Ph.D. DISSERTATION

Sleep Stage Classification and Disorder Prediction Using Simple Sensors and End-to-end Trainable Deep Neural Networks

간단한 센서와 종단간 학습가능한 깊은 신경망을 이용하는 수면 단계 분류 및 장애 예측

BY

IKSOO CHOI

AUGUST 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Sleep Stage Classification and Disorder Prediction Using Simple Sensors and End-to-end Trainable Deep Neural Networks

간단한 센서와 종단간 학습가능한 깊은 신경망을 이용하는 수면 단계 분류 및 장애 예측

BY

IKSOO CHOI
AUGUST 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Sleep Stage Classification and Disorder Prediction Using Simple Sensors and End-to-end Trainable Deep Neural Networks

간단한 센서와 종단간 학습가능한 깊은 신경망을
이용하는 수면 단계 분류 및 장애 예측

지도교수 심 병 효

이 논문을 공학박사 학위논문으로 제출함

2023년 8월

서울대학교 대학원

전기 정보 공학부

최 익 수

최익수의 공학박사 학위 논문을 인준함

2023년 8월

위 원 장:	조 남 익
부위원장:	심 병 효
위 원:	성 원 용
위 원:	정 교 민
위 원:	최 정 욱

Abstract

Sleep is essential for both physical and psychological health, as poor sleep quality can potentially undermine cognitive abilities, learning, memory, and even lead to depression. Given the multitude of personal factors that can contribute to sleep disorders, creating reliable and easy-to-use models for predicting sleep stages and detecting sleep disorders is a difficult task. This dissertation seeks to contribute to this field by developing a deep neural network (DNN) architecture that leverages principles and techniques from automatic speech recognition (ASR) to automate sleep stage classification.

We initially compared the performance of various DNN architectures using poly somnography (PSG) signals, aiming to circumvent the need for unwieldy electroencephalogram sensors. Subsequently, we developed a DNN model for sleep stage classification using electrocardiogram (ECG) signals. To overcome the challenges inherent in using ECG signals for sleep stage classification, we created a DNN architecture that integrates a feature extraction-sequence modeling system akin to that utilized in ASR. This system processes overnight sleep sequences to capture the cyclical characteristics of sleep stages.

Furthermore, we developed sleep models inspired by language models from natural language processing. These sleep models focus on the sequence of sleep stages to compensate for the information deficit arising from the use of limited signal sources. They also enhance the accuracy of the automatic sleep stage classification systems by decoding classified sleep stages.

Moreover, we designed a model to classify apnea events from audio signals captured during natural sleep. This model can differentiate between four types of apnea events. Though identifying brain-related apnea events based solely on sound presents significant challenges, other events can be successfully classified during normal sleep.

We used the model's prediction results to estimate the apnea-hypopnea index, further calibrating these estimates for increased accuracy.

Overall, this dissertation contributes to the development of an efficient DNN architecture for automated sleep stage classification using ECG signals, as well as an apnea classification model from audio signals. These models can assist clinicians in diagnosing and treating sleep disorders, thereby improving sleep wellness at an individual level. We specifically focused on signals that, while perhaps limited and less informative, are comfortable, highly accessible, and easily applied on mobile devices.

keywords: polysomnography, sleep stage, obstructive sleep apnea, sequence modeling

student number: 2014-21751

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation and Subject of Research

Sleep wellness is a crucial aspect of physical and psychological health for all individuals. Sleep deprivation or poor sleep quality can negatively impact cognitive abilities, including learning and memory, and lead to depression. Multiple factors can cause sleep disturbances, including mental health conditions, vascular diseases, medication or alcohol use, and lifestyle patterns. Clinical psychology therapy often requires patients with sleep disorders to take sleep tests to identify the underlying causes and determine appropriate medical treatments.

However, manual classification of sleep stages using polysomnography (PSG) is a time-consuming and labor-intensive process that requires highly trained specialists. Furthermore, it subjects patients to significant discomfort. As the number of individuals undergoing sleep tests increases, the need for efficient and automated sleep stage classification methods becomes more pressing.

Even for those without sleep disorders, personal factors such as age, gender, and lifestyle can affect the ratio or transitions between sleep stages, making it challenging to develop a universal and precise sleep-stage prediction model. As a result, research efforts are focused on developing a robust deep neural network (DNN) architecture

for sleep stage classification that incorporates knowledge from other domains, such as automatic speech recognition (ASR).

This dissertation aims to contribute to this research field by developing an effective DNN architecture for automated sleep stage classification. The study will leverage insights from ASR and apply them to PSG data analysis to improve the accuracy and efficiency of sleep stage classification. The ultimate goal is to develop a system of simple devices that can be used every day to assist individuals in managing their sleep.

The sleep stages are generally categorized into 5 stages, wake (W), rapid eye movement (REM), and 3 states of non-REM (N1, N2, and N3) [7, 8]. In adults, the expected proportion of sleep stages excluding the W stage is 5%, 50%, 30%, and 25% for the N1, N2, N3, and REM stages [9].

N1 sleep stages appear in the transition from wakefulness to sleep. Therefore, the quantity or the percentage of the N1 sleep stage is related to sleep fragmentation caused by sleep apnea, snoring, or surrounding environments. N2 sleep stages are a large part of sleep times and appear after the N1 sleep stage. The N2 sleep stage increases by sleep fragmentation, obstructive sleep apnea, medication effect, or age-related sleep patterns. The N3 sleep stage is also known as the deep sleep stage. Sleepwalking and drowsiness occur during the N3 sleep stage, and the N3 stages may increase as the rebound of frequent sleep disturbances. REM sleep generally appears every 90 to 120 minutes as an indicator of a sleep cycle. Although the exact function of the REM sleep stage is still unconcluded, REM sleep stages are an essential part of sleep, consuming one-quarter of sleep time. Dream-enacting behavior, nightmares, and sleep apnea emerge during REM sleep.

There are many causes of sleep disruptions, such as drugs, cardiac disses, sleep apnea related to snoring, psychological aspect, or environmental factors. Figure 1.1 shows the sleep stages of 2 subjects. One epoch in this figure corresponds to a 30-second period. The sleep stage transitions shown above demonstrate repeated switches between wakefulness (W) and N1 stages, indicating sleep disturbances. Sleep stages

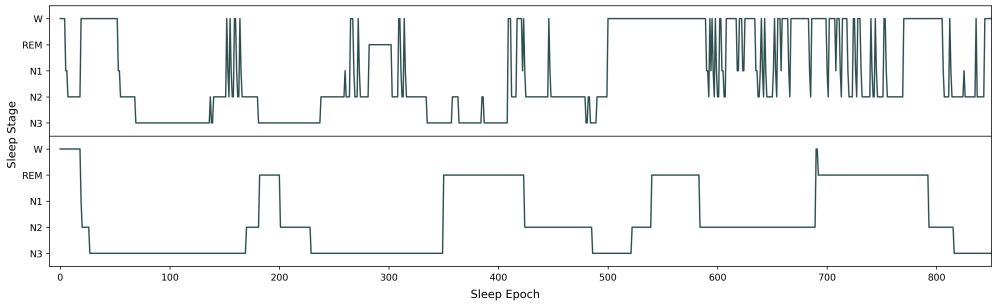


Figure 1.1: Examples of sleep stages.

from the other subject are relatively stable. It shows non-REM to REM stage cycles and only 1 epoch of the W stage during sleep.

1.2 Background Information of PSG Signals

The PSG signal is a crucial diagnostic tool for identifying sleep-related disorders. It involves the recording of vital signs during an overnight sleep study, and the primary signals recorded include Electroencephalogram, Electrooculogram, Electromyogram, and Electrocardiogram. These signals help doctors assess the activity of the brain, eyes, chin, and heart, respectively [10]. In addition, other sensors such as airflow, respiration, body position, and leg movements may be used to provide further insight into the patient's sleep patterns.

PSG data is analyzed by qualified experts who divide the recording into 30-second intervals known as epochs. During each epoch, the sleep stage is identified based on the characteristic patterns of the signals recorded. This analysis helps to classify sleep stages and identify any abnormalities present, making PSG a valuable diagnostic tool in the assessment and treatment of sleep-related disorders.

1.2.1 Electroencephalogram

During a PSG study, electrodes are placed on the scalp to record electrical activity in the brain, which is known as an Electroencephalogram (EEG). The EEG signals are

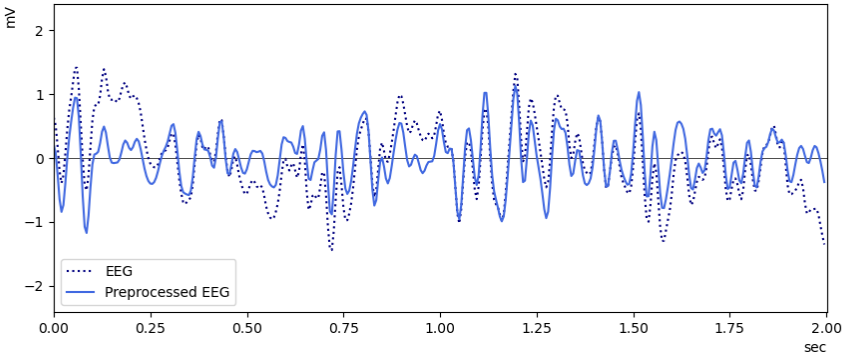


Figure 1.2: Electroencephalogram signal sampled at 200 Hz.

then analyzed to identify different sleep stages and any abnormalities.

To analyze EEG signals, they are decomposed into different frequency bands, with delta, theta, alpha, beta, and gamma being the most commonly used. The delta band represents frequencies below 3 Hz, while the theta, alpha, beta, and gamma bands correspond to frequencies between 4-8 Hz, 8-13 Hz, 13-30 Hz, and above 30 Hz, respectively. Delta and theta waves are associated with deep sleep stages, while alpha and beta waves are associated with lighter sleep stages and wakefulness. Gamma waves, on the other hand, are associated with cognitive processing and attention [11].

To filter out unwanted noise and focus on the information carried by EEG signals, bandpass filters are employed. Filters with passbands ranging from 4 Hz to 40 Hz are commonly used in EEG analysis. These filters play a crucial role in improving the accuracy and reliability of EEG-based analyses by enhancing the signal-to-noise ratio of EEG signals [12, 13].

1.2.2 Electrooculogram

The Electrooculogram (EOG) measures eye movement, and the differences in EOG signals between non-REM and REM sleep, as well as eye blinking in wakefulness, can help distinguish between different sleep stages. Understanding these EOG signal characteristics is crucial for developing accurate and reliable methods for sleep stage

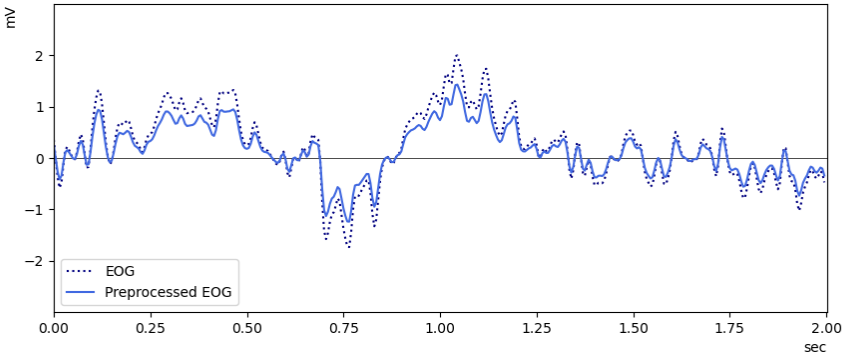


Figure 1.3: Electrooculogram signal sampled at 200 Hz.

classification [7].

One advantage of EOG signals is that they can be easily acquired with portable and wearable devices. To obtain EOG signals, a sensor is attached to the upper right or lower left corner of the eye, and another sensor is attached to the earlobe. This setup allows for continuous monitoring of eye movement during sleep, providing valuable information about an individual's sleep patterns and overall sleep quality [14–16].

The use of EOG signals in sleep stage classification has become increasingly important in recent years due to its ease of use, non-invasive nature, and ability to provide valuable information about an individual's sleep patterns. With the ongoing development of wearable technology, EOG signals are likely to continue playing a significant role in sleep research and clinical practice.

1.2.3 Electromyogram

In PSG, an Electromyogram (EMG) sensor is usually attached to the chin to measure muscle activity in this area. While the amplitude of EMG signals is too variable to directly classify sleep stages, analyzing relative amplitude differences between epochs and irregular high-frequency components can provide clues to sleep stages, as well as to body movements, snoring, or wakefulness [13].

It is important to note that EMG signals can be affected by noise from the skin

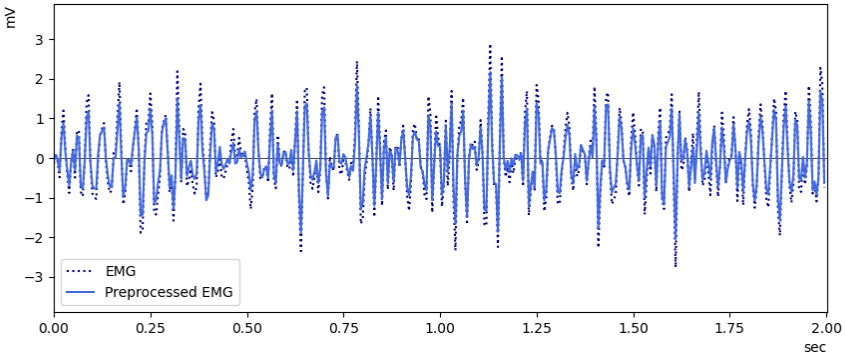


Figure 1.4: Electromyogram signal sampled at 200 Hz.

layers, which occurs at frequencies above 500 Hz and below 20 Hz. Therefore, to extract meaningful information from EMG signals, it is necessary to apply appropriate filtering techniques, such as bandpass filters with a passband of 20 Hz to 500 Hz [17, 18]. This filtering approach is effective in removing noise and focusing on the relevant information contained within the signal.

1.2.4 Electrocardiogram

Electrocardiogram (ECG) is a valuable tool for monitoring the heart's rhythm and electrical activity. ECG is considered a primary vital signal and is commonly used to assess the condition of the heart. In sleep studies, ECG is used to monitor basic heart rate or to detect arrhythmia, an abnormality in physiological rhythm. While ECG can provide valuable information about the heart's condition during sleep, it has not been considered necessary for determining sleep stages [7].

Sleep stage classification typically relies on other signals, mainly EEG combined with EMG, and EOG, which provide direct information about the state of the brain, muscles, and eye movements during sleep. However, the use of ECG in sleep stage classification cannot be entirely ruled out. In certain cases, such as when assessing sleep-related cardiac events, ECG may provide valuable information about the physiological state of the individual during sleep [4].

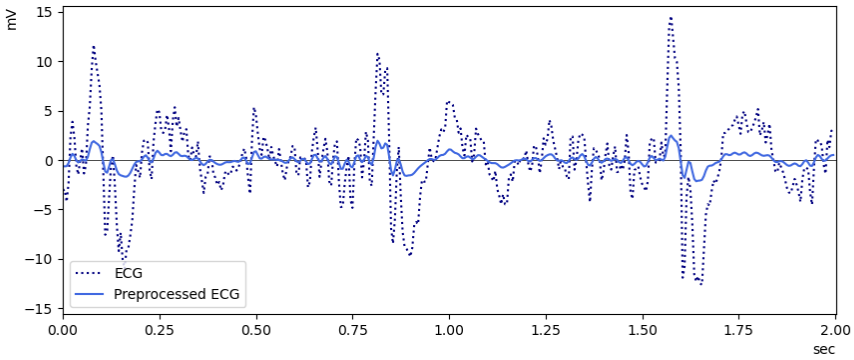


Figure 1.5: Electrocardiogram signal sampled at 200 Hz.

