 85% for sleep modes, and 65% and 77% for wake modes, respectively. Nevertheless, its Cohen's Kappa remained lower than that of the CNN-GRU model byTang et al. (2022).

Apart from these two models, CNN-GRU is also compared with CoSleepNet, which demonstrates superior performance. Although CNN-GRU achieves an accuracy of 83.06%, which is close to CNN-LSTM's 83.55%, its Kappa value (0.6013) is significantly lower than CoSleepNet's 0.7693. This gap results from CoSleepNet's hybrid architecture, which integrates focal loss to handle class imbalance and employs a discrete cosine transform to correct irregular data features (Efe & Ozsen, 2023). Incorporating focal loss or similar mechanisms may enhance CNN-GRU's ability to manage imbalance prior to normalization and optimization.

IV. CONCLUSIONS

CNN-GRU is developed from SleepECG's Bi-GRU to provide higher accuracy and a higher Kappa coefficient. The model achieves an accuracy of 83.06% and a Kappa value of 0.6013 for Wake-Sleep classification, and 71.01% with a Kappa value of 0.5214 for Wake-NREM-REM classification. In addition, the model is designed to optimize memory utilization, reducing RAM usage from 8.1 GB to 7.4 GB. These results indicate that CNN-GRU surpasses SleepECG's Bi-GRU in terms of both performance and efficiency.

Despite these improvements, CNN-GRU still requires further development to achieve higher precision and F1 Score values. Compared to advanced models such as CoSleepNet, which can classify five levels of sleepiness, CNN-GRU remains limited in its classification capability. To address this limitation, performance can be enhanced by training and testing on larger datasets, modifying stride and padding structures in pooling layers, and using padding values beyond zero. Furthermore, employing alternative optimizers such as Adam in comparison with RMSProp and exploring multi-dimensional features may help improve both accuracy and the Cohen's Kappa coefficient.

Table 4 Comparison of the Performance of Other Previous Deep Learning Models with the Proposed.

CLASSIFIER	MODEL	ACC (%)	COHEN'S KAPPA	ARTICLE
ake-N1-N2-Sleep	CNN-DA	65.1	0.749	(Tang et al., 2022)
Wake-N1-N2-N3-Sleep	CNN-GRU	83.15	0.76	(Pei et al., 2022)
Wake - Sleep	GRU	79.84	0.5459	(Brunner & Hofer, 2023)
Wake-NREM- Sleep	UTSN	89.5	-	(Tezuka et al., 2021)
Wake-N1-N2-N3-Sleep	CoSleepNet (CNN-LSTM)	83.55	0.7693	(Efe & Ozsen, 2023)
Wake - Sleep	CNN-GRU	83.06	0.6013	Proposed research
		81.78	0.594	
Wake-NREM-REM	CNN-GRU	71.01	0.5214	Proposed research
		68.85	0.5112	

AUTHOR CONTRIBUTIONS

Conceived and designed the analysis (Modified recurrent neural network (Bi-GRU) to be CNN-layered Bi-GRU); Collected the data (Gathered Data from MESA and SHHS to be featured); Performed the analysis (Performed the computer simulation); Wrote the paper (Wrote the methodology and results on paper ComTech), S. P. H.; Conceived and designed the analysis (Supported the idea of modifying Bi-GRU with convolutional layered neural network); Wrote the paper (Corrected the flow and structure of the paper); Other contribution (Guided the direction of the research and making of the paper), N. S.

DATA AVAILABILITY

The data that support the findings of this study are available from National Sleep Research Resource. Restrictions apply to the availability of these data, which were used under license for this study. Data are available at https://sleepdata.org/datasets/ with the permission of National Sleep Research Resource.

REFERENCES

Albasrawi, R., Fadhil, F. F., & Ghazal, M. T. (2022). Driver drowsiness monitoring system based on facial Landmark detection with convolutional neural network for prediction. *Bulletin of Electrical Engineering and Informatics*, *11*(5), 2637–2644. https://doi.org/10.11591/eei.v11i5.3966

Blaha, M. J., & DeFilippis, A. P. (2021). Multi-Ethnic Study of Atherosclerosis (MESA): JACC focus seminar 5/8. In *Journal of the American College of Cardiology* (Vol. 77, Issue 25, pp. 3195–3216). Elsevier Inc. https://doi.org/10.1016/j.jacc.2021.05.006

Brunner, C., & Hofer, F. (2023). SleepECG: A Python package for sleep staging based on heart rate. *Journal of Open Source Software*, *8*(86), 5411. https://doi.org/10.21105/joss.05411

Carneiro, T., Da Nobrega, R. V. M., Nepomuceno, T., Bian, G. Bin, De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, *6*, 61677–61685. https://doi.org/10.1109/ACCESS.2018.2874767

