CRT: a Convolutional Recurrent Transformer for automatic sleep state detection

CRT: A Convolutional Recurrent Transformer for Automatic Sleep State Detection

SM Nuruzzaman Nobel, S M Masfequier Rahman Swapno, Md Mohsin Kabir, M. F. Mridha, *Senior Member, IEEE*, Nilanjan Dey, *Senior Member, IEEE*, R. Simon Sherratt, *Fellow, IEEE*

***Abstract*— Sleep is a crucial period of rest necessary for optimal cognitive function, psychological well-being, and execution of everyday tasks. In the field of sleep healthcare, the primary objective is to identify and classify the various sleep states. Implementing sleep state detection in a system is problematic and essential for accurate diagnosis. Our study used an integrated framework to recognize sleep states. The dataset contained approximately eight lakh data points sorted into two groups: onset and wake-up. We successfully deployed a cutting-edge Convolutional Recurrent Transformer (CRT) model for sleep state detection. The training accuracy of our detection model was measured at 97.83%, a constant validation accuracy of 97.07%, and a testing measurement accuracy of 97.23%, were maintained. These scores indicate the model's proficiency in precisely recognizing the sleep states. Our system's detection capabilities demonstrate the ability to identify different sleep states, enhance the accuracy of diagnoses and increase healthcare outcomes in this specialized field.**

***Index Terms*— Machine Learning, Deep Learning, Sleep State, CNN, RNN, Healthcare, CRT, Transformer.**

I. Introduction

SLEEP [1] is a fundamental physiological process crucial for cognitive function, mental well-being, and daily functioning. Despite its importance, the prevalence of sleep-related disorders such as insomnia, sleep apnea, and restless legs is increasing. Optimal sleep [2] encompasses various factors, including duration, quality, scheduling, and consistency [3]. Lack of sufficient sleep affects cognitive abilities, hunger regulation, insulin response, immune function, emotional state, response time, focus, and memory. Moreover, many of the population suffer from sleep issues, and many cases remain undiagnosed [4]. Specifically, light and deep sleep phase [5], [6] are crucial for supporting various biological processes, including immune function and memory.

Sleep is essential [7], [8] for the molecular maintenance and regulation of many biological processes, which aids in the restoration of human physical and mental health as well as appropriate brain function during the day [9]. Unfortunately, sleep disorders are frequently disregarded, resulting in poor sleep quality [10]. According to Stranges et al. [10], sleep-related issues are a serious global health concern. They examined the prevalence of sleep issues in low-income nations using data from the International Network for the Demographic Evaluation of Populations and Their Health (INDEPTH) and the World Health Organization (WHO). According to reports, 150 million adults, or 16.6% of the total population, suffer from sleep issues, and current trends suggest that this number will rise to 260 million by 2030.

Recently, sleep state classification has become increasingly essential for the effective care and management of individuals with sleep disorders. Ambulatory sleep detection and bedtime monitoring are common strategies for detecting and treating sleep-related issues [11]. However, traditional methods such as polysomnography (PSG) are impractical for continuous monitoring, owing to their intrusive nature [12]. Human specialists' current manual classification of sleep stages suffers from variability and cost constraints [13]. Portable monitoring devices offer promise for improving accessibility and reducing costs; however, reliability concerns persist [14]. To address these challenges, reliable and efficient automatic sleep state classification has become crucial.

Researchers specialize in building automatic and intelligent systems and employ various methods, such as machine learning [15]–[17], and deep learning [18]. Furthermore, numerous experts have focused on sleep state classification and worked on two sleep stages: onset and wakeup. Ram et al. [19] introduced a machine-learning technique to identify two sleep states: onset and waking phases. They focused on identifying children's sleep states. An RCNN model was utilized to detect sleep states. They identified the precision and IOU matrix and attained a model accuracy of 70.05%. Bjorvatn and Pallesen et al. [20] discovered circadian rhythm sleep disorders. They evaluated two data points for implementing their system: onset and wakeup. In addition, they identified sleep phase abnormalities. They focused on determining the regular phase by analyzing a thorough patient history. Fraiwan and Lweesy et al. [21] introduced a method for identifying the sleep states of newborns using deep learning. They are

SM Nuruzzaman Nobel is with the Department of CSE, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh (e-mail: smnuruzzaman712@gmail.com)

S M Masfequier Rahman Swapno is with the Department of CSE, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh (e-mail: masfequier.cse.bubt@gmail.com).

Md Mohsin Kabir is with Superior Polytechnic School, University of Girona, 17071 Girona, Spain (e-mail: mdmkabi@gmail.com).

M. F. Mridha is with the Department of Computer Science, American International University-Bangladesh, Dhaka 1229, Bangladesh (e-mail: firoz.mridha@aiub.edu).

Nilanjan Dey is with Techno International New Town, 700156, Kolkata, India. (e-mail: nilanjan.dey@tint.edu.in).

R. Simon Sherratt is with the Department of Biomedical Engineering, University of Reading, RG5 4XW UK. (e-mail: r.s.sherratt@reading.ac.uk).

engaged in deep learning, feature extraction and classification. Twelve characteristics were derived from a solitary electroencephalogram (EEG) recording. The retrieved characteristics were derived from statistical parameters obtained from the temporal and spectral domains of the EEG signal. They achieved an accuracy of 80.04% for the complete dataset. The authors conducted experiments on sleep state detection, utilizing various methods to classify sleep states. Ram et al. [19] implemented a Recurrent Convolutional Neural Network (RCNN) approach for this purpose, whereas Bjorvatn and Pallesen et al. [20] proposed a novel methodology for detecting sleep states. Additionally, Fraiwan and Lweesy et al. [21] focused on EEG recordings as the basis for their analysis. EEG is a type of signal that reflects the brain's electrical activity and is commonly used to analyze and interpret different sleep states. Despite employing different techniques and criteria, all studies share a common objective: to advance the detection and understanding of sleep states.

The motivation behind this research is the recognition of sleep as a key foundation of human health and well-being, which is crucial for cognitive function, emotional regulation, and overall physiological functions. Precise identification and categorization of sleep stages are crucial for accurate diagnosis and efficient treatment of sleep disorders. Nevertheless, the existing techniques frequently depend on subjective judgments or complicated procedures, underscoring the necessity for an inventive method. This study provides a strong foundation for sleep state recognition by utilizing breakthroughs in deep learning, namely the CRT model. The objective is to attain high accuracy in detecting sleep stages by analyzing large datasets. Identifying sleep onset and wakeup states holds significant importance in healthcare due to its direct relevance to timely interventions and personalized management of sleep disorders. By focusing on these particular transitions, healthcare professionals can promptly initiate suitable therapies or relaxation techniques when they identify the beginning of sleep, thereby enhancing sleep quality and effectively managing sleep-related problems. Customizing monitoring tactics to focus on these vital periods optimizes resource use and reduces extra data collection, which is particularly important in healthcare settings with limited resources or during ambulatory monitoring. Moreover, understanding individual sleep initiation and awakening patterns allows for tailoring therapies, resulting in enhanced efficacy of treatment strategies for sleep-related illnesses. This eventually leads to improved health outcomes and the long-term management of sleep disorders.

Our study achieved a significant milestone by pioneering an advanced sleep state detection method. In this system, a CRT model is developed that is specifically tailored for this purpose. The contributions of this study are as follows:

- Development of a novel *CRT* model dedicated to the effective identification of sleep states. This model presents a precise, fast, and fully automatic approach to classifying sleep states. Its superiority lies in its advanced ability to perform classification tasks more effectively than that of other existing methods. This model ensures high accuracy and significantly reduces the time required for classification, making it an invaluable tool in medical diagnostics.
- A novel, advanced, and complex layer based *CNN Transformer* model was created to construct a powerful model for sleep state classification. This novel function outlines the increased efficiency, driving the model towards faster and more accurate results.
- An advanced, high-precision sleep state classification system that outperforms existing solutions in terms of both speed and accuracy is developed. This system establishes notable detection standards, and sets a new benchmark for precision and efficiency in the field of sleep diagnostics.