One attempting feature of ECG signal is its ease of acquisition using small and portable devices, making it a convenient and accessible tool for monitoring heart rate and detecting arrhythmias in various settings, including during sleep. Therefore, while ECG is not typically used for determining sleep stages, it remains an important signal in sleep studies as it can provide valuable information about the heart's condition during sleep. The use of ECG in sleep stage classification may be limited, but it cannot be entirely ruled out, particularly in cases where assessing cardiac events during sleep is important.

1.2.5 Heart Rate Variability from ECG

Heart rate variability (HRV) is a significant feature that can be extracted from ECG signals. HRV reflects the variations in time intervals between consecutive heartbeats, which is an important indicator of the heart's autonomic nervous system activity [19].

The HRV signal can be analyzed in the time and frequency domains. The time-domain features are typically derived from the statistical analysis of the inter-beat intervals and include measures such as the standard deviation of the normal beats and the root-mean-square of successive differences between the normal beats. The frequency-domain features are derived from the power spectral density of the HRV signal, which Table 1.1a summarizes time-domian HRV features, and Table 1.1b for

Table 1.1: The representative heart rate variability.

(a) Time-domain features

HRV (unit)	Description
SDNN (ms)	Standard deviation of the time intervals between successive normal heart beats (NN).
SDSD (ms)	Standard deviation of differences between adjacent R wave peak interval.
RMSSD (ms)	Root mean square of the difference between adjacent R wave peak interval.
NN50	Number of interval differences of NN greater than 50 ms.
pNN50 (%)	Percentage of NN50 out of total NN.
HR statistics (bpm)	Mean, maximum, minimum, and standard deviation of heart rate.

(b) Frequency-domain features

HRV (unit)	Description
VLF (ms^2)	Absolute power of the very-low-frequency band (0.003 Hz $\sim$ 0.04 Hz). This band reflects an intrinsic rhythm produced by the heart.
LF (ms^2)	Absolute power of the low-frequency band (0.04 Hz $\sim$ 0.15 Hz). LF is affected by breathing from 3 to 9 bpm.
HF (ms^2)	Absolute power of the high-frequency band (0.15 Hz $\sim$ 0.40 Hz). Reflect fast changes in beats due to parasympathetic activity. Also called as the respiratory band and influenced by breathing from the 9 to 24 bpm.
LF / HF	Low frequency to high frequency ratio. Estimation of the balance between sympathetic and parasympathetic nervous system.
LFNU, HFNU	Normalized low- or high-frequency power by the summed power of the low- and high-frequency.

frequency-domain features.

HRV has been used as a feature in sleep stage classification, and some studies have shown promising results. However, selecting representative and discriminatory HRV features for each sleep stage remains a challenge [4,20,21]. Some studies have focused on feature selection problems using machine learning algorithms such as support vector machines and decision trees. However, it is not clear whether these solutions are suitable for deep neural networks trained in an end-to-end manner.

In addition to its potential use in sleep stage classification, HRV has also been studied in the context of heart-brain interactions, as it reflects the regulation of the autonomic nervous system and is associated with a variety of physiological processes such as blood pressure and gut function. Moreover, the use of small and portable devices has made it easier to obtain HRV measurements in various settings, including during sleep. Therefore, further research is needed to investigate the potential applications of HRV in sleep studies and beyond.

1.3 Deep Neural Network Architectures

Deep neural networks (DNNs) have recently gained attention due to their versatility and flexible application to a wide range of tasks. In this study, we utilized fully connected, convolutional, recurrent, and Transformer neural network layers for automatic sleep stage classification.

DNN layers vary in type, depending on their specialization. The most basic and straightforward layer is the multilayer perceptron, also known as a fully connected neural network (FCDNN). This layer connects different dimensional layers or produces desired outputs from network inputs.

Convolutional neural networks (CNNs) are widely used for producing outcomes by applying convolution operations to inputs. CNNs are particularly effective when dealing with inputs of large size, such as images or very long sequences, as they can

capture the single or multidimensional locality of the inputs. This makes CNNs highly valuable for generating features from the inputs.

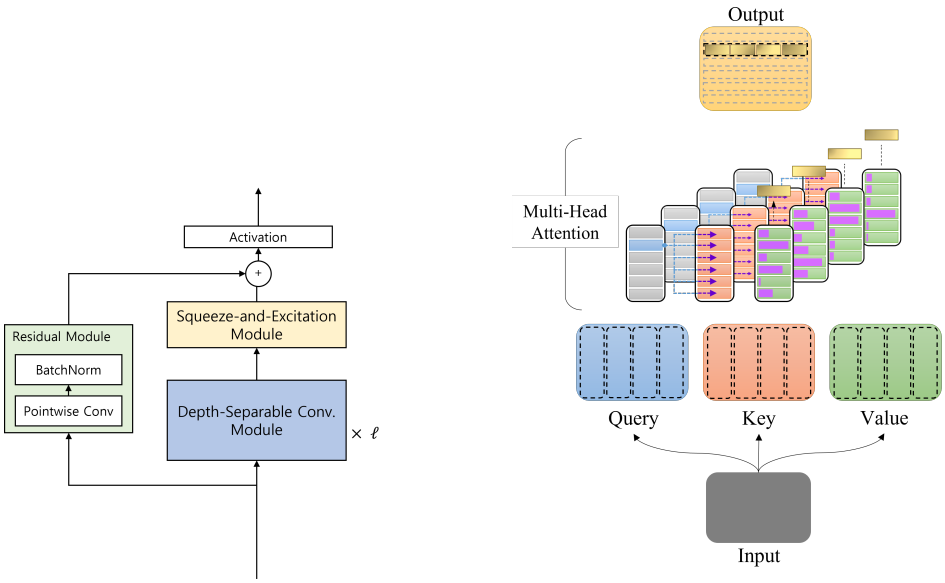
Recurrent neural networks (RNNs) are designed to focus on sequences, order, or relationships of a series of inputs, which is essential for sleep stage classification.

1.3.1 ContextNet

Figure 1.6a depicts two key component of the ContextNet: depth-separable convolutions and squeeze-and-excitation blocks. These constituents are engineered to conserve domain-specific information while augmenting dimensionality [22]. The depth-separable convolution compartmentalizes convolution operations, thereby minimizing the interdependence of dimensional information. For spectrogram inputs, it treats frame-related features and frequency dimension information separately by executing a 1-dimensional convolution along each respective axis. The squeeze-and-excitation block applies weight to the channel, depth, or frequency dimension. It compresses (squeezes) and then re-enhances (excites) the channel dimension information by processing it through a quarter-sized and then an original-sized FCDNN. The output from the excitation layer is then multiplied along the input's channel axis. As such, each channel is either intensified or attenuated by the squeeze-and-excitation block, effectively managing the prominence of various features in the input.

1.3.2 Transformer

The Transformer layer or block is introduced in [23] and designed for the context of machine translation using an encoder-decoder system. However, its powerful ability to refer to related information and generate refined outputs has led to its application in various other sequential tasks [24–26]. Transformer networks are particularly effective for processing inputs of very long length, such as lengthy texts or speech recognition tasks. As shown in Figure 1.6b, the primary mechanism of the Transformer block is self-attention. The self-attention block allows each input to identify itself, its position



(a) The basic block of ContextNet

(b) The self-attention block of Transformer

Figure 1.6: The base block of (a) ContextNet and (b) Transformer.

in the sequence, and its relationship to other parts of the sequence. The self-attention operation consists of three steps. At first, the query and key-value pairs are produced from each input. The similarity between the query and keys, attention vectors, is calculated with the softmax function, and produce outputs based on the attention vectors and values corresponding to the keys. With self-attention, Transformer layers are replacing RNNs in many sequential tasks and extending their applications.

1.4 Scope of Dissertation

This paper covers sleep stage classification and sleep disorder detection from Chapter 2 to Chapter 4. Chapter 2 investigates the potential and importance of different signals used in PSG testing for automatic sleep stage classification. We use identical DNN architectures and experimental methods to test the signals, with a focus on identifying signals that can be easily acquired in a home setting for preliminary sleep testing.

Chapter 3 examines different approaches to sleep stage modeling and their benefits

and limitations. We develop sleep models that focus on the sequence of sleep stages to compensate for the lack of information that inevitably arises when using limited signal sources. These sleep models function similarly to language models in ASR systems, and we improve the automatic sleep stage classification system by decoding the classified sleep stages with the sleep model.

In Chapter 4, we develop a DNN architecture for sleep stage classification using ECG signals. While ECG signals alone do not provide sufficient information about sleep stages, we have developed an end-to-end model for ECG-based sleep stage classification, inspired by the approach used in ASR, specifically by leveraging the sequential nature of the data.

In Chapter 5, we develop an audio-based apnea classification model that can classify four types of apnea events from normal sleep and estimate the apnea-hypopnea index from the model's prediction results. We can classify physically interrupted apnea events from normal sleep using sound and improve the accuracy of apnea-hypopnea index estimation, although distinguishing brain-related apnea events from other events relying on sound is challenging.

Chapter 2

PERFORMANCE ASSESSMENT OF PARTIAL PSG SENSORS AND NEURAL NETWEORKS

2.1 Introduction

Sleep-related disorders are becoming increasingly prevalent, leading to a growing demand for sleep tests in clinical settings. Polysomnography (PSG) is the most commonly used sleep test and involves the use of sensors such as electroencephalogram (EEG), electrooculogram (EOG), electromyogram (EMG), electrocardiogram (ECG), and respiratory sensors to record various physiological signals during sleep. These signals exhibit characteristic waveforms that correspond to different stages of sleep and can be used for manual or automated sleep stage classification.

However, PSG-based sleep tests have their limitations. They are expensive and often uncomfortable, making it difficult for subjects to obtain proper sleep and hindering accurate data acquisition. Among the sensors used, EEG sensors are critical for conventional sleep stage classification but are very cumbersome to use.

Developing accurate sleep stage classification methods that do not rely on complex sensors is a challenging task, but this could provide significant advantages in clinical and academic settings. In this chapter, we aim to establish a more convenient sleep

stage classification test by assessing the degree of performance loss when EEG sensors are removed.

Recent studies have used deep neural networks (DNNs) for sleep stage classification, with some employing convolutional neural networks (CNNs) for feature generation and others using simple linear classifiers or recurrent neural networks (RNNs) for final classification [1, 3, 4, 27–31].

The authors of [1] showed the sleep stage classification results using Haagleanden Medisch Centrum Sleep Database (HMC) [32]. We also used this public dataset for this study. The performance of inter- and intra-database models was also analyzed using additional datasets. The research in [27] used the Montreal Archive of Sleep Studies dataset and showed sleep stage classification using a hierarchical RNN model [33].

Recent studies focus on the use of wearable devices for assessing the quality of sleep [34, 35]. However, its primary goal is to provide simple and summarized classification results, such as the sleep duration, instead of epoch-by-epoch precise results. The recently developed Transformer model displays a better performance in sequence learning, capable of taking full advantage of long spans of input data [23]. However, the application of Transformer models in sleep stage classification needs to be studied.

We use deep neural network technology and compare the performance of sleep classification with whole or ablated PSG data. A CNN is used for extracting 64-dimensional features from PSG data [1]. Then, for sleep stage classification with these features, we compare four DNN algorithms, which are simple linear classifier, fully connect DNN (FC-DNN), RNN, and Transformer. This study measures the performance of these different DNN models when only selected signals in PSG data are used. Since the removal of some sensor data inevitably introduces an accuracy drop, we use a more powerful sequence model, Transformer. Whereas conventional models only utilize short, fragmented local information that lasts from 30 seconds to an hour at maximum, the Transformer model can analyze extensive data that comprise the entirety of one night's sleep.

We measured the performance of each model when EEG, EOG, and EMG signals were used, when EEG was omitted, and when ECG was added. We analyze the results to know the effects of removing some of these sensor signals.

The rest of this chapter is organized as follows. Section 2.2 describes the sleep stage classification method using deep neural network models. Section 2.3 presents the performance test results of these models on various combinations of input sensors. Finally, Section 2.4 draws the conclusion of this study.

2.2 Sleep Stage Classification with Deep Neural Network Modles

2.2.1 PSG Data and Feature Preparation

We used the public PSG data from the Haagleanden Medisch Centrum Sleep Database (HMC) [1], which includes a total of 154 sleep records. Among them, three records (SN014, SN064, SN135) that contain only DC signals were excluded. This PSG contains EEG, EMG, EOG, and ECG signals that are sampled at 256 Hz. Two EEG channels, one EMG, and one EOG were used in the experiments. According to the annotations, only the time between light-off and light-on was used, and the duration of each PSG data is about seven to eight hours.

We do not have personal information for each record. However, the aggregated subject information is available and listed in Table 2.1.

The data was resampled at a frequency of 100 Hz after applying pre-filtering according to the method in [1]. For eight hours of data, each channel contains a total of 8 hours $\times$3,600 seconds/hour $\times$100 samples/second. This time-series data is divided into 30-second intervals, which is called an epoch, and classified into one of the five sleep stages (Wake, REM, N1, N2, and N3).

Each epoch's label is obtained from the manual classification by medical specialists, which is considered the gold standard. The HMC dataset was classified according

Table 2.1: The aggregated subject information of HMC dataset, recordings of 88 male and 66 female participants.

	Age	TIB	TST	SE	AHI_{TST}	ArI_{TST}
Mean	53.8	7.5	6.2	82.7	14.6	20.1
Std.	15.4	1.2	1.5	14.4	17.0	15.2

TIB, time in bed, time from 'light on' to 'light off' in PSG annotation.

TST, total sleep time, time without wake stages.

SE, sleep efficiency, percentage of TST to TIB.

AHI, apnea-hypopnea index, the number of apnea or hypopnea events per an hour.

ArI, arousal index, the arousal per an hour.

to version 2.4 of the American Academy of Sleep Medicine guidelines [7].

2.2.2 Deep Neural Network Models

We use CNN for feature generation from one epoch of input sensor data. The CNN for feature generation consists of three layers, and each layer contains a convolution layer, a batch normalization layer, an activation layer, and a pooling layer in order [36]. The three CNN layers contain 8, 16, and 32 output channels, and the size of the kernel is 101 in the time dimension. The activation layer uses rectified linear units [37]. The pooling layer employs the average pooling of size 1×2 with a stride of 2 to maintain the size of the channel but halve the length. By applying the CNN layers, each epoch that corresponds to 30 seconds of PSG data is transformed into a 32×325-dimensional vector. Then, a linear layer is used to form a 64-dimensional feature vector.

The first method to classify the sleep stage is to apply a linear classifier to the 64-dimensional feature obtained above. In this case, the classification is conducted using only one epoch of sensor data, without consulting past or future data. This is why we need several sensor data, including EEG, EOG, and EMG, for sleep classification. Note that the EEG signal shows particular waveforms corresponding to each sleep stage.