Chen, X., Wang, R., Zee, P., Lutsey, P. L., Javaheri, S., Alcántara, C., Jackson, C. L., Williams, M. A., & Redline, S. (2015). Racial/ethnic differences in sleep disturbances: The Multi-Ethnic Study of Atherosclerosis (MESA). *Sleep*, *38*(6), 877–888. https://doi.org/10.5665/sleep.4732

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN Encoder-Decoder for statistical machine translation*. http://arxiv.org/abs/1406.1078

Chollet, F. (2021). *Deep learning with Python, Second Edition* (2nd ed.). Manning. http://ieeexplore.ieee.org/document/10280481

Choquette, J., Gandhi, W., Giroux, O., Stam, N., & Krashinsky, R. (2021). NVIDIA A100 tensor core GPU: Performance and innovation. *IEEE Micro*, *41*(2), 29–35. https://doi.org/10.1109/MM.2021.3061394

Curtin, A. E., Burns, K. V., Bank, A. J., & Netoff, T. I. (2018). QRS complex detection and measurement algorithms for multichannel ECGs in cardiac resynchronization therapy patients. *IEEE Journal of Translational Engineering in Health and Medicine*, *6*. https://doi.org/10.1109/JTEHM.2018.2844195

Devi, G. R., Al-Tmimi, H. M., Ghadir, G. K., Sharma, S., Patnala, E., Bala, B. K., & Baker El-Ebiary, Y. A. (2024). COOT-Optimized real-time drowsiness detection using GRU and enhanced deep belief networks for advanced driver safety. *International Journal of Advanced Computer Science & Applications*, *15*(4).

Dhillon, A., & Verma, G. K. (2020). Convolutional Neural Network: A review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, *9*(2), 85–112. https://doi.org/10.1007/S13748-019-00203-0/METRICS

Dutt, M., Redhu, S., Goodwin, M., & Omlin, C. W. (2023). SleepXAI: An explainable deep learning approach for multi-class sleep stage identification. *Applied Intelligence*, *53*(13), 16830–16843. https://doi.org/10.1007/s10489-022-04357-8

Efe, E., & Ozsen, S. (2023). CoSleepNet: Automated sleep staging using a hybrid CNN-LSTM network on imbalanced EEG-EOG datasets. *Biomedical Signal Processing and Control*, *80*, 104299. https://doi.org/https://doi.org/10.1016/j.bspc.2022.104299

Ellis, C. A., Zhang, R., Carbajal, D. A., Miller, R. L., Calhoun, V. D., & Wang, M. D. (2021). *Explainable sleep stage classification with multimodal electrophysiology time-series*. https://doi.org/10.1101/2021.05.04.442658

Elshamy, R., Abu-Elnasr, O., Elhoseny, M., & Elmougy, S. (2023). Improving the efficiency of RMSProp optimizer by utilizing Nestrove in deep learning. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-35663-x

Florez, R., Palomino-Quispe, F., Alvarez, A. B., Coaquira-Castillo, R. J., & Herrera-Levano, J. C. (2024). A real-time embedded system for driver drowsiness detection based on visual analysis of the eyes and mouth using Convolutional Neural Network and mouth aspect ratio. *Sensors*, *24*(19). https://doi.org/10.3390/s24196261

Guthikonda, P. S. R. (2023). IoT based driver drowsiness detection and alerting system using Haar Cascade and eye aspect ratio algorithms. In *Integration of AI-Based Manufacturing and Industrial Engineering Systems with the Internet of Things* (1st ed.). CRC Press.

Kirley, B., Robison, K., Goodwin, A., Harmon, K., O'Brien, N., West, A., Harrell, S., Thomas, L., & Brookshire, K. (2023). *Countermeasures that work: A highway safety countermeasure guide for state highway safety offices 11th EDITION 2 0 2 3*. https://rosap.ntl.bts.gov.

Korompili, G., Amfilochiou, A., Kokkalas, L., Mitilineos, S. A., Tatlas, N. A., Kouvaras, M., Kastanakis, E., Maniou, C., & Potirakis, S. M. (2021). PSG-Audio, a scored polysomnography dataset with simultaneous audio recordings for sleep apnea studies. *Scientific Data*, *8*(1). https://doi.org/10.1038/s41597-021-00977-w

Maniatopoulos, A., & Mitianoudis, N. (2021). Learnable Leaky ReLU (LeLeLU): An alternative accuracy-optimized activation function. *Information (Switzerland)*, *12*(12). https://doi.org/10.3390/info12120513

Mewada, H. (2023). 2D-wavelet encoded deep CNN for image-based ECG classification. *Multimedia Tools and Applications*, *82*(13), 20553–20569. https://doi.org/10.1007/s11042-022-14302-z

Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Convolutional Neural Networks. In *Multivariate statistical machine learning methods for genomic prediction* (pp. 533–577). Springer International Publishing. https://doi.org/10.1007/978-3-030-89010-0_13