This study was divided into several sections, each serving a specific purpose. In Section II, the research methodology is explained along with the technique used and the dataset analysis in the study. The results are explained in Section III, where the analyzes are presented. An in-depth discussion of the topic and a critical evaluation of the results and implications are provided in Section IV. Finally, Section V presents the study results and recommendations for future research directions.

II. Methodology of Research

A. Dataset Analysis and Preprocessing

This study obtained a dataset from the Child Mind Institute in New York, which is publicly available in Kaggle [22]. This dataset is designed to classify the two distinct sleep states. The sleep states were clearly labelled and classified within the dataset, which is freely accessible and has been developed specifically for research purposes. The dataset includes over 500 wrist-worn accelerometer data recordings over several days that were annotated with two event types: awakening, which marks the end of sleep, and onset, which marks the start of sleep. You are responsible for identifying when these two occurrences occur in the accelerometer series and totalling 987 megabyte. Every recording was carefully documented using two specific event categories: "onset", which indicates the beginning of sleep, and "wakeup", which indicates the end of sleep. The dataset includes various data types, such as numerical, object-oriented, and time series data. The dataset has 7,670,073 records spread across ten columns, providing a vast and comprehensive information collection.

Our data set included sleep duration data collected on different days of the week. The dataset detected outliers in the sleep duration data for six days. This study implemented preprocessing techniques for imputation (mean, median, or feature-derived value) to enhance the precision of the results and mitigate the influence of outliers. Fig. 1 shows the outliers associated with the sleep duration. Outliers in a dataset may have a detrimental effect on the classification process. To solve this problem, we used a series of preprocessing methods to identify and eliminate outliers, producing a more trustworthy dataset better suited for efficient categorization. In our preprocessing pipeline, a set of reliable techniques is utilized to improve the quality of our dataset. The min-max scaler maintained the original distribution while adjusting the values to fit within a specified range. In addition, mean imputation, standard deviation, and other methods were employed to

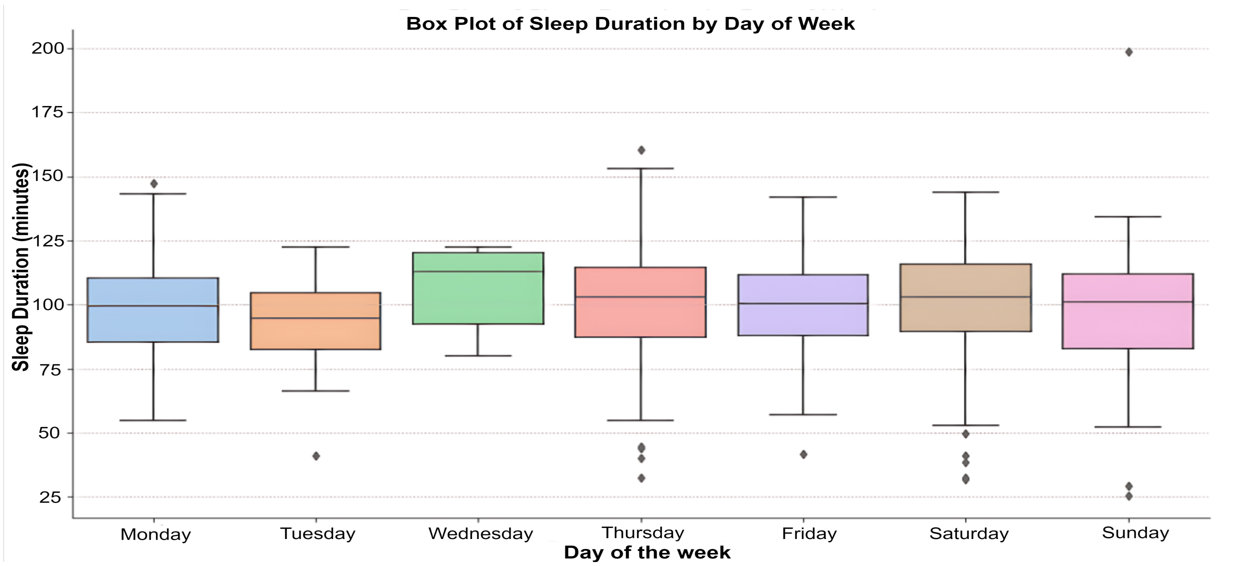


Fig. 1. Box Plot of Daily Sleep Durations The plot highlights the median, quartiles, and outliers in sleep durations, enabling the analysis of anomalies and weekly sleep patterns.

handle missing values in time series data. These measures fixed our dataset's missing data and made future research more reliable. The standard deviation can measure sample variability or dispersion around the mean.

This study includes the synthetic minority oversampling tool synthetic minority over-sampling technique (SMOTE) [23], [24] as a feature engineering tool, acknowledging the significance of managing unbalanced datasets. By interpolating between minority class samples that already exist, the SMOTE approach creates synthetic samples. Although they are incorporated into the training process, these extra samples are not directly utilized in the accuracy computation. To provide an objective evaluation of the performance, the accuracy was tested on a different test set that excluded the synthetic samples. SMOTE successfully achieves data distribution balance by intelligently producing synthetic samples for the minority class. Fig. 2 displays a graphical depiction of the balanced data. This system carefully partitioned data to provide a rigorous assessment of our model. For training, assigned 70% of the dataset points, corresponding to a total of 5,369,051 data points. The validation set, which accounted for 20% of the data, was created using 1,534,014 instances. The testing set, which comprised 10% of the dataset, consisted of 767,008 data points.

B. Execution of the CRT Model to Detect Sleep State

Our method employs a reliable algorithm based on the CRT model to identify the different sleep states precisely. In this system, a complex network was intentionally constructed for this model, guaranteeing that our approach is state-of-the-art and efficient for accurately detecting sleep phases. Fig. 3 depicts the complex structure of the CRT model utilized in our sleep state detection technique. Here, block (a) is the input block where num, dense, and concatenated operations are executed. From the input block, the connection links to block (b) that was positional encoding execution. The convolutional

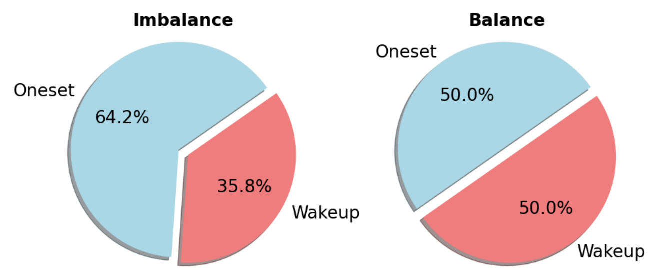


Fig. 2. The pie chart illustrates the proportions of balanced and unbalanced data.

block (c) were working that block execution processing from input block. After executing the convolutional block, the transformer block (d) works, and finally, the RNN block (e) will be executed from the output of our convolutional block.

1) Positional encoding block: First, the model takes data through an input block where the positional encoding function starts. This state entails utilizing a positional sequence function to establish the *"maxlen"* parameter. Here, n is a random number used to calculate the positional encoding (PE). The PE assigns a distinct position-based vector to each token in the input sequence, allowing the model to comprehend each element's sequential connection and location within the input data.

$$PE\left(pos, 2i\right) = Sin\left(\frac{pos}{n^{2i/d_{model}}}\right) \tag{1}$$

$$PE\left(pos, 2i+1\right) = Cos\left(\frac{pos}{n^{2i/d_{model}}}\right) \tag{2}$$

Here, *PE (pos,2i)* denotes the numerical representation of the positional encoding at a specific location and dimension, and *2i* represents the *pos* position in the sequence. *PE (pos,2i+1)* represents the value of the positional encoding at a particular position and dimension *2i+1* for the *pos* position in the sequence. The expression *PE (pos,2i+1)* denotes the specific value of the positional encoding at position *pos* and dimension