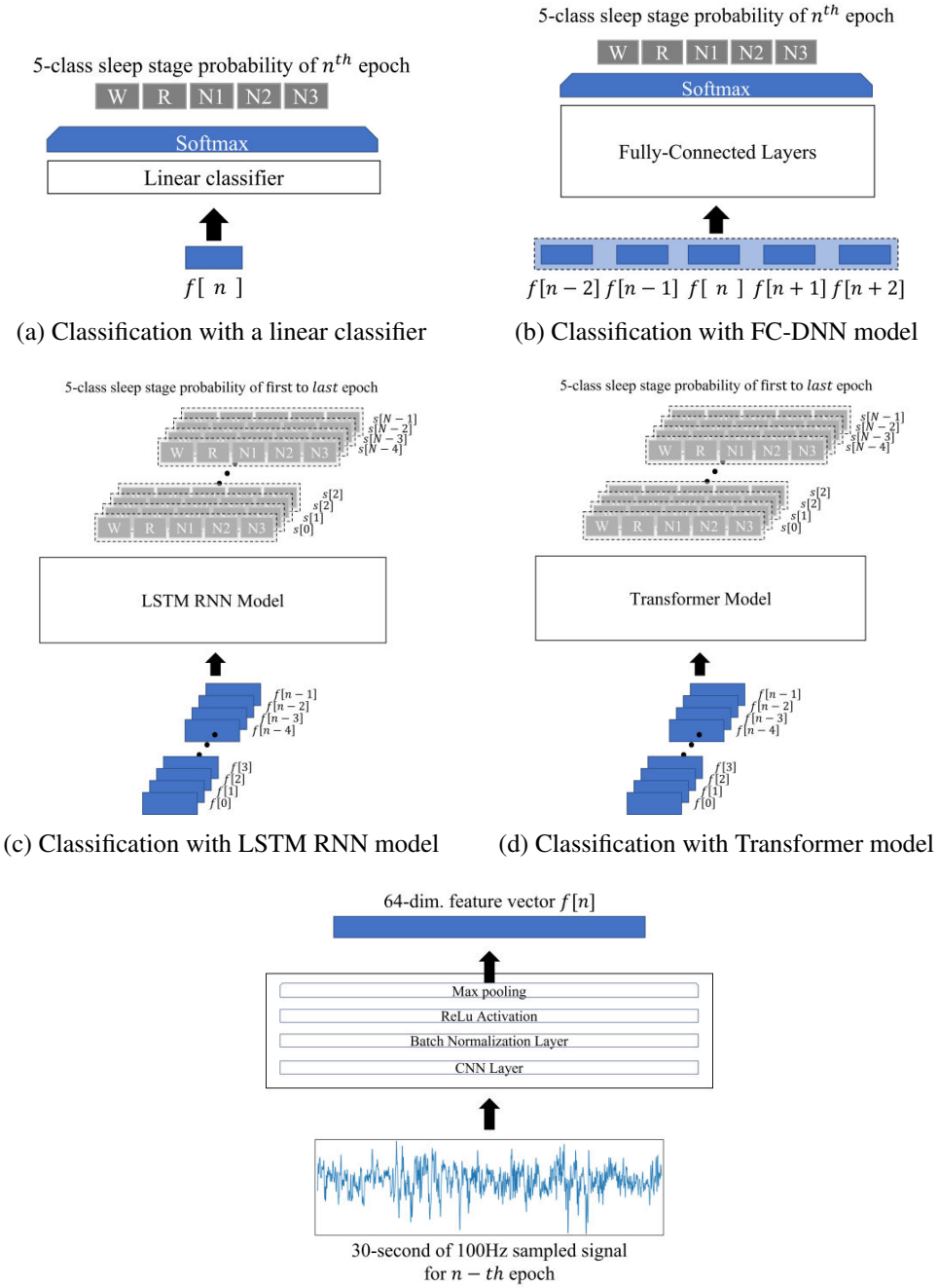
The second method uses FC-DNN with a layer size of 320 and a depth of 2. The input signal is first converted to 64-dimensional features using CNN, and then the FC-

DNN processes five epochs of input data, f_{n-2}, f_{n-1}, f_n, f_{n+1}, and f_{n+2}, to make one classification, s_n. Since this model consults nearby neighboring inputs as well as employs deeper models for classification, improved classification over the linear classifier is expected.

The third method uses an RNN model, where the obtained CNN features are applied to the long short-term memory (LSTM) RNN model [38]. We use the same LSTM RNN that was employed in [1], where two layers of bidirectional LSTM models with a dimension of 64 in each direction were used. Although the RNN model receives one feature at a time, it has the ability to memorize the past and perform sequence learning using them. A bidirectional RNN has the ability to utilize both the past and future contexts. RNN models have been widely used for many sequence learning applications such as speech recognition and language understanding. However, the length of the past that can be well remembered is very limited to tens of time steps. That is, even if a 30-second time step is used, the sleep phase is predicted using only local information whose length is less than one hour.

We also use the Transformer model for sleep classification as a fourth method [23]. The model consults the entire input sequence for classification. Due to this characteristic, the model has been recently used a lot for sequence understanding [26, 39–41]. Since eight hours of sleep data consists of 960 epochs, the input length to the Transformer is 960. The Transformer model consist of 2 layers, and each layer is represented with a 64-dimensional vector and the number of heads is two. The feature vector for every time step is 64 as described above.

In this experiment, if eight hours of data are divided into 30-second units, a total of 960-time steps is obtained. These 960 feature vectors are used as the input of the LSTM and Transformer model, and are trained to predict the sleep stages (Wake, REM, N1, N2, and N3) obtained from PSG data. Figure 2.1 shows the feature extraction using CNN and compares four classification approaches, the linear classifier (Linear), FC-DNN, RNN, and Transformer models.



(a) Classification with a linear classifier

(b) Classification with FC-DNN model

(c) Classification with LSTM RNN model

(d) Classification with Transformer model

(e) Feature generation with CNN model

Figure 2.1: Sleep stage classification with deep neural networks. (2.1e) Feature generation, (2.1a) linear classifier-, (2.1b) FC-DNN-, (2.1c) LSTM RNN-, and (2.1d) Transformer-based.

2.2.3 Performance Measurements

We conducted experiments to compare the performance of the Transformer model with the existing methods using a linear classifier, FC-DNN and RNN. Cohen's Kappa (κ) is used as a metric, which is defined as follows:

$$\kappa = \frac{P_a - P_c}{1 - P_c} \tag{2.1}$$

$$P_a = \frac{1}{N} \sum \mathbb{1} \tag{2.2}$$

$$P_c = \frac{1}{N} \sum_c n_c \times t_c, \tag{2.3}$$

where P_a and P_c accuracy and probability of random agreement, respectively. Here, c represents the class index, N is the total number of epochs, $\mathbb{1}$ is 1 if the model prediction is correct else 0, n_c is the number of epochs that model predicts as class c, and t_c denotes the number of epochs belonging to class c. Note that Cohen's Kappa eliminates the bonus of randomly predicting the output label.

We also measured the accuracy, precision, recall, and F1 score for each classified sleep stage. These measures are defined as follows, where *TP, TN, FP,* and *FN* mean true positive, true negative, false positive, and false negative, respectively.

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN} \tag{2.4}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.6}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{2.7}$$

Although the results on linear classifier in Table 2.2 reproduce those of [1], our results are slightly lower. Note that in [1], when splitting the dataset, all data were

divided into 30-second time steps, then randomly mixed and divided into training, validation, and test data. As results, one participant's recordings can be mixed into training, validation, and test sets. To prevent data mixing strictly, we divide 98 records out of 151 available records into training, 24 records for validation, and 29 records for a test.

The accuracy of test data was measured with the model parameters saved when training showed the highest κ in validation.

2.3 Experimental Results

We employed four kinds of DNN models for performance evaluation, which are the linear classifier, FC-DNN, RNN, and Transformer. The input features are generated using CNN. Eliminating some of the sensor signals for sleep measurement helps to relieve the burden of sleep tests. We consider EEG measurements the most burdensome among EEG, EMG, EOG, and ECG sensors. Thus, we try to measure the performance when EEG sensor signals are deleted. The combination of input signals used for this ablation study are (a) EEG, EOG, and EMG, (b) EOG and EMG, (c) EOG, EMG, and ECG, (d) EMG and ECG, (e) EOG and ECG, (f) ECG, (g) EMG, and (h) EOG. In experiment (a), we first assess the performance using EEG, EOG, and EMG as the input. In experiments (b) and (c), the performances when EEG is missing are assessed using the linear classifier (Linear), FC-DNN, RNN, and Transformer models. The experiments for (d), (e), (f), (g), and (h) are performed using FC-DNN and Transformer models.

2.3.1 EEG, EOG, and EMG as the input

This experiment utilizes all the data in ordinary PSG data. Linear classifier, FC-DNN, RNN, and Transformer models are used for the classification. As we can find in Table 2.2, the test accuracy improves in the order of Linear, FC-DNN, RNN, and Trans-

Table 2.2: Cohen's Kappa and accuracy according to classification models. EEG, EMG, and EOG are used as input.

Model	Context length	Validation set		Test set	
		κ	Acc.	κ	Acc.
Linear	1 epoch	0.72	0.79	0.67	0.75
FC-DNN	5 epochs	0.74	0.80	0.68	0.76
RNN	Less than 1 hr	0.74	0.80	0.68	0.76
Transformer	Overnight	0.72	0.78	0.70	0.77

former. However, the performance improvements when we employ Transformer is not substantial. We also show the normalized confusion matrix in Table 2.3, when using Transformer model. As the normalized confusion matrix in Table 2.3 shows, the recall for the N1 sleep stage is quite low, about 34%, but those for other stages are fairly high. About 37% of N1 is misclassified into N2, and 17% is confused into Wake. Some sleep stage classification methods do not discern N1 and N2 stages because these two stages are quite confusing [3, 5, 35].

2.3.2 EOG and EMG (no EEG) as the input

This experiment is conducted to know the performance effects when the EEG signal is missing. Table 2.4 shows the results with only EMG and EOG data. We can find that the κ value drops by about 0.1 for Linear, RNN, and Transformer models.

Table 2.3: Normalized confusion matrix of the Transformer model with EEG, EMG, and EOG input on the test set.

		Model prediction					
		W	$N1$	$N2$	$N3$	REM	SUM
True label	W	0.146	0.011	0.002	0.000	0.002	0.161
	$N1$	0.021	0.040	0.044	0.000	0.013	0.118
	$N2$	0.006	0.010	0.300	0.042	0.014	0.374
	$N3$	0.008	0.000	0.003	0.175	0.000	0.187
	REM	0.002	0.003	0.019	0.001	0.135	0.160
	SUM	0.184	0.065	0.370	0.218	0.164	1

Table 2.4: Cohen's kappa and accuracy of the four models using EOG and EMG inputs, with the addition of an ECG signal.

Input signals	Model	Validation set		Test set	
		κ	Acc.	κ	Acc.
EOG, EMG	Linear	0.64	0.73	0.60	0.70
	FC-DNN	0.68	0.76	0.61	0.71
	RNN	0.64	0.73	0.58	0.68
	Transformer	0.60	0.70	0.58	0.69
EOG, EMG, ECG	Linear	0.60	0.70	0.57	0.68
	FC-DNN	0.63	0.72	0.60	0.70
	RNN	0.63	0.72	0.59	0.68
	Transformer	0.60	0.70	0.61	0.71

2.3.3 EMG, EOG, and ECG as the input

The cardiac signal changes at night with specific patterns. We add ECG to EMG and EOG to utilize this characteristic. The validation and test accuracies are shown for Linear, FC-DNN, RNN, and Transformer models in Table 2.4. Table 2.4 shows that adding ECG signal helps to improve the performances of RNN and Transformer models, although slightly.

2.3.4 Other ablation of input signal

This subsection describes the performance evaluation of sleep stage classification using different input combinations, including EMG and ECG, EOG and ECG, only ECG, hand-picked heart rate variability (HRV) features, only EMG, and only EOG. HRV features are generated by the HRV analysis library[1] developed by Aura Healthcare with frequency domain, geometric, Poincare plots, and cardiac sympathetic and vagus index features for windows 30, 120, and 270 seconds long [42–45]. We used a total of 83 features for the HRV, including the energy difference and the overall mean of the RR or NN intervals.

[1] https://github.com/Aura-healthcare/hrv-analysis

The FC-DNN and Transformer models are used for the evaluation since they outperform Linear and RNN models. The performance results are summarized in Table 2.5. The FC-DNN model does not converge in training when only ECG is used. Thus, the result is missing in the Table 2.5. Here we can find that the EOG signal is the most critical when EEG is not available. With EOG only, we can obtain the Kappa value of 0.58 using the Transformer model. With EOG only or EOG plus ECG, the Transformer outperforms the FC-DNN model by a fairly large margin, although both models show comparable performance when all the input signals are used.

Figure 2.2 shows a comparison of the performance (Cohen's Kappa) of FC-DNN and Transformer for different input combinations. The figure shows that the Transformer model performs better when the number of sensor signals is reduced.

Regarding only ECG-based sleep classification, Table 2.5 compares the performance of CNN-based and hand-picked HRV features. The HRV features show better performance than the CNN-based ones, but the overall accuracy still needs improvement.

Table 2.5: Cohen's Kappa and accuracy of each model according to various inputs. ECG is the CNN-based feature and HRV is the hand-picked feature derived from ECG.

Model	Input signals	Validation set		Test set	
		κ	Acc.	κ	Acc.
FC-DNN	EMG, ECG	0.19	0.40	0.14	0.37
	EOG, ECG	0.49	0.62	0.45	0.59
	ECG	-	-	-	-
	HRV	0.15	0.41	0.16	0.41
	EMG	0.20	0.43	0.21	0.43
	EOG	0.50	0.63	0.46	0.60
Transformer	EMG, ECG	0.28	0.49	0.27	0.46
	EOG, ECG	0.55	0.66	0.53	0.65
	ECG	0.14	0.52	0.11	0.39
	HRV	0.21	0.45	0.21	0.43
	EMG	0.29	0.48	0.27	0.47
	EOG	0.64	0.73	0.58	0.69

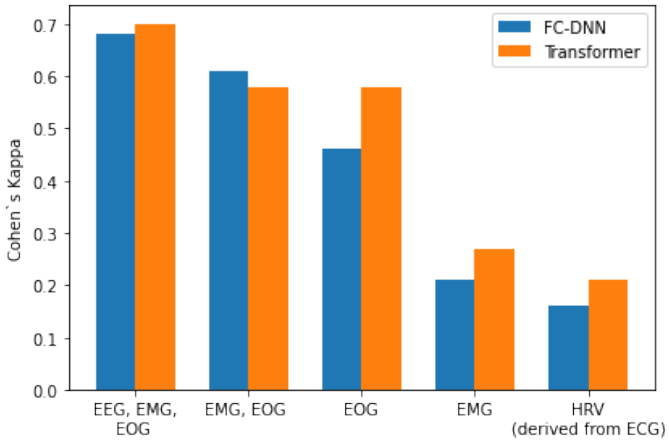


Figure 2.2: Cohen‘s Kappa according to ablation of input signals.

2.4 Conclusion

We studied deep neural network-based sleep stage classification and assessed the performance of simplified measurements that employed only EMG, EOG, electrocardiogram (ECG), or combinations of them. We find that conventional classifiers, such as the simple linear classifier or fully-connected deep neural network, show good performance when the whole set of polysomnography data is used for sleep stage classification, but the Transformer model that analyzes the entirety of one night's sleep displays a superior performance when the EEG signal is ablated. When only ECG is used, the manually selected heart rate variability feature showed better performance than the convolutional neural network-based one.

Chapter 3

SLEEP MODEL: A SEQUENCE MODEL FOR PREDICTING THE NEXT SLEEP STAGE

3.1 Introduction

The current prevalence of sleep disorders has resulted in many people seeking help from psychiatrists or specialists. The sleep test breaks down a sleep period into epochs that are typically 30 seconds long and assigns a sleep class label to each epoch. While PSG remains the gold standard for clinical evaluation of sleep, it is impractical for long-term monitoring of sleep problems at home. In recent years, several surrogate measures have been studied to reduce the costs and discomfort associated with PSG testing.

Many sleep classification studies have employed a subset of PSG sensors, including single-channel EEG, EMG, EOG, ECG, or PPG [3, 4, 31, 35, 46, 47]. However, these surrogate tests inevitably result in a drop in accuracy when compared to full PSG. Deep neural networks (DNN) provide new opportunities to significantly improve the accuracy of simple sleep tests [27, 48, 49].

The prediction of the next sleep stage can be challenging, but sleep generally follows typical patterns [50]. It is unlikely that there will be a change in sleep stage at

every epoch. When observing sleep patterns, short-term inertia or predictability is evident. Additionally, the sleep pattern typically includes four to six long-term cycles overnight, each of which comprises both REM and non-REM stages [51, 52].