Morokuma, S., Hayashi, T., Kanegae, M., Mizukami, Y., Asano, S., Kimura, I., Tateizumi, Y., Ueno, H., Ikeda, S., & Niizeki, K. (2023). Deep learning-based sleep stage classification with cardiorespiratory and body movement activities in individuals with suspected sleep disorders. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-45020-7

Pei, W., Li, Y., Siuly, S., & Wen, P. (2022). A hybrid deep learning scheme for multi-channel sleep stage classification. *Computers, Materials and Continua*, *71*(1), 889–905. https://doi.org/10.32604/cmc.2022.021830

Quan, S. F., Howard, B. V., Iber, C., Kiley, J. P., Nieto, F. J., O'Connor, G. T., Rapoport, D. M., Redline, S., Robbins, J., Samet, J. M., & Wahl, P. W. (1997). The sleep heart health study: Design, rationale, and methods. *Sleep*, *20*(12), 1077–1085. https://doi.org/10.1093/sleep/20.12.1077

Shamrat, F. M. J. M., Majumder, A., Antu, P. R., Barmon, S. K., Nowrin, I., & Ranjan, R. (2022). Human face recognition applying Haar Cascade classifier. *Lecture Notes in Networks and Systems*, *317*, 143–157. https://doi.org/10.1007/978-981-16-5640-8_12

Sulistiyono, M., Ariadi, M. V., Kharisma, R. S., & Saputro, U. A. (2024). Performance analysis of AMD Ryzen 5 4600 H mobile processor undervolting using AMD APU tuning utility on Cinebench R23. *Journal of Soft Computing Exploration*, *5*(2), 133–144. https://doi.org/10.52465/joscex.v5i2.369

Sun, H., Ganglberger, W., Panneerselvam, E., Leone, M. J., Quadri, S. A., Goparaju, B., Tesh, R. A., Akeju, O., Thomas, R. J., & Westover, M. B. (2020). Sleep staging from electrocardiography and respiration with deep learning. *Sleep*, *43*(7). https://doi.org/10.1093/SLEEP/ZSZ306

Tang, M., Zhang, Z., He, Z., Li, W., Mou, X., Du, L., Wang, P., Zhao, Z., Chen, X., Li, X., Chang, H., & Fang, Z. (2022). Deep adaptation network for subject-specific sleep stage classification based on a single-lead ECG. *Biomedical Signal Processing and Control*, *75*, 103548. https://doi.org/https://doi.org/10.1016/j.bspc.2022.103548

Tezuka, T., Kumar, D., Singh, S., Koyanagi, I., Naoi, T., & Sakaguchi, M. (2021). Real-time, automatic, open-source sleep stage classification system using single EEG for mice. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-90332-1

Tornede, T., Tornede, A., Fehring, L., Gehring, L., Graf, H., Hanselle, J., Mohr, F., & Wever, M. (2023). PyExperimenter: Easily distribute experiments and track results. *Journal of Open Source Software*, *8*(84), 5149. https://doi.org/10.21105/joss.05149

Ullah, M. I., & Tamanna, S. (2023). Racial disparity in cardiovascular morbidity and mortality associated with obstructive sleep apnea: The sleep heart health study. *Sleep Medicine*, *101*, 528–534. https://doi.org/https://doi.org/10.1016/j.sleep.2022.12.007

van Schagen, I. (2021). *European Road Safety Observatory Road Safety Thematic Report-Fatigue*. https://www.swov.nl/en/facts-figures/factsheet/fatigue0

Venkatesh, N. P., Arya, B., Dhananjay, B., & Sivaraman, J. (2022). The role of optimal and modified lead systems in electrocardiogram. In *Advanced methods in biomedical signal processing and analysis* (pp. 311–340). Elsevier. https://doi.org/10.1016/B978-0-323-85955-4.00014-4

Wang, C., Chen, L., Zhou, B., Xian, Y., Zhao, Y., & Huan, Z. (2025). CILOSR: A unified framework for enhanced class incremental learning based open-set human activity recognition using wearable sensors. *Expert Systems with Applications*, *273*, 126893. https://doi.org/https://doi.org/10.1016/j.eswa.2025.126893

Warrens, M. J. (2015). Five ways to look at Cohen's Kappa. *Journal of Psychology & Psychotherapy*, *05*(04). https://doi.org/10.4172/2161-0487.1000197

Zhang, J., Tang, Z., Gao, J., Lin, L., Liu, Z., Wu, H., Liu, F., & Yao, R. (2021). Automatic detection of obstructive sleep apnea events using a deep CNN-LSTM model. *Computational Intelligence and Neuroscience*, *2021*. https://doi.org/10.1155/2021/5594733

Zulqarnain, M., Alsaedi, A. K. Z., Sheikh, R., Javid, I., Ahmad, M., & Ullah, U. (2024). An improved gated recurrent unit based on auto encoder for sentiment analysis. *International Journal of Information Technology*, *16*(1), 587–599. https://doi.org/10.1007/s41870-023-01600-4