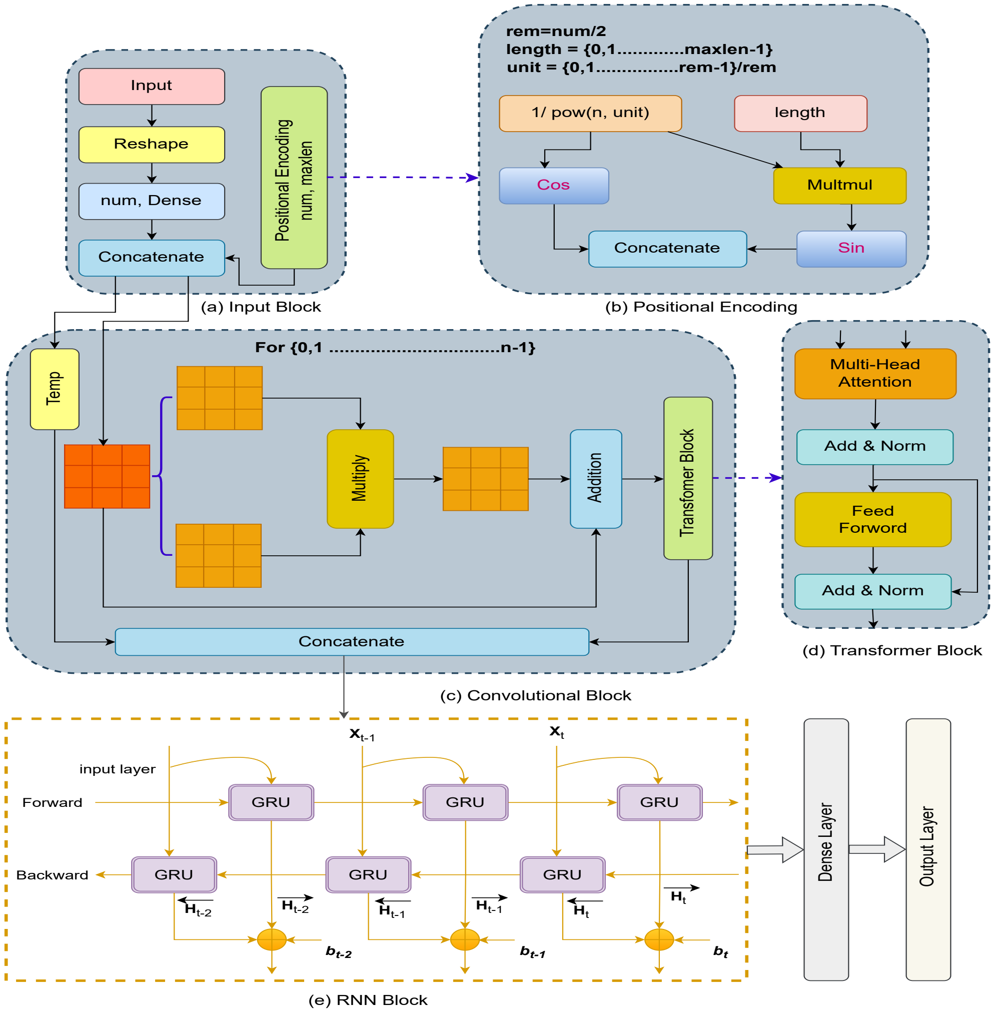


Fig. 3. Our proposed CRT model for sleep state identification, which integrates input, positional encoding, convolutional, transformer, and RNN components. It integrates spatial, temporal, and self-attention strategies to enhance identification.

'2i+1' in the sequence. *"pos"* represents the location inside the sequence. *"i"* denotes the dimension of the encoding. d_{model} represents the number of dimensions in the model.

2) Convolutional block: The input and position encoding outcome of encoding is concatenation, which follows the Convolutional block. Here, creating convolutional filters and other operations, NumPy is frequently utilized for practical numerical computations, including matrix operations and tensor manipulations. In order to improve gradient flow and preserve important information, addition is frequently used in designs with residual or skip connections to blend features from previous layers with the output of the current layer. By using concatenation to mix feature maps from several layers or routes, the model can effectively learn more complicated patterns by combining a variety of feature representations. These techniques enhance convolutional neural networks' performance and adaptability. This conv block controls the execution process: Consider $N * N$ square neuron layer and that our convolutional layer comes next. When $m * m$ filter ω were applied, the size of our convolutional layer's output will be $(N - m + 1) * (N - m + 1)$. To determine the input for a given unit prior to non-linearity x^l_{ij}, we need to combine the contributions from the cells in the previous layer in our layer.

$$x^l_{ij} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y^{l-1}_{(i+a)(j+b)} \tag{3}$$

Next, the nonlinearity function is applied by the convolutional layer.

$$y^l_{ij} = \sigma \left(x^l_{ij} \right) \tag{4}$$

Here, E is the output of each neuron output, $\left(\frac{\delta E}{\delta y^l_{ij}} \right)$. First, let us determine the gradient component for each weight by employing the chain rule. It is important to note that the chain rule must aggregate the contributions of all expressions in which the variable is present.

$$\frac{\delta E}{\delta \omega_{ab}} = \sum_{i=0}^{n-m} \sum_{j=0}^{n-m} \frac{\delta E}{\delta x^l_{ij}} \frac{\delta x^l_{ij}}{\delta \omega_{ab}} = \sum_{i=0}^{n-m} \sum_{j=0}^{n-m} \frac{\delta E}{\delta x^l_{ij}} y^{l-1}_{(i+a)(j+b)} \tag{5}$$

In this scenario, it is necessary to calculate the sum of all the x^l_{ij} expressions that contain the occurrence of ω_{ab}. The equality $\frac{\delta x^l_{ij}}{\delta \omega_{a,b}}$ can be deduced simply by examining the forward propagation equations. To calculate the gradient, it is necessary to determine the values. $\frac{\delta E}{\delta x^l_{i,j}}$ Calculating the deltas is rather simple because it involves using the chain rule once again:

$$\frac{\delta E}{\delta x^l_{ij}} = \frac{\delta E}{\delta y^l_{ij}} \frac{\delta y^l_{ij}}{\delta x^l_{ij}} = \frac{\delta E}{\delta y^l_{ij}} \frac{\delta}{\delta x^l_{ij}} \left(\sigma \left(x^l_{ij}\right)\right) = \frac{\delta E}{\delta y^l_{ij}} \acute{\sigma} \left(x^l_{ij}\right) \quad (6)$$

By leveraging our knowledge of the error in the present layer, $\frac{\delta E}{\delta y^l_{ij}}$, can efficiently calculate the deltas $\frac{\delta E}{\delta x^l_{ij}}$, in the same layer by employing the derivative of the activation function, $\acute{\sigma}(x)$. Given our knowledge of the mistakes at the present layer, poss all the necessary information is required to calculate the gradient of the weights employed by this convolutional layer.

To compute the weights for this convolutional layer, it is necessary to propagate the errors back to the previous layer. The chain rule can then be reemployed.

$$\frac{\delta E}{\delta y^{l-1}_{i,j}} = \sum_{a,b=0}^{m-1} \frac{\delta E}{\delta x^l_{(i-a)(j-b)}} \frac{\delta x^l_{(i-a)(j-b)}}{\delta y^{l-1}_{i,j}} = \frac{\delta E}{\delta x^l_{(i-a)(j-b)}} \omega_{ab} \quad (7)$$

Upon reviewing the forward propagation equations, it is evident that $\frac{\delta x^l_{(i-a)(j-b)}}{\delta y^{l-1}_{i,j}}$. This yields the above number for the fault in the preceding layer. The appearance bore some resemblance to that of convolution. The filter ω is applied to the layer, but instead of $x_{(i+a)(j+b)}$, it has $x_{(i-a)(j-b)}$. Furthermore, it is important to observe that the formula above applies only to locations with a minimum distance of m from the top and left boundaries. To rectify this issue, it is necessary to append zeros to the top and left borders. By doing that, conducting an essential convolution utilizing ω which reversed along both axes.