Our study proposes a sleep model that takes into account the sequential nature of sleep and leverages it to improve classification accuracy. By relying on sleep models, it becomes possible to enhance accuracy, especially when using simple surrogate sensors. The concept of the sleep model is derived from the language model, which predicts the likelihood of the next word or character in speech or text, based on previous sequences [53]. Language models are particularly important for improving the accuracy of speech recognition by providing language-based information that can correct recognition errors or grammatically infeasible sentences caused by noise or other disturbances in speech signals [54, 55]. In a similar vein, sleep models aim to learn and predict the probability of the next sleep stage based on previous observations. Just like language models, sleep models can provide valuable information that can correct errors in classification that might be caused by noise or other disturbances in the sleep signal.

Section 3.2 provides an overview of the related works and background, including the sleep data and signal processing models used in the study. We also introduce language models in this section. In Section 3.3, we present our proposed sleep models and beam search decoding. Experimental results and analyses are provided in Section 3.4. Finally, Section 3.5 concludes the study.

3.2 Sleep Data and Background Information

3.2.1 Sleep Dataset

There are many sleep datasets publicly available due to a considerable number of PSG tests. These sleep datasets serve two purposes: firstly, to extract sleep sequences from simplified input signals using signal processing and deep neural networks; secondly,

to develop sleep models based on sleep patterns observed in PSG tests. To this end, we use the Haagleanden Medisch Centrum Sleep Database (HMC) dataset for the first purpose, and both HMC and NCH Sleep DataBank (NCHSDB) datasets for the second purpose [32, 56].

The HMC dataset was partitioned into 98 training, 24 validation, and 29 test sets, while the NCHSDB dataset used 3036, 379, and 380 records for training, validation, and testing, respectively. Prior to processing, the signal data underwent pre-filtering and resampling at a rate of 100 Hz, following the method outlined in [32]. Each epoch of the time-series data was then assigned to one of five sleep stages (W, REM, N1, N2, and N3), based on the annotations provided for a sleep recording typically lasting eight hours.

3.2.2 Signal Model

Our study involved the development of sleep signal-processing models capable of classifying sleep stages using either the entire PSG data or a subset of it. The aim was to evaluate the performance of these models in both beam search decoding and greedy decoding. In conventional automatic sleep-stage classification methods, the sleep class with the highest probability at each epoch is selected, a process referred to as greedy decoding in this context.

To build our sleep signal model, we followed the DNN model proposed in [47], which consisted of feature extraction and classification blocks as depicted in Figure 3.1.

The input sensor data which included two EEG channels (C4-M1 and C3-M2), one EMG (chin), and one EOG (E2-E1) channel were applied to the feature extraction block. The feature extraction block consisted of 3 convolutional neural networks (CNN) and 1 fully-connected DNN (FC-DNN) [57, 58]. In this block, the eth epoch that corresponded to 30-second of PSG data, d_e, was transformed to a 64-dimensional feature vector, f_e. The classification block employed in this study was formed using

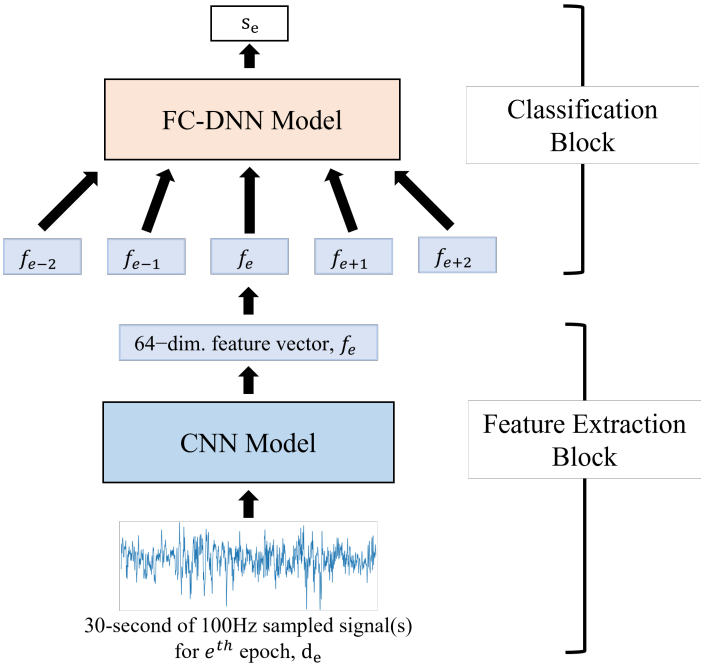


Figure 3.1: Signal model employing deep neural networks.

FC-DNN with a layer size of 320 and a depth of 2. To classify one sleep stage, s_e, the classification block processed five epochs of input feature vectors: f_{e-2}, f_{e-1}, f_e, f_{e+1}, and f_{e+2}.

The HMC sleep dataset was used to develop two sleep signal models. The first model utilized four channels, which included EEG, EOG, and EMG as inputs. The second model only employed EOG and had a single channel.

3.2.3 Language Model

A language model predicts the probability distribution of the next word in a corpus. The inspiration for the proposed sleep model came from language models that are frequently used in computational linguistics and probabilistic fields [59,60]. An excellent example of such a large language model is ChatGPT, which is capable of generating human-like responses [61].

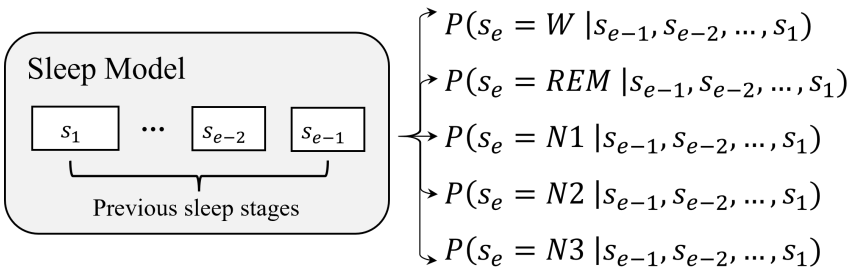


Figure 3.2: The sleep model that predicts the probabilities of sleep classes in the next epoch based on the previous ones.

They are traditionally trained on large corpus of text data, such as Wikipedia or speech transcriptions [62,63]. Traditionally, n-gram based language models have been used, but in recent years, DNNs such as LSTM-RNNs or Transformers have gained popularity due to their superior performance [23, 26, 33, 38, 53]. However, n-gram-based models are easier to build and require less time for inference [64–66].

3.3 Sleep Model and Beam Search Decoding

3.3.1 Sleep Model

The **SL**eep **M**odel (SLM) proposed in this study predicts the likelihood of each sleep class for the next epoch. These classes correspond to one of the five sleep stages, as shown in Figure 3.2. The SLM was constructed using training data that contains sleep sequences. Sleep stages exhibit fairly consistent patterns, and the following sleep class is heavily influenced by the preceding stages, much like language models [50, 51].

We built n-gram SLMs in manner of an n-gram language model. An n-gram SLM was built in this study by approximating the probability with the number of occurrences of certain patterns in the training dataset. The number of occurrences was counted while traversing the entire sleep-stage sequences in dataset.

For example, a 3-gram $(trigram)$ SLM shows the probability of the next sleep class based on the previous two sleep classes, which can have 25 $(= 5^2)$ cases. The

tri-gram probability of $P\left(\frac{\text{N3}}{\text{N1,N2}}\right)$, which is the probability of next stage being N3 following previous N1 and N2 stages, can be approximated as follows:

$$P\left(\frac{\text{N3}}{\text{N1},\text{N2}}\right) \approx \frac{C\left(\text{N1},\text{N2},\text{N3}\right)}{C\left(\text{N1},\text{N2}\right)}, \tag{3.1}$$

where $C\left(\text{N1},\text{N2}\right)$ denotes the number of occurrences or counts of sleep stages N1 followed by N2 in the data. Thus, the process of building an n-gram model is straightforward and the accuracy of prediction generally improves with the increasing value of n.

However, the above approximation cannot be accurate when the number of counts in the denominator, for example $C\left(\text{N1},\text{N2}\right)$ in Equation (3.1), is not sufficiently large, which is called the data scarcity problem. To avoid the problem of data scarcity, a large training data is required. Increasing n also results in an exponential growth in the number of possible cases ($= 5^{n-1}$). Generally, when the number of counts is very small, the modeling fallbacks to a lower n [67]. In addition, there should not be a zero-probability prediction for the next class. Thus, when the numerator count is zero, we need smoothing that adds small numbers to the numerator and denominator terms to ensure a computational safety.

A length of 8 hours sleep consists of 960 epochs or sleep stages, thus the total number of epochs for the HMC training set was approximately 100,000. To obtain a fairly accurate probability estimation, the number of the denominator count in Equation (3.1) needs to be around 100. If the sleep sequences are equally distributed, which is unrealistically optimistic, the number of different sequences that can be formed with 100,000 epochs is about 1,000, which can be approximated to 625 ($= 5^4$). Thus, we consider that the meaningful n-gram size for HMC dataset would be around $n = 5$.

We also developed SLMs based on the LSTM-RNN architecture by varying the number of LSTM layers or the hidden dimension of each LSTM layer [38]. The LSTM is a type of RNN architecture that has internal states to retain latent information from

previous sequences. This property makes LSTM based SLMs unrestricted by previous sequence length, in contrast to n-gram models, which are limited to a length of $n-1$. The network architecture of the LSTM based SLM consists of a sleep stage embedding layer, LSTM layers, and a softmax output layer.

3.3.2 Beam Search Decoding

Typical automatic sleep-stage classification only employs a sleep signal model. In the signal model, we can simply select the sleep class with the highest probability from each position in the sequence, which is often called greedy decoding. When surrogate sensors are used for sleep tests, the accuracy of the signal model with greedy decoding is not sufficiently high. The recognition accuracy of greedy decoding can be improved using a SLM. The output of the SLM provides the prior information for sleep classification. The posterior probability was determined by multiplying, addition in the $\log$ domain, the probability of the signal model (P_{sig}) by that of the SLM (P_{SLM}) using Equation (3.2).

$$\begin{aligned}\log P\left(s_e\right) = &\log P_{sig}\left(s_e|d_{e-2},\ldots,d_{e+2}\right)\\ &+\alpha\log P_{SLM}\left(s_e|s_1,\ldots,s_{e-1}\right),\end{aligned} \tag{3.2}$$

where d is input signal, s denotes the sleep stages, e in subscript means the eth epoch, and α is a parameter that assigns a balanced weight between the signal and sleep models.

Here, the signal model generates the likelihood of the sleep class using the input signal, whereas the SLM provides the prior probability. Sleep-stage classification can be considered a sequence recognition problem, that is, we need to consider the results over a long time span.

The beam-search-decoding algorithm can select multiple sleep classes for each epoch in a given sequence [68–70]. This means that even sleep classes that do not show the highest probability can be saved for further evaluations in the future. Because

the number of sequence candidates grows exponentially as decoding progresses, 5^e in our case, it is not possible to retain all of them. The algorithm chooses the W-best alternatives via a hyperparameter known as the beam width. The posterior probability for each epoch was obtained by multiplying the probabilities obtained from the signal and sleep models. To determine the most likely beam, the posterior probabilities of each beam sequence are multiplied, and the resulting probability values are compared to select the sequence with the highest probability.

3.3.3 Model Details and Metric

We built the n-gram model using KenLM, which implements fall-back, smoothing, and data compression [71]. We evaluated n-gram models on HMC and NCHSDB dataset, varying n from 2 to 9, and applied the fall-back option. We also developed four kinds of LSTM-RNN based SLMs, 2 or 4 LSTM layers with 256 or 1,024 hidden dimensions on training split of each dataset.

The performance of language models is commonly measured by the perplexity [72]. The perplexity is defined as the inverse probability of the sequence of words, w, normalized by the number of words, N, as shown in Equation 3.3,

$$Perplexity\left(w_1, w_2, \ldots, w_N\right) = \sqrt[N]{\frac{1}{P\left(w_1, w_2, \ldots, w_N\right)}}\,. \tag{3.3}$$

We also used the perplexity to measure the performance of SLMs. A lower perplexity value indicates better performance of the SLM in predicting the next sleep stage.

3.4 Experimental Results

3.4.1 n-gram and LSTM-RNN based Sleep Models

Table 3.1 displays the 2-gram (bi-gram) probabilities on the HMC dataset. These probabilities show that when the current sleep stage is N1, the probability of REM stage at

the next epoch is 8%. As expected, the diagonal terms have high values, indicating the tendency to repeat the same sleep classes. The transition probability from N1 to N2 is quite high and exhibits a low inertia of the N1 stage. As a result, classification between N1 and N2 introduces many errors.

The perplexity of n-gram SLMs, when assessed with the test dataset, is shown in Figure 3.3. For HMC dataset, it shows that the perplexity decreased as n increased until $n = 5$, but then increased thereafter. As described in Section 3.3, the back-off and smoothing algorithms used in KenLM have an impact on performance distortion when n is greater than 5. As the size of the NCHSDB dataset is approximately 20 times larger than the HMC dataset, we can expect improved SLM performance and less sensitivity to n.

We then trained LSTM-RNN based SLMs using the HMC or NCHSDB training set. Table 3.2 shows the perplexity for different LSTM-RNN based SLM configurations. The numbers in the first column represent the product of the number of layers and the size of the hidden dimension for each model configuration. The results confirm that LSTM-RNN based SLMs are superior to n-gram based ones.

The experimental results indicated that the size of training set had a significant impact on the SLM performance. The SLMs trained on the NCHSDB dataset performed relatively better than those with HMC dataset. These findings suggest that a minimum of 1,000 records is necessary to train a good SLM.

Table 3.1: 2-gram ($bigram$) sleep model probability table for HMC train split.

Previous Sleep Stage	Next Sleep stage				
	W	REM	$N1$	$N2$	$N3$
W	0.854	0.001	0.138	0.003	0.000
REM	0.016	0.907	0.066	0.010	0.000
$N1$	0.109	0.080	0.498	0.311	0.000
$N2$	0.019	0.014	0.062	0.864	0.040
$N3$	0.007	0.001	0.007	0.063	0.921

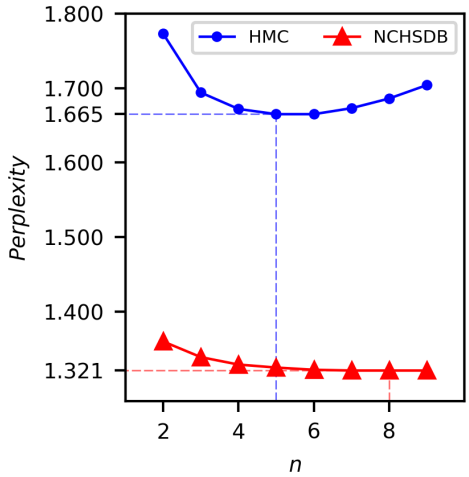


Figure 3.3: The test set perplexity of n-gram sleep models with HMC or NCHSDB dataset.

Table 3.2: The validation and test set perplexities of LSTM-RNN based SLMs trained with training split of HMC or NCHSDB dataset. The numbers in the first column represent (the number of layers) $\times$ (the size of the hidden dimension) for each SLM.

LSTM-RNN Sleep Model	HMC		NCHSDB	
	Valid. set	Test set	Valid. set	Test set
2× 256	**1.546**	1.609	1.293	1.287
2×1024	1.547	**1.608**	**1.291**	**1.286**
4× 256	1.547	1.613	1.293	1.287
4×1024	1.548	1.610	1.291	1.286

The performance of the SLM was affected by the characteristics of the sleep sequence, as evidenced by the difference in performance between the LSTM-RNN based SLM trained on the NCHSDB dataset and tested on the HMC test set.