The max-pooling process performed on an input volume or feature map X, which has dimensions $W_{in} * H_{in} * D_{in}$ (width, height, and depth) following a convolutional operation. The pooling window size was F*F, and the stride is S. The equations control the output dimensions after applying max-pooling to the output of a convolutional layer similar to those used for standalone max-pooling.

$$W_{out} = \frac{W_{in} - F}{S} + 1 \quad (8)$$

$$H_{out} = \frac{H_{in} - F}{S} + 1 \quad (9)$$

$$H_{out} = D_{in} \quad (10)$$

$$MaxPooling\,(x) = max\,(x_1, x_2, ..., x_n) \quad (11)$$

3) Transformer block: As an encoder, the transformer block adds a max-pooling layer after the convolutional layer to reduce the feature map dimensions. Downsampling preserves convolutional filter information while reducing spatial dimensions. The self-attention mechanism helps link the sequence components: the self-attention network and MLP block encoding structure with a normalizing layer and residual connections. The outcome is obtained skillfully by combining the keys, value pairs, and searches. The compatibility function assigns weights to the items and calculates their corresponding weighted quantities. The dot product of all queries with keys calculated by dividing each query by $\sqrt{d_k}$, considering inputs with dimensions d_k of queries and keys and d_v. The weights assigned to the value pairs are determined by using the softmax algorithm. The queries *(Q)*, keys *(K)*, and values *(V)* required for the simultaneous calculation of the attention function comprise an attention matrix. The attention calculation *(Q, K, V)* is carried out as follows:

$$Attention\,(Q, K, V) = softmax\left(\frac{Q * K^T}{\sqrt{d_k}}\right) * V \quad (12)$$

Multi-headed attention allows the model to simultaneously attend to the input from many representation subspaces at various times.

$$MultiHead\,(Q, K, V) = Concat\,(head_1, ..., head_h) * \omega_0 \quad (13)$$

$$head_i = Attention\left(QW^Q_i, KW^K_i, VW^V_i\right) \quad (14)$$

A multilayer perceptron (MLP) is a feed-forward neural network consisting of dense and dropout layers. These MLPs have identical structures and are layered with layer blocks. Mathematical definition of each block;

$$Z = \sigma\,(XU)\,, \tilde{Z} = s\,(Z)\,, Y = \tilde{Z}V, \quad (15)$$

$$z_0 = \left[x_{class}; x^1_p E; x^2_p E, .., x^n_p E\right] + E_{pos}, E \epsilon R^{(N+1)*D} \quad (16)$$

$$z^1_l = MSA\,(LNz_{l-1}) + z_{l-1}, l = 1, ..., L \quad (17)$$

$$z_l = MLP\left(LNz^l l\right), z^l_1, l = 1, ...L \quad (18)$$

$$Z = LN\left(z^0_l\right) \quad (19)$$

The activation function denoted by σ, and the softmax function applied to the function represented by *s*(.), where U and V are the linear projection dimensions of the channel, respectively. The layer captures spatial interactions across records, as defined by Equation 15 while autonomously computing the individual tokens. Prior to stacking in Equation 15, the class token, patch encoding, and learnable encoding positions of the layers are thoroughly explained in Equations 16 and 17. Equation 19 represents the ultimate output of the encoder.

$$S\,(Z) = softmax\; f\,(Z) \quad (20)$$

4) RNN block: The gated recurrent unit GRU [25] process assumes that the number of hidden units is *"h"*. The small-batch input at a certain time step *"t"* may be represented as $x_t \epsilon\, R^{n*d}$, while the hidden state at the previous time step *"t-1"* is denoted as $h_{t-1} \epsilon\, R^{n*h}$. At the current time step t, the output hidden state h of a single GRU can be expressed as follows:

$$R_t = \sigma\,(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \quad (21)$$

$$Z_t = \sigma\,(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \quad (22)$$

$$\tilde{H}_t = tan\,h\,(X_t W_{xh} + (R_t \odot H_{t-1})\, W_{hh} + b_h) \quad (23)$$

$$H_t = (1 - Z_t)\, \odot H_{t-1} + Z_t \odot\, \tilde{H}_t \quad (24)$$

The activation function of the sigmoid represented as σ, $\sigma(x) = 1/1 + e^{-x}; W_{xr}, W_{hr}, W_{xz}, W_{hz}$ represent the weights linking the input layer and reset gate, hidden layer and reset gate, input layer and update gate, and hidden layer and update gate, respectively. The terms b_r *and* b_z refer to the bias values of the reset and update gates, respectively. H_t refers to the hidden condition at the present moment. Step $t; \odot$ denotes the process of multiplying two matrices. *Tanh* is a hyperbolic tangent activation function where neural networks introduce non-linearity and normalize output values between -1 and 1, aiding in better convergence and learning. It is defined using the following formula:

$$tanh\left(X\right) = 1 - \frac{2}{1 + e^{-2x}} \tag{25}$$

Together, the forward and back-propagating GRU units comprise a bidirectional gated recurrent unit (BiGRU), a type of neural network. The hidden layer state H_t is the BiGRU determined by the current input X_t, the output $\overrightarrow{H_t}$ is the forward hidden layer, and the output $\overleftarrow{H_t}$ is the backward hidden layer at time step *t-1*.

$$\overrightarrow{H_t} = GRU\left(X_t, \overrightarrow{H}_{t-1}\right) \tag{26}$$

$$\overleftarrow{H_t} = GRU\left(X_t, \overleftarrow{H}_{t-1}\right) \tag{27}$$

$$H_t = w_t \overrightarrow{H}_t + v_t \overleftarrow{H}_t + b_t \tag{28}$$

The GRU(.) function is used to apply a nonlinear transformation to the input water quality data. The input vector encoded into the GRU hidden state, represented by w_t *and* v_t. The weights of the forward hidden layer $\overrightarrow{H_t}$ and the backward hidden layer $\overleftarrow{H_t}$ of the BiGRU at time t correspond with these states. Indicates that b_t is the bias in the state of the buried layer at time *t* in addition. The final dense layer combines the input with positional encodings, convolutions, and transformer layer outputs to distinguish and categorize sleep states.

To comprehend sequential data, such as sleep states, a transformer module was included to capture global relationships and long-range dependencies. Positional encoding also guarantees that the model detects temporal patterns. Convolutional layers were employed prior to the RNN to efficiently concentrate on temporal dependencies to extract localized spatial information and lower input complexity. Combining convolutions to handle local patterns, transformers to capture global dependencies, and RNNs to describe sequential dynamics enables hierarchical feature extraction. Combined, these modules improve the precision, scalability, and computational effectiveness, offering a novel and reliable method of detecting sleep states.

5) Hyparameter tuning of proposed model: By measuring the system's optimal setting, we compiled a list of parameters for the hyperparameter tuning processes to our CRT model. Hyperparameter tuning is essential for optimizing the performance of a model by identifying the best configuration of the parameters learned during the training process. Finding the ideal configuration of hyperparameters, such as learning rate, batch size, and network architecture, is crucial for deep learning experiments to maximize model performance. Avoiding underfitting or overfitting contributes to increased efficiency, accuracy, and generalization. Proper tuning improved model convergence, speeding up training and producing better outcomes. This guarantees that the model successfully adjusts to specific inputs and task specifications. Ultimately, maximizing the potential of deep learning models requires hyperparameter optimization. In summary, it is guaranteed that the model reaches its highest level of performance by optimizing its structure and configurations. This study analyzed many factors, such as the learning rate, dropout, padding, optimizer, weight delay, batch size, epoch, activation, and kernel initializer. In this study, we systematically explore and implement different parameter values to identify the optimal configuration of our system. This ensures the system operates with maximum accuracy, speed, and efficiency. Table I displays the outcome of the hyperparameter tuning process for our CRT model. By conducting methodical experiments, hyperparameter tuning optimizes the model's performance, improving its ability to forecast outcomes and its resilience accurately. For hyperparameter tuning, learning rate, dropout, padding, optimizer, weight decay, batch size, epoch, activation, and kernel initializer are essential because they directly impact the effectiveness, performance, and generalizability of the model. The model's learning rate regulates how quickly it learns, while dropout stops overfitting by turning off the neurons. To achieve optimal convergence, the optimizer modifies the model weights and applies padding to ensure constant input dimensions. Batch size affects training efficiency, weight decay is a regularizer, and the number of epochs dictates how long the model is trained. Kernel initializers assist in establishing the initial weights, and the activation functions add non-linearity, all of which contribute to a finely calibrated, practical model.

TABLE I
PRESENTING A DEMONSTRATION OF HYPERPARAMETER TUNING VARIABLES IN THE CRT MODEL FOR OPTIMIZING MEASUREMENT.

Parameter	Search space	Selected Value
Learning rate	[0.001, 0.0001, 0.00001]	0.0001
Dropout	[0.02, 0.3, 0.4]	0.3
padding	[same, valid]	same
Optimizer	[SGD, Adam, RMSprop]	Adam
Weight delay	[0.0001-0.001]	0.002
Batch size	[32,16,8]	16
Epoch	[20]	20
Activation	[Sigmoid]	Sigmoid
Kernel Initilizer	[glorot, he, normal, uni-form]	glorot

III. RESULT ANALYSIS

This section presents a detailed description of the phases of sleep analysis. This study analyzed the performance of the model by presenting its accuracy. Furthermore, we present the receiver operating characteristic (ROC) graphs for both image scales generated by our CRT model. In addition, analyzed the loss function of the model was analyzed. In addition, this study incorporated accuracy, recall, f1-score, support metrics, and confusion matrices—the results generated by the models incorporated into the findings.

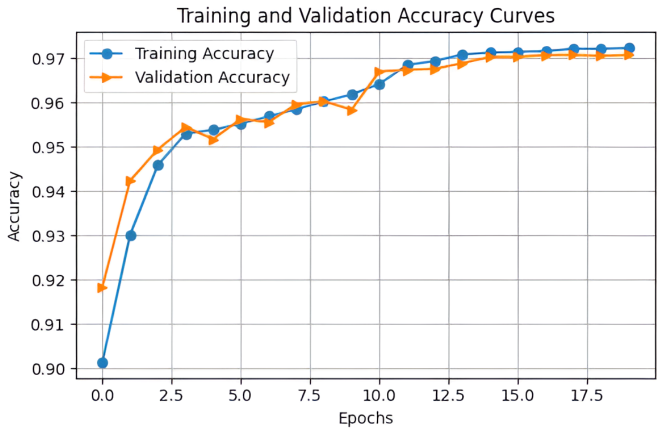


Fig. 4. The figure displays the accuracy curves for training and validation, demonstrating how the model's performance changes over epochs on known and unknown data, respectively.

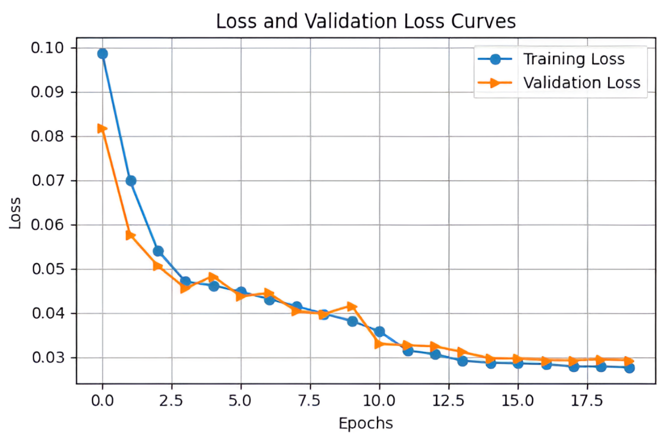


Fig. 5. The proposed CRT model demonstrates loss and validation loss curves. The curves depict the decrease in training loss and the validation performance.

To assess the performance of our sleep state detection model, we assessed it using the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) measures, among other important indicators. The ability of the model to accurately classify sleep phases is demonstrated using these data.

This study achieved excellent success in detecting sleep states. The CRT model has a remarkable training accuracy of 97.83%, with a notable validation accuracy of 97.03%. This accomplishment was achieved through rigorous training throughout 20 epochs. A succinct depiction of the training accuracy, validation accuracy, training loss, validation loss, and ROC curve is required for sleep state identification. Figs. 4 and 5 show the accuracy and loss of our model, respectively. The performance of a deep learning model on unknown data is evaluated by validation accuracy, whereas training accuracy gauges how well the model learns patterns from the training data. They guarantee that the model is adequately trained and can be generalized to real-world situations.

These observable indicators comprehensively understand the model's performance across the sleep state detection spectrum.

TABLE II
SHOWCASING THE PERFORMANCE MATRIX OF SEVERAL MODELS TO CHECKING OUR SYSTEM EFFICIENCY.

Model	Precision	Recall	F1-Score
Decision Tree	0.52	0.92	0.72
XGBoost Classifier	0.71	0.93	0.81
Logistic Regression	0.58	0.89	0.76
Gaussian NB	0.69	0.96	0.72
KNN Classifier	0.79	0.98	0.91
LGBM Classifier	0.65	0.96	0.68
AdaBoost Classifier	0.77	0.95	0.69
RNN	0.42	0.58	0.51
LSTM	0.52	0.69	0.61
RandomForest Classifier	0.79	0.97	0.85
Transformer	0.75	0.99	0.87
CRT (Proposed)	**0.96**	**0.99**	**0.98**

Precision, recall, and f1-score are crucial in deep learning because they serve as metrics for assessing a model's performance. It provides valuable information regarding the model's capability to accurately identify relevant instances (precision), capture all relevant instances (recall), and strike a balance between the two (f1-score). Table II lists the accuracies recall, precision, and f1-score of the implemented models. The precision of our CRT model was 0.96, the recall was 0.99, and the f1-score was 0.98. This refers to the accurate identification of the sleep stage.

A confusion matrix is an essential machine-learning method for assessing the efficacy of a detection model. The text fully explains the detection capabilities of the model and the degree to which they correspond to real class names. The matrix is often organized in a tabular format, with rows and columns representing the categorized and actual classes. We utilized the test data from our dataset to evaluate the confusion matrix. The dataset divided into training (70%), validation (20%), and test (10%) subsets. With a total of 767,008 data points, the test set was used exclusively to evaluate the confusion matrices. Fig. 6 illustrates the confusion matrix of the two classes. The ground truth in a confusion matrix is the actual, true labels of the data as established by a trustworthy source, such as real-world observations, expert annotations, or established benchmarks. It acts as the benchmark or standard by which predictions made by the model are evaluated. The confusion matrix's TP, TN, FP, and FN identified by contrasting the model's predictions with the ground truth. Accurate ground truth data are essential when assessing a model's performance because they guarantee that the confusion matrix offers an unbiased evaluation of its accuracy in classifying instances and areas of mistake.

This study further analyzed sleep state detection research further. By tested the decision tree, XGBoost, RNN, LSTM, random forest, transformer, GaussianNB, KNN, LGBM, logistic regression, and AdaBoost Classifier for the system performance study. Notably, our findings were consistent with those of the previous. CRT proves that our model can handle this crucial task even with higher-quality results. Table III displays the various models we implemented for sleep state detection and the proposed model. The accuracy rates of 97.83% for training, 97.07% for validation, and 97.23% for testing

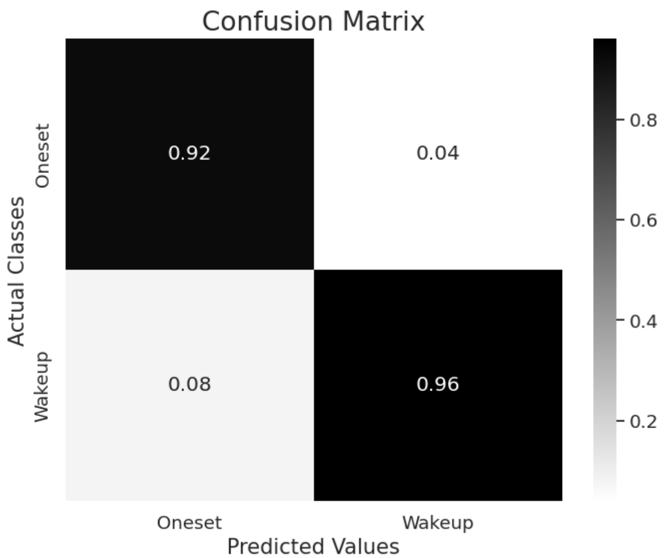


Fig. 6. The confusion matrix is a tool that compares the predicted labels of our suggested model to the actual ground truth labels in many categories. It provides a clear representation of the categorization outcomes.

TABLE III
AN ANALYSIS OF CLASSIFIER MODEL'S VALIDATION AND TRAINING ACCURACY IS CONDUCTED TO ASSESS THEIR ROBUSTNESS AND AID IN MODEL SELECTION.