The SLM was configured with 2 LSTM layers and a hidden dimension of 1,024 and achieved a perplexity of 1.286 on the NCHSDB test set, but only 1.608 on the HMC dataset. This performance difference can be attributed to the fact that the sleep patterns in the HMC dataset, which consists of records from adults, differ from those in the NCHSDB dataset, which is obtained from pediatrics.

3.4.2 Beam Search Decoding for Combining Signal and Sleep Models

We combined the 4-channel and 1-channel sleep signal models, explained in Section 3.2.2, with the LSTM-RNN based SLM through beam-search decoding. We used a beam width of 128 and searched for the best α on the validation set. The 4-channel signal model was combined with the LSTM-RNN based SLM from Section 3.4.1 with an optimum α, as described in Equation (3.2), of 0.12 and a beam width of 128. The evaluation results are shown in Table 3.3. The classification results based on greedy decoding for the signal models that do not utilize SLMs are labeled as 'Signal Model'. Despite our attempts to improve decoding performance by incorporating the SLM and varying the α values from 0.2 to 1.5, the results were not notably affected. This is likely due to the highly informative nature of the 4-channel signal model, which includes two channels of EEG and one channel each of EMG and EOG. However, when the 1-channel signal model was combined with the same SLM, there was a 6.5% improvement in Kappa score and a 4.3% improvement in accuracy. This improvement was observed despite the lower accuracy of sleep-stage classification when using EOG alone compared to the 4-channel PSG signal.

We used a beam width of 128 and searched for the best α on the validation set, and the results of the α search are listed in Table 3.4. The findings from this study indicate that the integration of SLMs into sleep-stage classification can offer significant

Table 3.3: The performance of the LSTM-RNN SLMs with 4 or 1 channel signal models. The SLMs had 2 layers of LSTM with 1,024 hidden dimensions.

Sleep Stage Classification Model	α	Valid. set		Test set	
		Kappa	Acc.	Kappa	Acc.
4-Ch. Signal Model	N/A	0.741	0.804	0.680	0.759
+ SLM Decording	0.12	0.680	0.759	0.680	0.759
1-Ch. Signal Model	N/A	0.505	0.629	0.464	0.602
+ SLM Decording	0.42	0.596	0.698	0.519	0.644

Table 3.4: The effect of probability weighting factor. The beam width was 128 and 1-channel signal model with 2×1024 LSTM-RNN SLM was evaluated.

α	Valid. set		Test set	
	Kappa	Acc.	Kappa	Acc.
0.34	0.587	0.692	0.517	0.642
0.38	0.592	0.696	0.516	0.642
0.42	**0.596**	**0.698**	**0.519**	**0.644**
0.46	0.588	0.693	0.515	0.640
0.50	0.594	0.697	0.511	0.638

benefits, particularly when working with restricted signal information, such as a single input channel or surrogate signals. This approach shows potential for application in other input modalities or different contexts to enhance sleep-stage classification.

3.5 Conclusion

Our study proposed sleep models for predicting the next sleep stage, which can significantly enhance sleep classification accuracy by utilizing information from numerous sleep-stage sequences and extending the context length. Our models were applied to sleep-stage classification model using a single EOG sensor. Sleep-stage sequences, rather than the signal sources, are the only data required to train sleep models, making them compatible with various sleep archives. Furthermore, once sleep models have been built, they can be employed in sleep-stage classification using different sensors, such as PPG, EOG, or ECG.

Chapter 4

SINGLE-CHANNEL ECG-BASED SLEEP STAGE CLASSIFICATION WITH END-TO-END TRAINABLE DEEP NEURAL NETWORKS

4.1 Introduction

A reliable and simple sleep test is imperative as sleep disorders are becoming more prevalent. Polysomnography (PSG) is the conventional standard sleep test that uses multiple sensors such as electroencephalogram (EEG), electrooculogram (EOG), electromyography (EMG), electrocardiogram (ECG), and respiration sensors [10]. However, PSG-based tests are not practical for everyday monitoring due to their inconvenience and high cost. A single-channel EEG and EOG test can classify sleep stages, but it still requires cumbersome sensors that disrupt sleep [46, 47].

To address these issues, we investigated sleep stage classification using single-channel ECG signals, which are easy to use and commonly used for diagnosing heart diseases [73]. Although it is more convenient than EEG or EOG, ECG signals alone do not provide strong information for sleep stage classification. In ECG-based sleep stage classification, heart rate variability (HRV) is considered the key information [2–5]. HRV reflects the time change between heartbeats and indicates physical and mental

health states.

In previous studies, researchers primarily utilized manually selected features such as the mean, median, percentile, and predefined frequency and entropy characteristics of heart rate and R-peak intervals for ECG-based sleep testing, but selecting the optimal manual features can be challenging. [12]. Research in speech recognition suggests that neural features that are fully trainable can lead to better results than hand-picked ones [39]. Inspired by these speech recognition studies, we used a ContextNet applied to ECG spectrograms to obtain the features [22]. Sleep stage classification can also benefit from sequence learning algorithms as sleep stages often follow specific sequences. Previous studies utilized Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for sequence learning, but we employed a Transformer model to incorporate the entire night's sequence information [23].

In sleep stage classification, sensor signals are divided into 30-second intervals, known as epochs. Based on guidelines [7], sleep experts classify each epoch into one of five stages: wake (W), rapid eye movement (REM), and three non-REM phases (N1, N2, and N3). These expert-classified labels are used for training and testing.

The structure of this paper is as follows: Section 4.2 provides a brief overview of the ISRUC-Sleep dataset utilized in the experiments and outlines the preprocessing method applied. Section 4.3 presents the proposed model. The experimental results are presented in Section 4.4. Finally, the paper is concluded in Section 4.5.

4.2 Dataset and Preprocessing

This study utilized the publicly available ISRUC-Sleep dataset, consisting of 126 sleep recordings obtained from the Sleep Medicine Centre of the Hospital of Coimbra University [74], and categorized into five sleep stages: W, REM, N1, N2, and N3, according to the American Academy of Sleep Medicine Guidelines [7]. Details of the ISRUC-Sleep dataset subgroups are provided in Table 4.1.

Table 4.1: ISRUC-Sleep dataset details. Sub-group2 consists of split sleep recordings with 8 subjects, resulting in a total of 16 recordings.

Sub-group	Age (mean, std.)	Subjects (male, female)	Remarks
1	20∼85(51, 16)	100 (55, 45)	Subjects with sleep disorders
2	26∼79(47, 19)	8 (6, 2)	
3	30∼58(40, 10)	10 (9, 1)	Healthy subjects

Each PSG recording consists of 19 channels, but the X2 channel (ECG) was used as the input signal, and underwent bandpass filtering with a passband frequency range from 0.3 Hz to 45 Hz. The filtered signals were then normalized using the Z-score method, as shown in equation (4.1),

$$\mathcal{Z}\left(s_t^i\right) = \frac{s_t^i - mean(s^i_{t_0,\ldots,t_{T-1}})}{std(s^i_{t_0,\ldots,t_{T-1}})}, \tag{4.1}$$

where s_t is a signal value at time t, T is the total length of the signal, and i is the record identification number in the training or test set. We transformed the filtered signal, $\mathcal{Z}\left(s_t^i\right)$, into a spectrogram using a short-time Fourier transform (STFT), and then divided it by the length of each epoch before being normalized using the Z-score method again. Finally, the normalized spectrogram of each epoch was used as the input for the model.

4.3 Proposed Model

The proposed model consists of three main components: a feature encoder, a sequence learning model, and a classifier. The feature encoder extracts the relevant features from the spectrogram of the ECG signals for each epoch. The sequence learning model then processes the sequence of feature vectors over the entire night, which can be over 900 epochs in length. Finally, the classifier uses the output of the sequence learning model to classify the sleep stage for each epoch, as illustrated in Figure 4.1.

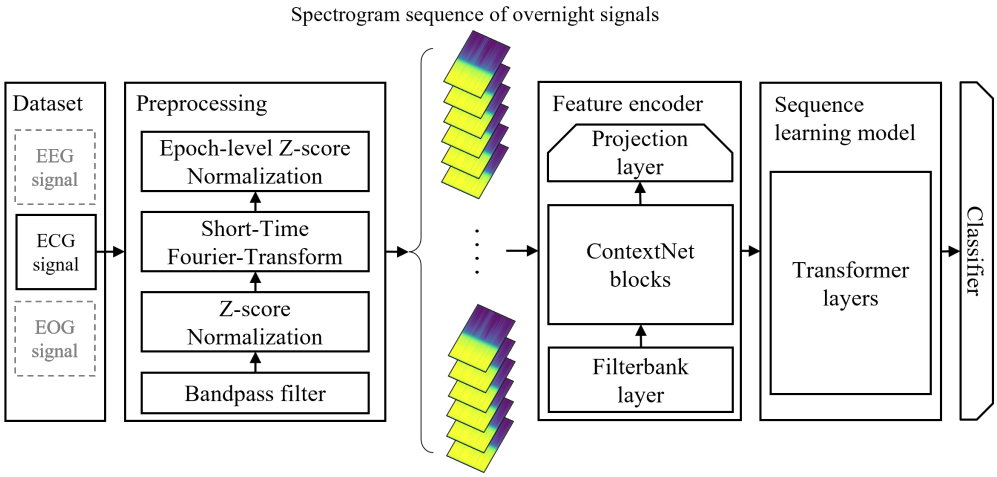


Figure 4.1: The proposed model overview and workflows. Please refer Section 4.3 for more details.

4.3.1 Feature Encoder

Filterbank layer

The filterbank layer in the feature encoder assigns weights to different frequency bins to reduce the dimensionality of the ECG signal in the frequency domain. Unlike using predetermined filterbank frequencies, a trainable layer is employed [27]. In addition, the output from this layer is modified using SpecAugment, a data augmentation technique for 2D signals like spectrograms [75]. SpecAugment modifies the spectrogram, which is an image representation of the ECG waveform, and has been shown to improve test accuracy in the experiments conducted.

ContextNet block

The ContextNet block is a type of CNN-based model that was originally developed for speech recognition tasks [22]. It employs depth-wise separable convolution and squeeze-and-excitation modules to extract features from the input signal.

In the depth-wise separable convolution, one-dimensional group convolutions are applied first along the time dimension without affecting the frequency direction, and

other convolutions are performed similarly along the frequency dimension. This reduces the correlation between the frequency and time dimensions.

The squeeze-and-excitation module takes the average of the inputs along the time dimension to produce a frequency-dimensional representation, which is then compressed and restored by a bottleneck-structured layer. Finally, the representation is multiplied with each corresponding frequency-dimensional vector through a sigmoid activation. Therefore, the squeeze-and-excitation module captures interdependencies between channels, and it does this by learning channel weights that reflect their relative importance. This allows the network to focus on the most informative channels while suppressing irrelevant or noisy ones. This module has been shown to be effective in various tasks including speech recognition and sleep stage classification.

Projection layer

The projection layer in the feature encoder flattens the output from the ContextNet block and produces a final output vector that contains the features for each epoch. This vector is then passed to the sequence learning model for further processing. The projection layer essentially reduces the output from the ContextNet block, which may have a high dimensionality due to the use of multiple convolutional layers, to a lower-dimensional representation that can be easily fed into the sequence learning model.

4.3.2 Sequence Learning Model and Output Classifier

The sleep stage pattern shows some typical sequences, such as transitions from REM stage to N2 or N3, followed by a return to REM. These rotations between REM and Non-REM stages occur about 3 to 5 times throughout the night. To leverage the sequential nature of sleep stages, our work employed Transformer layers [23]. The Transformer layer consists of a self-attention block and a feedforward block, and it has been widely used in many sequential tasks such as language modeling, speech recognition, and question-answering problems [24, 26, 39]. The self-attention block characterizes

each sequential input based on its unique position and relationship to other inputs. It can use temporal information that does not diminish over distance, unlike RNNs. In our prior research, we demonstrated that Transformer models performed better than RNNs in sleep stage classification using EEG, EMG, and EOG signals [47]. In this system, the Transformer layer processes the sequence of feature vectors over the entire night, which can be over 900 epochs in length.

The output of the Transformer layer is passed to the classifier, which maps the output to the probability distribution of sleep stages for each epoch.

4.3.3 Model Details

To process the ECG signals, we first transform them into spectrograms using a 512-point STFT with 0.5 second 'Hann' windows and 0.25 second intervals. This results in spectrograms with 256 frequency bins and 120 time bins, which serve as the input to the model in the form of a tensor with a shape of $(B, L, 256, 120)$, where B represents the batch size and L represents the sequence length.

The filterbank layer, shown in Figure 4.1, applies 128 trainable kernels with a size of 3 and a stride of 2 to the 256-frequency spectrogram, generating 128 outputs. These outputs are then fed into the feature encoder, which consists of a ContextNet model with three blocks. The first block has a configuration of $(1, 128, 256, 5)$, and the second and third blocks have configurations of $(5, 256, 256, 5)$ and $(5, 256, 128, 5)$, respectively. The last ContextNet block performs downsampling by half in the time domain, and the projection layer consists of a ContextNet block of configuration $(2, 128, 64, 5)$ and a fully-connected layer with an output size of 256. The final output of the feature encoder is a tensor of shape $(B, L, 256)$, which is then fed into the sequence learning model.

The sequence learning model employs a stack of 8 Transformer layers, each having 256-dimensional input and output, 4 attention heads, and a 1,024-dimensional feedforward block. Finally, the classifier outputs the probabilities of 5 classes for all $B \times L$

epochs.

4.4 Experimental Results and Comparison

4.4.1 Training and Testing

We measured the performance of the network using Cohen's kappa score, κ, which compares two outcomes for the same sample. Therefore, κ is a reliable metric in an environment with an unbalanced class distribution or a few number of classes.

We performed 5-fold cross-validation experiments, with each fold containing 25 or 26 sleep records. Each fold model started training with random initial parameters, and we used no validation set. To ensure that recordings from one subject do not exist in different folds, recordings in subgroup2 resulting from the split sleep test were tied according to the subject before randomly shuffling all recordings.

We trained the model using the Adam optimizer and a 3-staged learning rate scheduler [24, 76]. At first, the learning rate increased linearly from 0 to 0.0005 over 150 iterations. The learning rate was maintained for $1{,}200$ iterations in the second step. In the final step, the learning rate decreased according to the cosine annealing schedule for $1{,}750$ iterations. A total of $3{,}000$ training iterations covered approximately 120 epochs in each fold training set. We set B to 4. For each fold, the model was trained with the same hyper-parameter settings, but with different random seeds.

4.4.2 Results with the Developed Model

Table 4.2 presents a comparison of the results obtained by our proposed method and previous works. In this comparison, 'Spectrogram' indicates our proposed model, while 'Signal' indicates the direct application of ContextNet-based feature extraction to the input ECG signal. Although the results of directly applying ContextNet to the ECG signal are not as good as those of the spectrogram, it is still better than the previous CNN-based model [1]. Our experiments have shown that the spectrogram form

Table 4.2: Experimental results utilizing the proposed model. The experiment marked with * represents the average outcome of five trials utilizing different random seeds, and the experiment marked with † used the model from [1]. We referred [2] to generate manual features.

Input feature	κ	Acc.	F1 scores				
			W	N1	N2	N3	REM
Spectrogram*	**0.473**	**0.599**	0.733	0.312	0.591	0.622	0.556
Manual features	0.436	0.569	0.620	0.270	0.602	0.633	0.563
Signal	0.212	0.398	0.498	0.233	0.442	0.361	0.267
Signal†	0.096	0.340	0.360	0.041	0.288	0.217	0.159

is more suitable for end-to-end neural network training. The results presented in Table 4.2 were implemented by us using information from references or correspondence with authors.

Table 4.3 demonstrates the impact of sequence length and data augmentation on the performance of the model. We compared the results obtained by using sequence lengths of 60 and 1 with that using the full overnight sleep sequence length. Note that the sequence length of 60 corresponds to a duration of 30 minutes. The results show that increasing the sequence length to the entire duration of overnight sleep significantly improves the performance of the model. Moreover, the results indicate that SpecAugment, a powerful data augmentation method that can be applied to 2-dimensional signals such as spectrograms, plays a critical role in increasing the test accuracy [75]. Although we also tried other regularization methods such as adding Gaussian noise, time stretching, inverting amplitude, and amplitude clipping, these methods did not result in improved performance. It is important to note that ECG is a one-dimensional signal, and its amplitude does not contain informative information. In contrast, timing variation is an important factor in the classification of sleep stages, and there are limited options for data augmentation in this regard.

4.4.3 Comparison to Previous Research

We compared the experimental results obtained by the proposed method with previous studies on ECG-based sleep stage classification. Table 4.4 summarizes the comparison, indicating that previous studies utilized hand-picked features, while the proposed method used ContextNet derived features. The hand-picked features used in a previous study [2] were 182-dimensional and included both time-domain features such as heart rate, R-R interval, percentiles of HR and RR, etc., and 12 frequency-domain features. However, the design of these hand-picked features was challenging because it was a trial-and-error approach [12, 77].

The proposed method evaluated the performance using the Cardiology Challenge 2018 (CinC2018) dataset [78], which was used by a previous study [3]. The previous study only utilized stable sleep periods that had constant sleep stages within a 5-minute window. In contrast, our approach considers all epochs in the CinC2018 dataset for 5-

Table 4.3: Cohen's kappa score and accuracy under various sequence lengths and applying SpecAugment or not. Other conditions remain unchanged. Results of 5-fold cross-validation experiments utilizing the proposed model

Sequence length	SpecAugment	κ	Acc.
Overall	○	**0.477**	**0.600**
Overall	×	0.374	0.520
60	○	0.334	0.494
1	○	0.220	0.406

Table 4.4: Comparition with other papers [2–5]. All listed model used one ECG channel for 4 or 5 stage sleep classification. To convert 5 sleep stages to 4 stages, we merged the N1 and N2 stages as one stage [2].

Model	Method	Dataset	Record	Stage	κ	Acc.
Mustafa et al. [5]	Manual features + LSTM	SIESTA	588	4	0.61	0.77
Li et al. [3]	Manual features + CNN + SVM	CinC2018	994	4	0.31	0.66
Fonseca et al. [4]	Manual features + Bayesian discriminant	SIESTA	48	4	0.49	0.69
Mitsukura et al. [2]	Manual features + RNN	N/A	50	5	N/A	0.66
Ours	Spectrogram + CNN + Transformer	ISRUC-Sleep	126	4	0.54	0.69
			126	5	0.47	0.60
		CinC2018	994	4	0.60	0.76
			994	5	0.52	0.65

stage sleep classification, following the same preprocessing and training procedure as the ISRUC-Sleep dataset. Despite the disadvantage of classifying all epochs, our results showed good improvement compared to the previous study [3].

The 4-stage classification achieved by the proposed method was also comparable to another study [5]. However, it's important to note that study [5] used the test set for early stopping during training, which might have impacted the accuracy. Our experiments did not consult the test set during training.

Although making a direct comparison between the results of different studies is challenging due to differences in datasets and experimental settings, the proposed method achieved superior results despite using a relatively small amount of data. A previous study on the ISRUC-Sleep dataset, which included EEG, EOG, and EMG, achieved an accuracy of 71% for 5-stage classification [29],while the proposed method, which uses only a single ECG channel, achieved an accuracy of 60% in classifying 5 sleep stages, despite the limitations of the input signal.

4.5 Conclusion

We present a neural network model for classifying sleep stages using only single-channel ECG signals. Our approach focuses on two main goals: (1) using fully neural network-based trainable features derived from the ECG spectrogram instead of hand-picked ones and (2) utilizing the entire night's information for sequence learning. The model combines ContextNet blocks and Transformer layers to form the feature encoder and the sequence learning model. The use of spectrogram input provides the benefit of leveraging SpecAugment, a strong data augmentation technique for two-dimensional signals. Our results outperformed those of previous studies that employed many manual features. As the model is trained end-to-end, we anticipate improved results with the collection of more training data in the future.

Chapter 5

SOUND-LEVEL TOLERANT SLEEP APNEA DETECTION FROM SLEEP AUDIO

5.1 Introduction

Obstructive sleep apnea (OSA) is a common issue in the general adult population aged 30-70. It affects around 15-50% of the population and is consistently growing [79–81]. OSA is a recurrent troublesome breath caused by the partial or complete collapse of the upper airway during sleep. When the upper airway completely collapses, it is called apnea, and when it partially collapses, it is called hypopnea. The genioglossus muscle, which is the most important upper airway dilator muscle, contracts with each inspiration to prevent posterior collapse of the tongue, causing narrowing or blockage of the airway. If the cause of apnea is a nerve systematic problem, which means that the brain fails to send a signal to breathe, is called central apnea. When the apnea is caused by both the nerve system and obstruction in the upper airway, it is called mixed apnea. OSA is associated with either cortical arousal or a fall in blood oxygen saturation, resulting in increasing intermittent hypoxemia, sleep fragmentation, vascular disease, metabolic abnormalities, and inflammation. The sleep analysis manual defines OSA when the apnea or hypopnea lasts for at least 10 seconds [7].

Polysomnography (PSG) is the standard diagnostic method for sleep disorders like OSA. However, due to its cost and time-consuming nature, more affordable and accessible alternatives such as home sleep apnea tests and portable monitoring devices have been developed, albeit with possibly lower accuracy than PSG [82, 83].

Existing research has attempted to classify apnea events based on sleep audio. Studies by [84, 85] use features derived from sleep audio while studies [6, 86–88] employ features from the spectrogram domain. Studies [6, 85, 87] rely on deep neural network architectures comprising convolutional, recurrent, and transformer layers.

This study introduces a neural network model designed to classify OSA using audio signals. The proposed model processes sleep sound data from an entire night, identifying the temporal location of OSA events by classifying them into either 5 or 3 categories every 10 seconds. Adaptable for use on smartphones, this model provides a convenient, accessible solution for individuals aiming to regularly monitor their sleep issues. Furthermore, the proposed study calculates the apnea-hypopnea index (AHI) based on its predictions, indicative of OSA severity. We believe this model offers significant potential for those seeking daily sleep health monitoring.

5.2 Datasets

We sourced the publicly available data for this study from the Sleep Study Unit of the Sismanoglio – Amalia Fleming General Hospital of Athens. The PSG-Audio dataset comprises 272 polysomnography (PSG) recordings, synchronized with high-quality audio [89]. In this study, we utilized version 3 of the dataset.

The patients in the PSG-Audio dataset are classified according to sleep apnea severity, indicated by the Apnea-Hypopnea Index (AHI). The AHI is calculated as the ratio of the total number of apnea-hypopnea events to the total sleep time, expressed

in the number of episodes per hour, as shown in Equation 5.1.

$$\text{AHI} = \frac{\text{number of apnea-hypopnea events}}{\text{total sleep time}} \text{ episodes/h} \tag{5.1}$$

The severity levels are classified as 'Severe', 'Moderate', 'Mild', and 'Normal', with AHI thresholds of 30, 15, and 5, respectively. In the PSG-Audio dataset, 88.7% of patients are classified with 'Severe' apnea, while 9.0%, 0.9%, and 1.4% are categorized as 'Moderate', 'Mild' apnea, and 'Normal', respectively. Of all the patients, 76% are male, and the mean age of the entire cohort is 57.9 years, with a slightly lower mean age of 57.2 years for females.

The PSG-Audio recordings included in the dataset represent the first part of a split night sleep study and are approximately 4 hours in duration. In cases where the patient was not eligible for a split night study, all-night PSG recordings were collected instead. The microphone used to capture the audio signals is positioned above the patient's bed, at a height of over 1 meter, and placed directly above the patient's head. The audio signals are sampled at a rate of 48,000 Hz and stored in a 16-bit waveform audio format. The annotations of the dataset cover various aspects of sleep analysis, including sleep stages, cardiac episodes, limb movements, respiratory episodes, and more.

5.2.1 Data Preprocessing

We extracted features from the audio signal using the Mel-filter bank, typically employed in automatic speech recognition [90]. The data processing pipeline is shown in Figure 5.1, and consists of a four-step procedure: (1) resampling the signal from 48,000 Hz to 2,000 Hz to minimize computation in the subsequent pipeline, (2) transforming the signal into Mel spectrogram, (3) converting the filter output to a decibel (dB) scale, and (4) normalize the spectrum in each frequency bin. Our method incorporated a 4,096-point short time Fourier transform (STFT), with a Hanning window

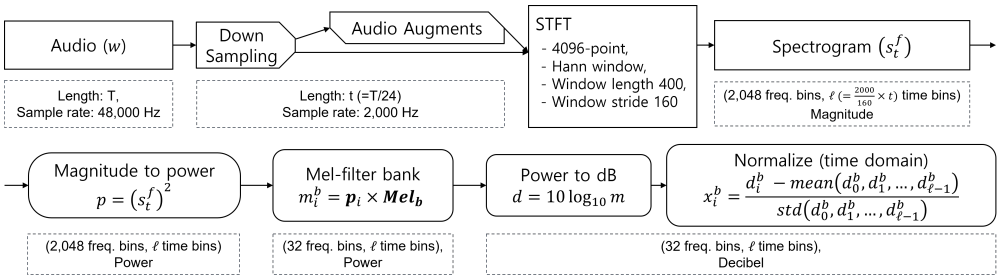


Figure 5.1: Data processing pipelines.

length of 400 and a slide of 160. The feature output remains unaffected by the amplitude of the input audio due to normalization. However, smaller scale input can be more vulnerable to environmental noise. The performance is unaffected by the gain of the microphone because it applies the same gain to both the input signal and the noise.

We categorized the apnea events in the annotation into 5 classes based on an epoch length of 10 seconds: Obstructive Apnea (OA), Hypopnea (HP), Mixed Apnea (MA), Central Apnea (CA), and Normal sleep (N). This epoch length was chosen based on the requirement that apnea events should persist for at least 10 seconds [7]. If an apnea event spans two continuous epochs, both epochs are marked as part of the apnea event.

5.3 Model Architecture

The developed model comprises a feature extractor and a sequence learning network. The feature extractor is responsible for drawing relevant information from the input audio signals, which are then fed into the sequence learning network for classification. In this case, the feature extractor network utilizes the ContextNet block as its primary architecture [22]. The ContextNet block excels at capturing contextual information by aggregating information from both past and future context frames, allowing for a more comprehensive understanding of the context. Figure 5.2 illustrates the ContextNet block architecture, which consists of depth-separable convolution, squeeze-and-excitation, and residual modules.

The sequence learning network is responsible for capturing the sequential patterns in the extracted features and making predictions for the apnea event classification task. The network comprises Transformer layers, a type of neural network architecture that has proven highly effective for sequence modeling tasks such as natural language processing and audio analysis [23, 24, 26, 39].

5.3.1 Feature Extractor Network

Our model incorporates five stacked ContextNet blocks, structured as detailed in Table 5.1. Table 5.1 specifies four key parameters pertaining to the ContextNet blocks: (1) the number of 1-D convolution layers in the depth-separable convolution module, (2) the quantity of output channels yielded by the ContextNet block, (3) the kernel size of the depth-wise convolution layer, and (4) whether the block applies downsampling respectively. Upon processing through the final ContextNet block, the output is flattened along the channel dimension and then directed into a fully-connected neural network yielding an output size of 256. Consequently, the feature extraction network outputs a matrix of size $L\times$ 256, where L denotes the input epoch length. As a result, the feature extraction network formulates a 256-dimensional feature vector encapsulating pertinent information for apnea event classification for each input epoch.

5.3.2 Sequence Learning Network

The sequence learning network is responsible for learning the sequential patterns in the extracted features and making predictions for the apnea event classification task. In

Table 5.1: The configuration of ContextNet blocks.

#Layers	#Channels	Kernel size	Down sampling
3	128	5	○
5	128	7	×
5	128	7	×
5	128	7	×
5	64	5	×

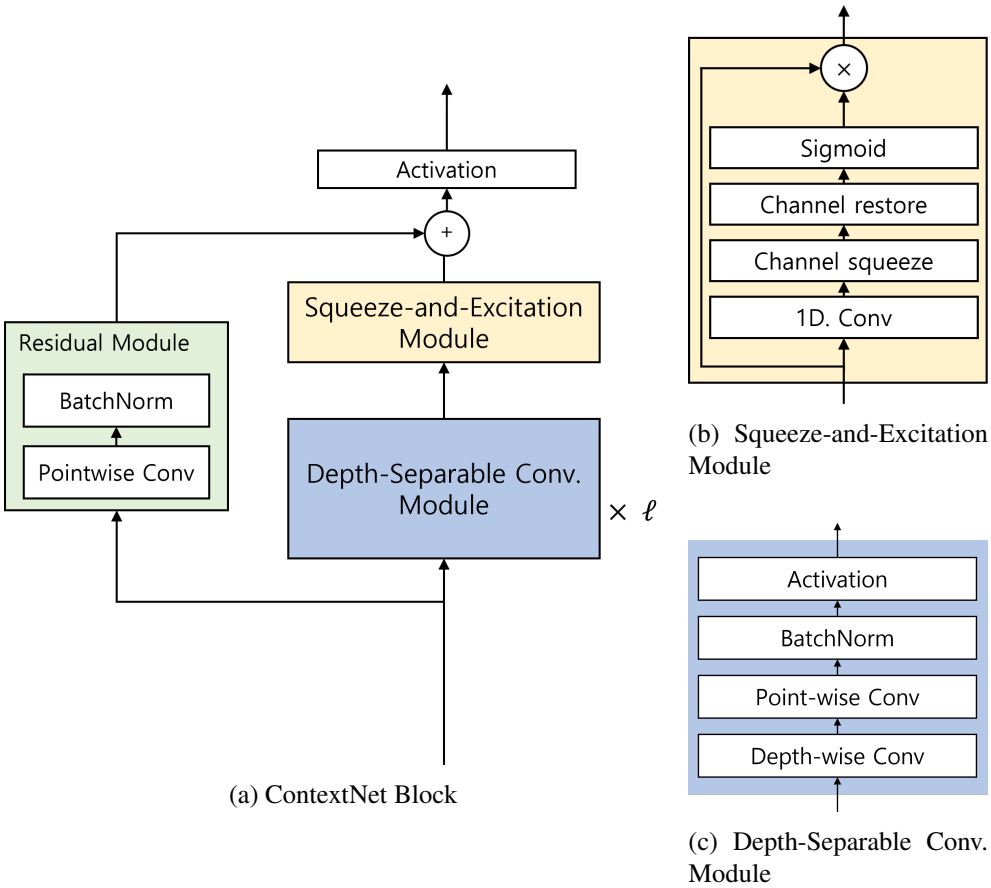


Figure 5.2: The architecture of ContextNet block, squeeze-and-excitation module and depth-separable convolution module.

our implementation, the sequence learning network consists of 4 layers of Transformer with 256-dimensional hidden states [23]. The multi-head attention mechanism in each Transformer layer has 4 heads, and the dimension of the feed-forward layer is set to 1,024.

Since the duration of apnea events can vary widely from at least 10 sec to over 30 sec, we found it important to capture the context before and after the event to help identify the start and end points accurately. This is important in accurately classifying apnea events and distinguishing them from other normal or abnormal respiratory patterns. Therefore, the sequence learning network captures patterns in inter-epochs

of continuous 2 to 6 time steps and detects irregular patterns. We limited the Transformer attention length to 7 by masking time steps that have a relative distance greater than 3 [91]. Each input incorporates information from its preceding and subsequent 3 epochs, as well as its own epoch, resulting in a total of 7 epochs or 70 seconds. This is because we found that a continuous 2 to 6 time steps were enough to classify apnea events [87].