Model	Train	Validation	Testing
Decision Tree	88.42%	87.04%	87.72%
XGBoost Classifier	86.78%	86.83%	86.55%
Logistic Regression	80.70%	80.71%	79.61%
Gaussian NB	69.38%	69.42%	70.69%
KNN Classifier	93.23%	90.12%	90.71%
LGBM Classifier	86.54%	86.55%	85.87%
AdaBoost Classifier	85.79%	85.96%	85.52%
RNN	56.06%	56.02%	56.02%
LSTM	62.49%	61.96%	62.21%
RandomForest Classifier	96.94%	91.65%	90.27%
Transformer	94.32%	95.68%	91.86%
CRT (Proposed)	**97.83%**	**97.07%**	**97.23%**

achieved by our CRT model signify its robust performance for accurately detecting sleep states. These high accuracies demonstrate the model's proficiency in correctly identifying sleep patterns during training and when applied to unseen data, indicating its reliability and potential utility in healthcare applications. This finding also suggests that the CRT model can effectively trained to generalize patterns within the data without encountering the significant issues of underfitting or overfitting.

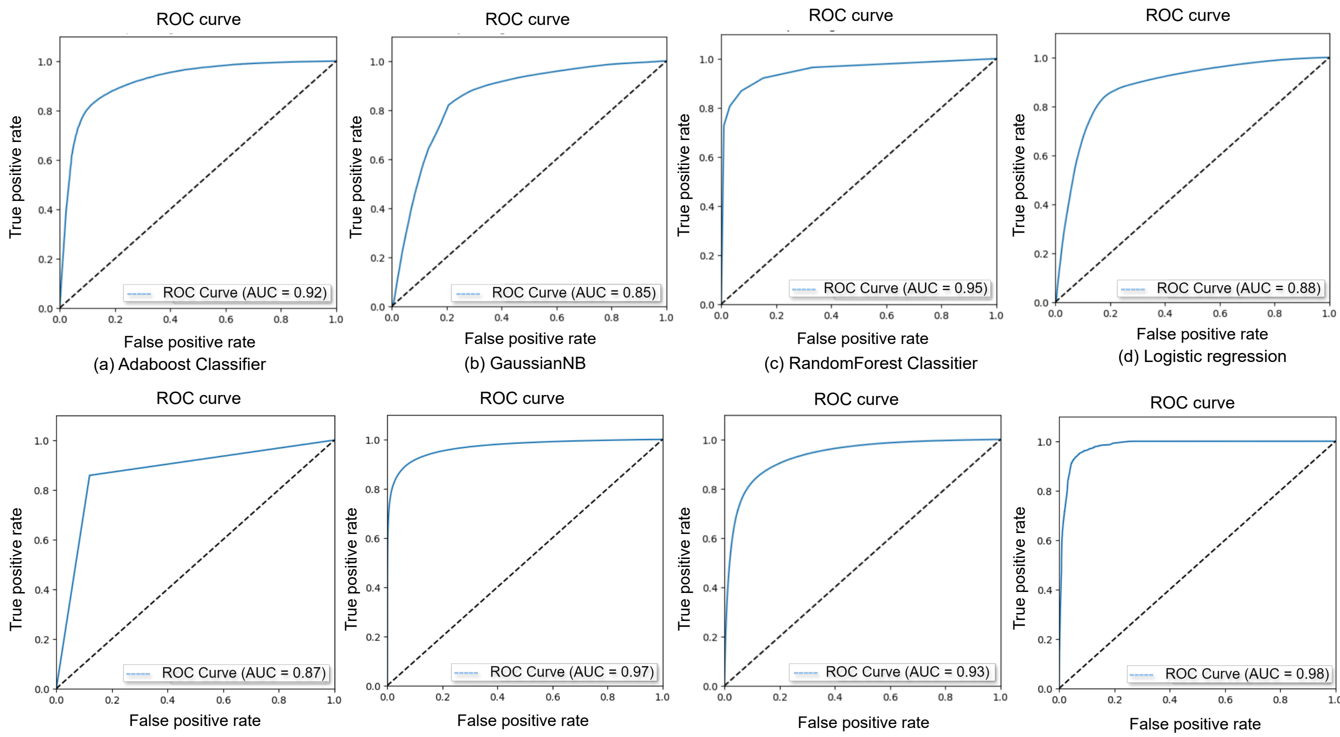
In this study, k-fold cross-validation, essential for profound learning studies, was employed to guarantee that the model's performance is resilient and effectively generalizes unfamiliar data. This study applied 10-fold cross-validation. 10-fold cross-validation offers a fair evaluation of model performance by utilizing distinct subsets of the data for training and testing, hence minimizing variability in the assessment. Table IV displays the results of the 10-fold cross-validation for the CRT model. To execute 10-fold cross-validation, 90% of the data was allocated for training and validation purposes,

TABLE IV
SHOWCASING THE PROPOSED CRT MODEL'S K-FOLD CROSS-VALIDATION RESULTS.

Fold	Train Acc	Validation Acc	Testing Acc	St. Dev	Mean
1	96.65%	95.96%	96.19%	0.064	0.885
2	97.39%	96.45%	96.67%	0.038	0.920
3	98.78%	97.92%	96.96%	0.042	0.926
4	99.56%	98.47%	97.68%	0.021	0.945
5	99.78%	99.35%	97.35%	0.014	0.953
6	99.45%	99.41%	97.92%	0.002	0.968
7	99.86%	99.76%	98.45%	0.004	0.973
8	99.52%	99.78%	99.22%	0.008	0.976
9	99.89%	99.86%	99.35%	0.005	0.984
10	99.89%	99.82%	99.57%	0.002	0.993

whereas the remaining 10% was reserved for testing. This experiment divided the dataset into k subsets, where the model was trained k times. Each time, a different subset is used as the validation set, while the remaining subsets serve as the training set. This aids in reducing overfitting and offers a thorough assessment of the model's correctness. Calculating the mean of the results from all folds provides a more dependable estimation of the model's performance. The testing accuracy was evaluated using k-fold cross-validation, and for each fold, the standard deviation (St. Dev) and the mean of the accuracy values calculated. These metrics provide insights into a model's performance's consistency and central tendency across the folds. Table IV presents the standard deviation and mean values computed for each fold, illustrating the variability and average performance of the cross-validation process. Specific patterns suggest superior performance when assessing the effectiveness of a DL model using accuracy, standard deviation, and mean value. Indicating consistency in the performance of the model, a reduced standard deviation ensures that its predictions are steady and not unduly affected by changes in the data. The objective of an effective model is to achieve high precision and consistency by maximizing mean accuracy and decreasing standard deviation. Taken together, they offer a thorough understanding of these measurements dependability of a model.

This study compared the proposed model with existing approaches in the field of sleep state identification. The evaluation demonstrated that our model surpassed previous approaches and correctly identified sleep phases better. Table V provides a concise overview of the thorough comparison between our suggested model and the current methods, emphasizing the notable progress made by our approach. Casal et al. [26] developed a network architecture using the TCN-Transformer method to classify sleep stages (awake or asleep) based solely on heart rate (HR) signals from a pulse oximeter. Huang et al. [27] proposed an automatic sleep staging model incorporating an improved attention module and a hidden Markov model (HMM). They leveraged single-channel EEG data and implemented the SENet model. Phan et al. [28] introduced a sequence-to-sequence sleep staging approach and employed the SleepTransformer method. Dutt et al. [29] conducted multiclass sleep stage classification using EEG data and proposed an explainable unified CNN-CRF



(a) Adaboost Classifier (b) GaussianNB (c) RandomForest Classitier (d) Logistic regression

(e) Decision tree classifier (f) Transformer (g) XGBoost Classifier (h) CRT

Fig. 7. The ROC curves illustrate the performance of the top eight models employed in this study for classification tasks. Each curve illustrates a model's capacity to differentiate by graphing the rate of false positive (1 - specificity) against the rate of genuine positive (sensitivity) at different decision thresholds.

approach. Alam et al. [30] focused on sleep apnea detection, utilizing inter-band energy ratio features extracted from multi-band EEG signals and employing a Linear Support Vector Machine (LSVM) classifier. Abasi et al. [31] studied sleep apnea detection and used the CNN method, while Javeed et al. [32] predicted sleep apnea based on electronic health data using the XGBoost model. Almarshad et al. [33] enhanced the diagnostic performance of oximetry for obstructive sleep apnea, aiming to reduce the time and costs associated with traditional polysomnography by employing a Transformer model. Bahrami et al. [34] processed and segmented data for sleep apnea detection and applied machine learning and deep learning methods using the ZFNet-BiLSTM model for their research. Hu et al. [35] proposed a hybrid Transformer model based on the self-attention mechanism for sleep apnea detection using single-lead electrocardiogram (ECG) data. Fernandez et al. [36] focused on sleep stage scoring using two-channel EEG signals with a CNN method, whereas Zhu et al. [37] proposed a CNN combined with an attention mechanism for automatic sleep staging. From the above analyses, our CRT model achieved higher test accuracy than the existing methods and demonstrated superior reliability.