By utilizing this architecture, we can effectively capture the sequential patterns in the extracted features and classify apnea events with high accuracy.

5.4 Experimental Results

5.4.1 Data Augmentation

We applied six audio and one spectral augmentation methods to provide regularization. The audio augmentations included (1) adding Gaussian noise with absolute amplitude values or according to signal-to-noise ratio, (2) adding background noises, and (3) convolution with a random impulse response [92]. A single augmentation, randomly chosen from these options, was potentially applied to each segment with a 40% probability.

Table 5.2 presents the results of preliminary experiments conducted on approximately 30% of the dataset. The experiments demonstrate that applying impulse response and adding Gaussian noise amplitude are more effective than other methods in improving the model's performance on the subset of the validation and test sets, respectively.

We also employed SpecAugment for spectral augmentation, applying two frequency masks (size 2) and two time masks (size 20) to each epoch's Mel-filter bank feature [93].

Table 5.2: Effectiveness of each augmentation.

Audio Augmentation	Validation subset		Test subset	
	κ	Acc.	κ	Acc.
-	0.434	0.748	0.439	0.710
Add Background Noise	0.436	0.754	0.431	0.718
Add Gaussian Noise (amplitude)	0.437	0.754	0.453	0.727
Add Gaussian Noise (SNR)	0.438	0.746	0.441	0.717
Apply Impulse Response	0.446	0.752	0.448	0.721
Gain	0.437	0.747	0.446	0.717
Pitch Shift	0.432	0.742	0.444	0.718

5.4.2 Metric

We utilize accuracy and the Cohen's Kappa score (κ) for the metric of performance of models. The κ provides a measure of the agreement between the predicted and actual classes, taking into account the agreement that would be expected by chance alone. Equation 5.2 is definition of the κ,

$$\kappa \equiv 1 - \frac{1 - p_0}{1 - p_e}, \tag{5.2}$$

where the parameter p_0 represents the observed agreement between predicted outcomes and ground truth labels. Conversely, p_e symbolizes the hypothetical probability of agreement by mere chance, which provides a measure of the random alignment between the model's predictions and the actual labels in the dataset. A κ of 1 indicates perfect agreement, while a score of 0 indicates agreement that is no better than chance. In the case of imbalanced datasets, the κ can be a more informative metric than accuracy.

5.4.3 Training Details

The PSG-Audio dataset is randomly split into training, validation, and test sets, consisting of 173, 44, and 55 PSGs, respectively. The audio signals in the training set were divided into segments, each with a length of 64 epochs. This decision was based on

empirical experiments that demonstrated the inefficiency of shorter or longer segment lengths. However, in the validation and test sets, we did not segment the audio signals; instead, we inferred the entire audio.

We utilized BatchNorm, LayerNorm, and Dropout techniques to regularize our neural networks [36, 94, 95]. Specifically, we applied Dropout between every layer except before the last classification layer with a probability of 25% to prevent overfitting. We trained the model using the Adam optimizer and a 3-staged learning rate scheduler [24, 76]. The batch size for the training was set to 96.

5.4.4 Apnea Classification

Table 5.3 presents the experimental results of apnea event classifications using different models. Model-0 serves as the baseline model and does not incorporate any audio augmentation. Conversely, Model-1 employs augmented audio. Given our use of full-length of sleep audio, we encountered a class imbalance during model training. Specifically, the normal (N) class comprises about 66% of all epochs. To mitigate this issue, we applied weighted-cross entropy loss. Model-2 additionally uses weighted-cross entropy loss, where the class weight is calculated based on the inverse of the class distributions.

Given that both CA and MA are linked to brain activity, their identification predominantly relies on electroencephalogram signals from PSG. Consequently, distin-

Table 5.3: The experimental results of 5 and 3 -class apnea classification models.

Apnea Classes	Model No.	Audio Augment	Validation set		Test set	
			κ	Acc.	κ	Acc.
5	0	×	0.445	0.746	0.461	0.722
	1	○	**0.458**	**0.751**	0.463	0.724
	2	○	0.455	0.749	**0.464**	**0.724**
3	0	×	0.544	0.802	0.557	0.785
	1	○	0.556	**0.806**	0.571	**0.791**
	2	○	**0.561**	0.802	**0.577**	0.790

Table 5.4: Comprehensive review of related works, including an accuracy of AHI cut-off models using a cutoff value of 15. Though the accuracy of [6] is not reported, it demonstrates a sensitivity of 0.81.

Reference No.	Dataset	#PSG	Input feature	Model	Classes	Acc.	Year
[84]	Private	423	Audio features	Machine learning	Under / Over AHI cutoff	0.82	2022
[85]	Private	61	Audio features	CNN+LSTM	Under / Over AHI cutoff	0.84	2020
[86]	PSG-Audio	50	Mel spectrogram	CNN+LSTM	Apnea snoring / Normal snoring	0.99	2023
[87]	Private	1315	Mel spectrogram	SoundSleepNet	Apnea / Hypopnea / Normal	0.86	2023
[88]	Private	500	Spectrogram	CNN+FC	Apnea / Normal	0.91	2021
[6]	Private	157	Mel spectrogram	CNN+FC	Under / Over AHI cutoff	N/A	2023
This work	PSG-Audio	272	Mel spectrogram	Contextnet+Transformer	Apnea / Hypopnea / Normal	0.79	2023

guishing between CA and MA from OA using solely audio signals poses a considerable challenge. Therefore, we've opted to consolidate CA and MA into the OA category. Table 5.3 presents the experimental results when apnea events are classified into three categories, namely OA, HP, and N.

Recent studies, summarized in Table 5.4, have primarily employed binary classification with audio, utilizing CNN + LSTM or CNN + fully connected layers (FC) [6, 85–88]. Given that the majority of previous works focused on binary classification using different datasets, comparing our performance with existing studies becomes challenging. To address this issue, the creation of a large public dataset specifically designed for this study is crucially needed.

5.4.5 AHI Estimation

Based on the classification results, we estimated AHI of each patient by counting the number of transient events whose class was not N. With Equation 5.1 in Section 5.2, we calculated estimated AHI, denoted as AHI_E, with the validation set. We performed a calibration on the results by applying a first-order regression on the ground truth AHI

values. The calibrated AHI, AHI_{cal}, is given by Equation 5.3,

$$\text{AHI}_{cal} = f\left(\text{AHI}_E\right), \tag{5.3}$$

$$f\left(x\right) = p_1^\star x + p_0^\star, \tag{5.4}$$

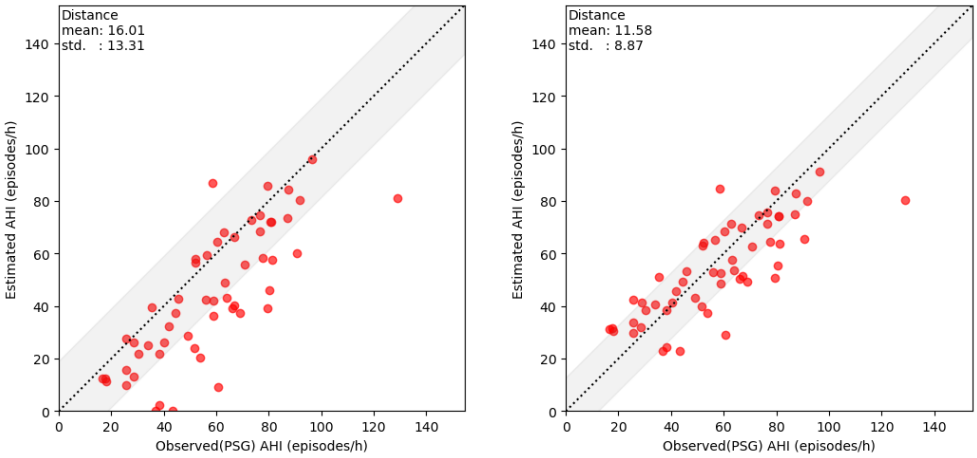
$$p_0^\star, p_1^\star = \arg\min_{p_0, p_1} \sum_{i=1}^{i=N} \left|\text{AHI}_T^i - \left(p_1 \text{AHI}_E^i + p_0\right)\right|^2, \tag{5.5}$$

where $f(x)$ is a linear function, $p_0^\star$ and $p_1^\star$ are the intercept and slope of the linear function, respectively. AHI_T is the ground truth AHI and i is audio record index from N audio examples [86].

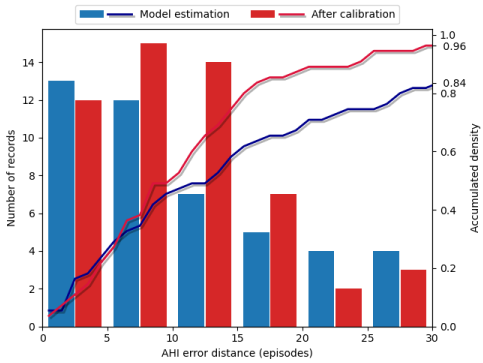
The AHI estimation results obtained from the best-performing models on the test set are illustrated in Figure 5.3, including the results after calibration. Each dot in the graphs represents an example in the test set and the dotted line represents an exact match between the estimated and the AHI values from PSG. The gray-colored areas represent the distance below one standard deviation of the difference between the observed and estimated AHI values. The graphs on the right side show calibrated AHI estimations based on Equation 5.3. The coefficients, $p_0^\star$ and $p_1^\star$, for calibration were calculated using examples from the validation set.

Figure 5.3b illustrates a histogram and cumulative density of the AHI distance between observed AHI and estimated AHI from the model. After calibration, the model captures 96% of test set examples with an AHI distance less than 30 and approximately 80% with an AHI distance less than 15.

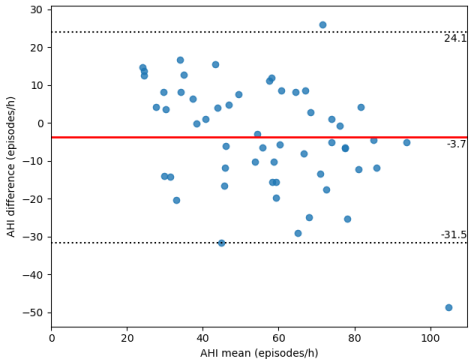
The Bland-Altman plot in Figure 5.3c contrasts observed and calibrated AHI. The horizontal axis indicates the average of the actual and predicted AHI, while the vertical axis displays the corresponding AHI estimation error. The two dotted horizontal lines denote $\pm 1.96\times$ the standard deviation (SD) of the AHI difference from the mean difference, corresponding to the 95% limits of agreement. Our analysis infers that the AHI estimation errors bear no correlation with AHI. Moreover, due to calibration, the mean of the AHI error gravitates towards zero.



(a) AHI estimation (left) and after calibration results (right).



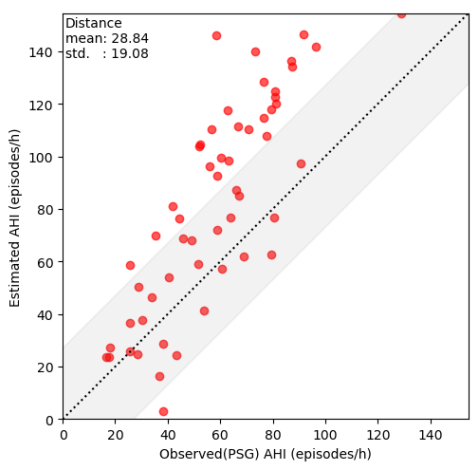
(b) AHI estimation distance histogram and accumulated density.



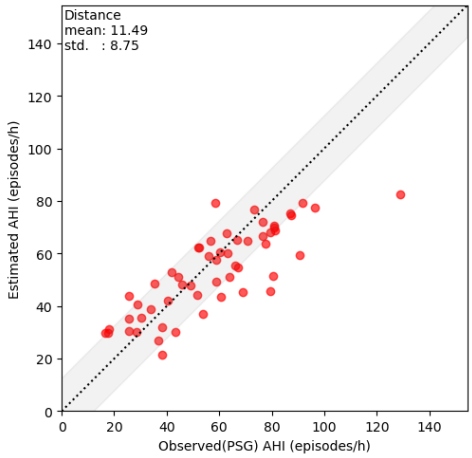
(c) Bland-Altman plot of observed and calibrated AHI.

Figure 5.3: AHI estimation results with the 5-class apnea classification model-2.

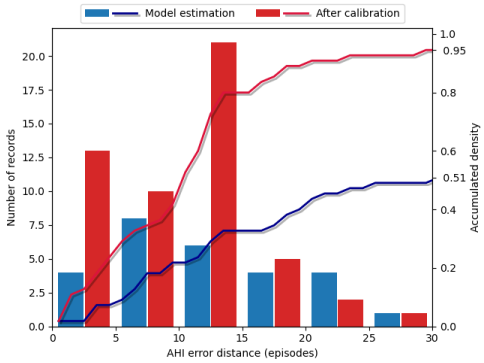
Figure 5.4 presents analogous plots derived from the 3-class model. Compared to the 5-class model, AHI calibration realizes a substantial enhancement. As shown in Figure 5.4b, both the mean distance and standard deviation (SD) are reduced by over half following calibration. Prior to calibration, as shown in Figure 5.4c, AHI estimation encompassed 51% of recordings within a distance less than 30. However, after calibration, this coverage increased to 95% of the test set examples under the same distance. Despite these improvements, the Bland-Altman plot for the 3-class model reveals results similar to those of the 5-class model.



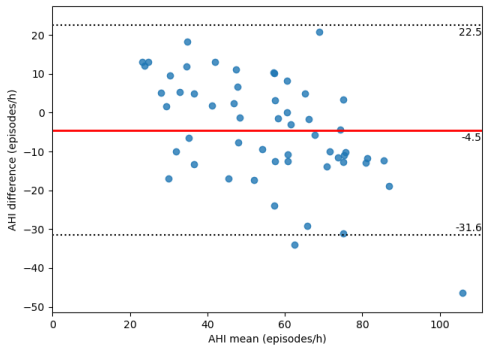
(a) AHI estimation results.



(b) Calibrated AHI estimation results.



(c) Estimation distance histogram and accumulated density.



(d) Bland-Altman plot of observed and calibrated AHI.

Figure 5.4: AHI estimation results with the 3-class apnea classification model-2.

We conducted performance tests with varying input audio levels ranging from -20 dB to +20 dB. Throughout the test, no significant performance changes were observed due to the scale normalization of the input audio at the feature extraction stage, as illustrated in Figure 5.1. It is evident that lower levels of breathing sound are more susceptible to environmental noise. To simulate this, we introduced noise corresponding to -30 dB of the reference amplitude. The reference amplitude was determined by measuring the root-mean-square (RMS) value from the top 10 audios with the highest RMS in the training set. Consequently, sound samples with lower RMS values were

disadvantaged in this test. We observed a slight decrease in accuracy in the noise-added case, from 79.1% to 78.9%. When the noise scale was set to -20 dB, the accuracy further dropped to 77.1%.

Table 5.5 presents confusion matrices of the 3-class apnea classification model-2 under the influence of added noise level of -30 dB in sleep audios in the test set.