The ROC curve visually displayed the sensitivity and specificity of the top eight models at various threshold settings. The ROC curve results demonstrate the superior performance of our proposed CRT model, achieving an AUC of 0.98, which surpasses that of all other implemented models. Comparatively, the ROC curve results for the other classifiers are as follows: Transformer (0.97), Random Forest (0.95), XGBoost (0.93), AdaBoost (0.92), Logistic Regression (0.88), Decision Tree (0.87), and GaussianNB (0.85). These results highlight the exceptional discrimination ability of our CRT model compared with existing approaches. Fig. 7 provides a clear and visually imploring representation of the ROC curve, allowing a better understanding of how the models perform under different thresholds. The ROC curve highlights the trade-off between sensitivity and false positive rates by evaluating the classification performance of a model across thresholds. It provides information on the discrimination capacity of the model and is essential for unbalanced datasets. A higher AUC indicates better overall performance.

TABLE V
EVALUATION OF THE PROPOSED SLEEP STATE DETECTION APPROACH IN COMPARISON TO OTHER MODELS AND DATASETSMET

Author & year	Technique Employed	Accuracy
Casal et al. [26] 2022	TCN - Transformer	90.0%
Huang et al. [27] 2022	SENet	84.6%
Phan et al. [28] 2022	SleepTransformer	84.9%
Dutt et al. [29] 2023	CNN-CRF	86.8%
Alam et al. [30] 2024	LSVM	94.81%
Abasi et al. [31] 2024	CNN	90.92%
Javeed et al. [32] 2023	XGBoost_BiLSTM	97.0%
Almarshad et al. [33] 2023	Transformer	80.0%
Bahrami et al. [34] 2022	ZFNet-BiLSTM	88.13%
Hu et al. [35] 2022	Hybrid Transformer	90.5%
Fernandez et al. [36] 2020	CNN	92.7%
Zhu et al. [37] 2020	Attention CNN	93.7%
Our work 2024	**CRT**	**97.83%**

The measurement results obtained from our CRT model are outstanding. Throughout the training process with the CRT model, an exceptional accuracy rate of 97.83%. Similarly, the validation and testing sets resulted in remarkable accuracies of 97.07% and 97.23%, respectively. Additionally, the ROC curve graphically represents the relationship between the true and false positive rates across different threshold values. These reliable outcomes emphasize the effectiveness of our technique in detecting sleep states. This promising achievement offers excellent potential in the medical field, indicating substantial advancements in enhancing sleep monitoring and diagnosis.

IV. Discussion

The investigation of sleep states in healthcare is crucial. Based on the analysis in the results section, our model demonstrates a higher accuracy level than previous studies. The existing strategy consists of a single or hybrid model with strong capabilities. This study compared the findings and approaches of previous studies in Table V, emphasizing the superior performance of our proposed model. Although several authors have used different strategies to obtain encouraging results, our new model performs better than the traditional tried-and-true techniques, showing greater accuracy and more useful results. This development demonstrates that our strategy can produce better outcomes in a specified field. However, our CRT model is more complex and reliable owing to the utilization of several advanced block strengths. The technical novelty of the proposed model is found in its creative fusion of RNN components, transformer blocks, convolutional layers, and positional encoding into a single architecture. The advantages of each model type were successfully combined in this hybrid technique to handle challenging tasks involving sequential and spatial data. Positional encoding guarantees that the model records the sequential order of the inputs, and the convolutional layers effectively extract local features, especially in spatially structured data. The transformer block's self-attention mechanism helps the model focus on pertinent segments of the input sequence, thereby improving its comprehension of global contextual linkages. Furthermore, by keeping track of prior inputs for sequential tasks, the RNN block allows the model to handle time-dependent data. These potent elements work together to provide our model with the ability to manage sequential dependencies, capture both local and global patterns, and enhance performance on various tasks. Our system has some limitations. Currently, our models are not designed for image classification and segmentation tasks. In addition, the system is limited to classifying only two sleep states. However, if the number of classes were to be increased, the model would become more effective in accurately classifying a wider range of sleep state types. The total parameter count for our CRT model is 0.3 million, and the number of trainable parameters is similarly 0.3 million. Despite our parameter count, precise results were obtained using a mix of three reliable blocks. This study performed complex operations within these three blocks to construct a highly advanced neural architecture. After a thorough analysis, the computational setup of our model was confirmed. A Windows 10 computer with an Intel(R) Core(TM) i7 CPU, 32GB of RAM, and a 12GB GPU was used for the training process. All offensive automatic traffic reduction models were implemented using TensorFlow 2.2.1 and Python 3.12.3. A notebook environment was effectively utilized to manage Python libraries, such as TensorFlow, which are frequently used for creating sophisticated models. Notably, our model required 36 min and 55 sec for training. However, the limitations of our model include its inability to process pictures, audio, and video. Additionally, our CRT model was specifically designed for classification tasks. The application of deep learning for sleep state detection, explicitly targeting the onset and wakeup times, holds significant clinical value. Accurate identification of these key sleep stages can revolutionize the management of sleep disorders such as insomnia, sleep apnea, and circadian rhythm disorders. By providing high accuracy and precise detection, this technology can facilitate personalized treatment plans, improve the efficacy of therapeutic interventions, and enhance patient monitoring both in clinical settings and at home. Additionally, it can aid research by offering reliable data for studying sleep patterns and the impact of various treatments on sleep quality. In the future, we plan to address this constraint and provide an alternative model capable of detecting more than two sleep phases. This detection is crucial in the healthcare domain because it aids in diagnosing sleep disorders, ensuring proper treatment, and promoting overall well-being. Accurate monitoring of sleep states enables personalized healthcare interventions and improves patient outcomes and quality of life (QOL).

V. Conclusion and Future Work

This study accurately identified sleep state as the onset of falling and waking up. The uniqueness of our work lies in the creation of an automated system that is built explicitly to identify sleep states. Our detection technology accurately and precisely recognizes complex sleep state patterns. The outcomes are indicative, and this study achieved a high degree of precision in the categorizing procedure, ensuring careful state identification. In addition, our detection technique demonstrates outstanding accuracy, precision, and testing accuracy. Implementing this system to detect sleep states represents significant progress in medical healthcare technology. The potential of this technology to enhance diagnosis and care in the healthcare industry is substantial, as it provides novel prospects for enhanced patient assessment and treatment alternatives.

In the future, we aim to address these limitations of the proposed system. We plan to expand our dataset to include more classes, enabling the model to classify a broader range of sleep state types and enhance its effectiveness. Additionally, we are developing our model to perform diverse tasks, including image classification and segmentation, creating a more versatile and efficient system.

Acknowledgment

The authors would such as to thank the Advanced Machine Intelligence Research (AMIR) Lab, Dhaka, Bangladesh, for its supervision and resources.

DATA AVAILABILITY

Data is available in a publicly accessible link: https://www.kaggle.com/c/child-mind-institute-detect-sleep-states/data

REFERENCES

[1] H. Alsolai, S. Qureshi, S. M. Z. Iqbal, S. Vanichayobon, L. E. Henesey, C. Lindley, and S. Karrila, "A systematic review of literature on automated sleep scoring," *IEEE Access*, vol. 10, 2022.

[2] E. Van Cauter, K. Spiegel, E. Tasali, and R. Leproult, "Metabolic consequences of sleep and sleep loss," *Sleep Medicine*, vol. 9, pp. S23–S28, 2008.

[3] C. A. Czeisler, "Duration, timing and quality of sleep are each vital for health, performance and safety," *Sleep Health: Journal of the National Sleep Foundation*, vol. 1, no. 1, pp. 5–8, 2015.

[4] K. M. Gunter, A. Brink-Kjær, E. Mignot, H. B. Sørensen, E. During, and P. Jennum, "Svit: A spectral vision transformer for the detection of rem sleep behavior disorder," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, pp. 4285–4292, 2023.

[5] G. Rauchs, B. Desgranges, J. Foret, and F. Eustache, "The relationships between memory systems and sleep stages," *Journal of Sleep Research*, vol. 14, no. 2, pp. 123–140, 2005.

[6] Y. Dai, X. Li, S. Liang, L. Wang, Q. Duan, H. Yang, C. Zhang, X. Chen, L. Li, and X. Li, "Multichannelsleepnet: A transformer-based model for automatic sleep stage classification with PSG," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 9, pp. 4204–4215, 2023.

[7] A. D. Laposky, J. Bass, A. Kohsaka, and F. W. Turek, "Sleep and circadian rhythms: key components in the regulation of energy metabolism," *FEBS Letters*, vol. 582, no. 1, pp. 142–151, 2008.

[8] M. Mostafa Monowar, S. N. Nobel, M. Afroj, M. A. Hamid, M. Z. Uddin, M. Kabir, and M. Mridha, "Advanced sleep disorder detection using multi-layered ensemble learning and advanced data balancing techniques," *Frontiers in Artificial Intelligence*, vol. 7, p. 1506770.

[9] J. W. Cho and J. F. Duffy, "Sleep, sleep disorders, and sexual dysfunction," *The world journal of men's health*, vol. 37, no. 3, pp. 261–275, 2019.

[10] S. Stranges, W. Tigbe, F. X. Gómez-Olivé, M. Thorogood, and N.-B. Kandala, "Sleep problems: An emerging global epidemic? Findings from the indepth who-sage study among more than 40,000 older adults from 8 countries across Africa and Asia," *Sleep*, vol. 35, no. 8, pp. 1173–1181, 2012.

[11] A. Sano, W. Chen, D. Lopez-Martinez, S. Taylor, and R. W. Picard, "Multimodal ambulatory sleep detection using LSTM recurrent neural networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1607–1617, 2018.

[12] 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, pp. 1587–1597, 2016.

[13] N. Ying, H. Li, Z. Zhang, Y. Zhou, H. Chen, and M. Yang, "ViTDFNN: A vision transformer enabled deep fuzzy neural network for detecting sleep apnea-hypopnea syndrome in the internet of medical things," *IEEE Transactions on Fuzzy Systems*, vol. Early Access, pp. 1 – 12, 2024.

[14] J. Corral-Peñafiel, J.-L. Pepin, and F. Barbe, "Ambulatory monitoring in the diagnosis and management of obstructive sleep apnoea syndrome," *European Respiratory Review*, vol. 22, no. 129, pp. 312–324, 2013.

[15] S. Swapno, S. Nobel, P. Meena, V. Meena, A. T. Azar, Z. Haider, and M. Tounsi, "A reinforcement learning approach for reducing traffic congestion using deep Q learning," *Scientific Reports*, vol. 14, no. 1, pp. 1–20, 2024.

[16] S. N. Nobel, S. M. R. Swapno, M. R. Islam, M. Safran, S. Alfarhood, and M. Mridha, "A machine learning approach for vocal fold segmentation and disorder classification based on ensemble method," *Scientific reports*, vol. 14, no. 1, p. 14435, 2024.

[17] S. M. R. Swapno, S. N. Nobel, P. Meena, V. Meena, J. Bahadur, and A. Appaji, "Accelerated and precise skin cancer detection through an enhanced machine learning pipeline for improved diagnostic accuracy," *Results in Engineering*, p. 104168, 2025.

[18] S. N. Nobel, S. M. R. Swapno, M. B. Islam, A. Azad, S. A. Alyami, M. Alamin, P. Liò, and M. A. Moni, "A Novel Mixed Convolution Transformer Model for the Fast and Accurate Diagnosis of Glioma Subtypes," *Advanced Intelligent Systems*, p. 2400566, 2024.

[19] Z. J. A Ram, S Keshari, "Annotating sleep states in children from wrist-worn accelerometer data using machine learning," *arXiv preprint arXiv:2312.07561*, 2023.

[20] B. Bjorvatn and S. Pallesen, "A practical approach to circadian rhythm sleep disorders," *Sleep Medicine Reviews*, vol. 13, no. 1, pp. 47–60, 2009.

[21] L. Fraiwan and K. Lweesy, "Neonatal sleep state identification using deep learning autoencoders," in *2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA), Penang, Malaysia*, 2017, pp. 228–231.

[22] "Child Mind Institute - Detect Sleep States," https://www.kaggle.com/c/child-mind-institute-detect-sleep-states/data, [Accessed 29-12-2023].

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[24] S. N. Nobel, S. Sultana, S. P. Singha, S. Chaki, M. J. N. Mahi, T. Jan, A. Barros, and M. Whaiduzzaman, "Unmasking banking fraud: Unleashing the power of machine learning and explainable ai (xai) on imbalanced data," *Information*, vol. 15, no. 6, p. 298, 2024.

[25] M. B. R. Bappi, S. M. R. Swapno, and M. F. Rabbi, "Skin cancer disease detection using mcd-gru: A deep learning approach," in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*. IEEE, 2024, pp. 445–450.

[26] R. Casal, L. E. Di Persia, and G. Schlotthauer, "Temporal convolutional networks and transformers for classifying the sleep stage in awake or asleep using pulse oximetry signals," *Journal of Computational Science*, vol. 59, p. 101544, 2022.

[27] 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.

[28] H. Phan, K. Mikkelsen, O. Y. Chén, P. Koch, A. Mertins, and M. De Vos, "Sleeptransformer: Automatic sleep staging with interpretability and uncertainty quantification," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 8, pp. 2456–2467, 2022.

[29] 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. 16 830–16 843, 2023.

[30] M. S. Alam, Y. Siddiqui, M. Hasan, O. Farooq, and Y. Himeur, "Energy-efficient FPGA based sleep apnea detection using EEG signals," *IEEE Access*, vol. 12, pp. 40 182–40 195, 2024.

[31] A. K. Abasi, M. Aloqaily, and M. Guizani, "Bare-bones based honey badger algorithm of CNN for sleep apnea detection," *Cluster Computing*, pp. 1–21, 2024.

[32] A. Javeed, J. S. Berglund, A. L. Dallora, M. A. Saleem, and P. Anderberg, "Predictive power of xgboost_bilstm model: a machine-learning approach for accurate sleep apnea detection using electronic health data," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 188, 2023.

[33] M. A. Almarshad, S. Al-Ahmadi, M. S. Islam, A. S. BaHammam, and A. Soudani, "Adoption of transformer neural network to improve the diagnostic performance of oximetry for obstructive sleep apnea," *Sensors*, vol. 23, no. 18, p. 7924, 2023.

[34] M. Bahrami and M. Forouzanfar, "Sleep apnea detection from single-lead ecg: A comprehensive analysis of machine learning and deep learning algorithms," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[35] S. Hu, W. Cai, T. Gao, and M. Wang, "A hybrid transformer model for obstructive sleep apnea detection based on self-attention mechanism using single-lead ECG," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[36] E. Fernandez-Blanco, D. Rivero, and A. Pazos, "Convolutional neural networks for sleep stage scoring on a two-channel EEG signal," *Soft Computing*, vol. 24, pp. 4067–4079, 2020.

[37] T. Zhu, W. Luo, and F. Yu, "Convolution-and attention-based neural network for automated sleep stage classification," *International Journal of Environmental Research and Public Health*, vol. 17, no. 11, p. 4152, 2020.