5.4.6 Ablation Study for Model Design

In this subsection, the network architecture was modified by replacing the feature extractor or sequence learning network with FC. The modified network models are FC-Transformer and ContextNet-FC. In the FC-Transformer architecture, the feature extractor network was replaced with 4 layers of 512 dimensional FC. This network generates features for Transformer layers by feeding flattened Mel-filter bank features of an epoch. The input dimension of the first layer is the number of Mel-filter banks multiplied by duration of an epoch and divided by the stride of STFT. The input and output dimensions of the other layers are 512. On the other hand, the ContextNet-FC architecture utilizes the epoch-wise output of the feature extractor to feed a FC with 512 dimensions. This architecture has no capability of learning sequential information. The Transformer and ContextNet architectures of each model were configured in the same way as the proposed model. The Mel-filter bank configurations, window length,

Table 5.5: Confusion matrices for the 3-class apnea classification Model-2. The matrix on the left is derived from the test set, while the one on the right is generated from test set audio with added noise

		Model prediction			
		AP	HP	N	SUM
True class	AP	0.24	0.01	0.07	0.32
	HP	0.02	0.01	0.05	0.08
	N	0.06	0.01	0.54	0.61
	SUM	0.32	0.02	0.66	1.00

		Model prediction			
		AP	HP	N	SUM
True class	AP	0.23	0.01	0.09	0.32
	HP	0.02	0.00	0.05	0.08
	N	0.05	0.00	0.55	0.61
	SUM	0.30	0.02	0.69	1.00

Table 5.6: The results with different architectural composition with 5-class apnea classification.

Feature extractor	Sequence learning	Validation set		Test set	
		κ	Acc.	κ	Acc.
ContextNet	Transformer	**0.445**	**0.746**	**0.461**	**0.722**
FC	Transformer	0.356	0.714	0.375	0.687
ContextNet	FC	0.292	0.679	0.357	0.668

stride of window, and the number of filter banks were fixed at 400, 160, and 34, respectively. Table 5.6 shows the results of each model configuration. These results suggest that our feature extractor and sequence learning network structure are well-suited for the apnea classification task, and that each architectural choice is appropriate for the specific subtask assigned to the network.

5.5 Conclusion

The proposed DNN model predicts the incidence of sleep disturbances by analyzing distinct sounds produced during episodes of apnea or hypopnea. The architecture comprises a feature extractor leveraging ContextNet and a sequential learning module utilizing Transformer capabilities. The proposed system predicts one of five possible categories: Obstructive apnea, Hypopnea, Central apnea, Mixed apnea, and Normal. Although the classification accuracy for Central and Mixed apnea classes is lower, the system adeptly predicted Obstructive apnea and Hypopnea, which are more common. Even when the input sound level fluctuates between ± 20 dB, the proposed system maintains fairly consistent accuracy, addressing potential issues associated with smartphone sensitivity and sleep positioning.

Chapter 6

CONCLUSION

In this dissertation, we discussed data processing, network architectures, and improvement techniques for utilizing biosignals especially Polysomnography (PSG). PSG includes various signals, Electroencephalogram (EEG), Electrooculogram (EOG), Electromyogram (EMG), and Electrocardiogram (ECG), that are used as a diagnosis basis but cost and accessibility of PSG obstruct early screening and appropriate treatment. This dissertation aims to address the issues associated with PSG by utilizing a simplified input consisting of a single signal source. The objective is to make daily testing feasible and accessible from the comfort of one's own home.

In Chapter 2 of this dissertation, we conducted a comprehensive search for a suitable base architecture for a sleep stage classification model. We compared several models, including a simple fully-connected layer (DNN), a DNN with sequence concatenated inputs, a recurrent neural network (RNN), and a Transformer architecture with a structurally identical convolutional neural network (CNN) as a feature extraction network. Furthermore, we investigated the efficacy of each PSG signal, including EEG, EOG, EMG, and ECG, for the sleep stage classification task. While EEG is the primary source signal for distinguishing sleep stages, it requires sensors to be attached to the head, which is a burdensome process. Thus, we wanted to exclude EEG from our analysis while keeping in mind the ease of signal collection. The results showed

that the exclusion of EEG led to a drop in performance for each combination of signals and models. However, the Transformer model showed the possibility of a classification model without EEG signal, as it was able to access the overnight length of the signal and learn from it.

In Chapter 3, we introduced the SLeep Model (SLM) to improve the performance of sleep stage classification models using less informative ECG signals. Inspired by language models used in the decoding process of automatic speech recognition (ASR), the SLM predicts the next sleep stage based on the previous stages. We explored two different approaches to build the SLM, including n-gram models and RNN models. By incorporating the SLM into the sleep stage prediction model through beam search decoding algorithms, we were able to achieve improved performance. One of the major advantages of SLM is that it only requires sleep stage sequences instead of signal sources to train the models, making it compatible with various sleep archives. Furthermore, once the sleep models have been trained, they can be applied to sleep-stage classification using different sensors, such as PPG, EOG, or ECG. This flexibility makes the SLM a promising approach for sleep stage classification in diverse settings and applications.

In Chapter 4, we proposed a neural network model for sleep stage classification using only a single-channel ECG signal. We chose ECG signals as input sources due to their ease of acquisition by mobile devices such as smartwatches and in-house monitors designed for detecting cardiac-related diseases. The proposed model comprises a feature extraction network based on ContextNet architecture and a sequence modeling network based on a Transformer layer. Prior to feeding the ECG signal into the model, we carefully processed it using previously studied characteristics of the ECG signal and transformed it into a spectrogram format. Unlike previous research that focuses on heart rate variability and requires a hand-picking process, our proposed model eliminates the feature selection process by feeding the spectrogram-formatted signal to the model, enabling an end-to-end training process.

In Chapter 5, we developed a neural network model utilizing sleep audio for the Apnea-Hypopnea classification task. The approach is based on the fact that apnea or hypopnea generates characteristic breathing sounds, changes in breath lengths, and even snoring. We used a neural network architecture similar to the one used in the sleep stage classification models, consisting of ContextNet followed by Transformer. The data processing pipeline was adapted from the techniques commonly applied in ASR tasks. To simulate a scenario in which audio is recorded from a smartphone microphone, we augmented the audio by adding noises, changing the gain, or applying room impulse responses to make the model robust against low-quality, in-house audio sampling. Additionally, we estimated the Apnea-Hypopnea Index (AHI) from the prediction results and calibrated it with the validation results for better AHI estimation.

In summary, this dissertation demonstrates that instead of using complex PSG tests, it is possible to achieve sleep stage classification and sleep apnea prediction by utilizing simple sensors such as ECG or smartphones, provided that well-trained deep neural networks are employed. It is anticipated that this research will greatly contribute to improving the health of many individuals.

Bibliography

[1] D. Alvarez-Estevez and R. M. Rijsman, "Inter-database validation of a deep learning approach for automatic sleep scoring," *PloS one*, vol. 16, no. 8, p. e0256111, 2021.

[2] Y. Mitsukura, K. Fukunaga, M. Yasui, and M. Mimura, "Sleep stage detection using only heart rate," *Health Informatics Journal*, vol. 26, no. 1, pp. 376–387, 2020.

[3] Q. Li, Q. Li, C. Liu, S. P. Shashikumar, S. Nemati, and G. D. Clifford, "Deep learning in the cross-time frequency domain for sleep staging from a single-lead electrocardiogram," *Physiological measurement*, vol. 39, no. 12, p. 124005, 2018.

[4] P. Fonseca, X. Long, M. Radha, R. Haakma, R. M. Aarts, and J. Rolink, "Sleep stage classification with ecg and respiratory effort," *Physiological measurement*, vol. 36, no. 10, p. 2027, 2015.

[5] M. Radha, P. Fonseca, A. Moreau, M. Ross, A. Cerny, P. Anderer, X. Long, and R. M. Aarts, "Sleep stage classification from heart-rate variability using long short-term memory neural networks," *Scientific Reports*, vol. 9, no. 1, p. 14149, Oct 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-49703-y

[6] H. E. Romero, N. Ma, G. J. Brown, and E. A. Hill, "Acoustic screening for obstructive sleep apnea in home environments based on deep neural networks,"

IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 7, pp. 2941–2950, 2022.

[7] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, R. M. Lloyd, C. L. Marcus, and B. Vaughn, "The aasm manual for the scoring of sleep and associated events, version 2.4," *Chicago: American Academy of Sleep Medicine*, 2017.

[8] A. Rechtschaffen, "A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects," *Brain information service*, 1968.

[9] D. Shrivastava, S. Jung, M. Saadat, R. Sirohi, and K. Crewson, "How to interpret the results of a sleep study," *Journal of community hospital internal medicine perspectives*, vol. 4, no. 5, p. 24983, 2014.

[10] C. Armon, A. Roy, and W. Nowack, "Polysomnography: Overview and clinical application," *E-Medicine. March*, 2007.

[11] J. W. C. Medithe and U. R. Nelakuditi, "Study of normal and abnormal eeg," in *2016 3rd International conference on advanced computing and communication systems (ICACCS)*, vol. 1. IEEE, 2016, pp. 1–4.

[12] S. Khalighi, T. Sousa, G. Pires, and U. Nunes, "Automatic sleep staging: A computer assisted approach for optimal combination of features and polysomnographic channels," *Expert Systems with Applications*, vol. 40, no. 17, pp. 7046–7059, 2013.

[13] J. Ehiabhi and H. Wang, "A systematic review of machine learning models in mental health analysis based on multi-channel multi-modal biometric signals," *BioMedInformatics*, vol. 3, no. 1, pp. 193–219, 2023.

[14] A. Bulling, D. Roggen, and G. Tröster, "Wearable eog goggles: Seamless sensing and context-awareness in everyday environments," *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 157–171, 2009.

[15] ——, “It’s in your eyes: Towards context-awareness and mobile hci using wearable eog goggles,” in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 84–93.

[16] A. Ubeda, E. Ianez, and J. M. Azorin, “Wireless and portable eog-based interface for assisting disabled people,” *IEEE/ASME Transactions on mechatronics*, vol. 16, no. 5, pp. 870–873, 2011.

[17] D. C. Toledo-Pérez, J. Rodríguez-Reséndiz, R. A. Gómez-Loenzo, and J. Jauregui-Correa, “Support vector machine-based emg signal classification techniques: A review,” *Applied Sciences*, vol. 9, no. 20, p. 4402, 2019.

[18] M. A. Oskoei and H. Hu, “Myoelectric control systems—a survey,” *Biomedical signal processing and control*, vol. 2, no. 4, pp. 275–294, 2007.

[19] F. Shaffer and J. P. Ginsberg, “An overview of heart rate variability metrics and norms,” *Frontiers in public health*, p. 258, 2017.

[20] J.-S. Wang, G.-R. Shih, and W.-C. Chiang, “Sleep stage classification of sleep apnea patients using decision-tree-based support vector machines based on ecg parameters,” in *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*, 2012, pp. 285–288.

[21] Y.-C. Yeh, W.-J. Wang, and C. W. Chiou, “Feature selection algorithm for ecg signals using range-overlaps method,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 3499–3512, 2010.

[22] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, “Contextnet: Improving convolutional neural networks for automatic speech recognition with global context,” *Proc. Interspeech 2020*, pp. 3610–3614, 2020.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[26] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[27] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Seqsleepnet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 3, pp. 400–410, 2019.

[28] Z. Cui, X. Zheng, X. Shao, and L. Cui, "Automatic sleep stage classification based on convolutional neural network and fine-grained segments," *Complexity*, vol. 2018, 2018.

[29] N. Banluesombatkul, P. Ouppaphan, P. Leelaarporn, P. Lakhan, B. Chaitusaney, N. Jaimchariyatam, E. Chuangsuwanich, W. Chen, H. Phan, N. Dilokthanakul *et al.*, "Metasleeplearner: A pilot study on fast adaptation of bio-signals-based sleep stage classifier to new individual subject using meta-learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 6, pp. 1949–1963, 2020.

[30] P. Ghasemzadeh, H. Kalbkhani, S. Sartipi, and M. G. Shayesteh, "Classification of sleep stages based on lstar model," *Applied Soft Computing*, vol. 75, pp. 523–536, 2019.

[31] M. Perslev, M. Jensen, S. Darkner, P. J. Jennum, and C. Igel, "U-time: A fully convolutional network for time series segmentation applied to sleep staging," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] D. Alvarez-Estevez and R. Rijsman, "Haaglanden medisch centrum sleep staging database (version 1.0.1)," *PhysioNet. https://doi.org/10.13026/7egw-0p30*, 2021.

[33] I. Choi, J. Park, and W. Sung, "Character-level language modeling with gated hierarchical recurrent neural networks." in *INTERSPEECH*, 2018, pp. 411–415.

[34] D. M. Roberts, M. M. Schade, G. M. Mathew, D. Gartenberg, and O. M. Buxton, "Detecting sleep using heart rate and motion data from multisensor consumer-grade wearables, relative to wrist actigraphy and polysomnography," *Sleep*, vol. 43, no. 7, p. zsaa045, 2020.

[35] M. Radha, P. Fonseca, A. Moreau, M. Ross, A. Cerny, P. Anderer, X. Long, and R. M. Aarts, "A deep transfer learning approach for wearable sleep stage classification with photoplethysmography," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–11, 2021.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[40] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[41] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International conference on machine learning*. PMLR, 2018, pp. 4055–4064.

[42] Electrophysiology, Task Force of the European Society of Cardiology the North American Society of Pacing, "Heart rate variability: standards of measurement, physiological interpretation, and clinical use," *Circulation*, vol. 93, no. 5, pp. 1043–1065, 1996.

[43] G. D. Clifford, "Signal processing methods for heart rate variability," Ph.D. dissertation, Oxford University, UK, 2002.

[44] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.

[45] J. Jeppesen, S. Beniczky, P. Johansen, P. Sidenius, and A. Fuglsang-Frederiksen, "Using lorenz plot and cardiac sympathetic index of heart rate variability for detecting seizures for patients with epilepsy," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 4563–4566.

[46] M. M. Rahman, M. I. H. Bhuiyan, and A. R. Hassan, "Sleep stage classification using single-channel eog," *Computers in biology and medicine*, vol. 102, pp. 211–220, 2018.

[47] I. Choi and W. Sung, "Performance assessment of automatic sleep stage classification using only partial psg sensors," in *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2022, pp. 670–674.

[48] M. Radha, P. Fonseca, A. Moreau, M. Ross, A. Cerny, P. Anderer, X. Long, and R. M. Aarts, "Sleep stage classification from heart-rate variability using long short-term memory neural networks," *Scientific reports*, vol. 9, no. 1, p. 14149, 2019.

[49] D. Kiyasseh, T. Zhu, and D. A. Clifton, "Clocs: Contrastive learning of cardiac signals across space, time, and patients," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5606–5615.

[50] I. Feinberg, "Changes in sleep cycle patterns with age," *Journal of psychiatric research*, vol. 10, no. 3-4, pp. 283–306, 1974.

[51] A. Babloyantz, J. Salazar, and C. Nicolis, "Evidence of chaotic dynamics of brain activity during the sleep cycle," *Physics letters A*, vol. 111, no. 3, pp. 152–156, 1985.

[52] A. Crivello, P. Barsocchi, M. Girolami, and F. Palumbo, "The meaning of sleep quality: a survey of available technologies," *IEEE access*, vol. 7, pp. 167 374–167 390, 2019.

[53] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," *arXiv preprint arXiv:1708.02182*, 2017.

[54] K. Hwang and W. Sung, "Character-level incremental speech recognition with recurrent neural networks," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 5335–5339.

[55] T. Nakatani, “Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration,” in *Proc. Interspeech*, 2019.

[56] G.-Q. Zhang, L. Cui, R. Mueller, S. Tao, M. Kim, M. Rueschman, S. Mariani, D. Mobley, and S. Redline, “The national sleep research resource: towards a sleep data commons,” *Journal of the American Medical Informatics Association*, vol. 25, no. 10, pp. 1351–1358, 2018.

[57] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[58] Y. LeCun, Y. Bengio *et al.*, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[59] Y. Bengio, R. Ducharme, and P. Vincent, “A neural probabilistic language model,” *Advances in neural information processing systems*, vol. 13, 2000.

[60] P. F. Brown, V. J. Della Pietra, P. V. Desouza, J. C. Lai, and R. L. Mercer, “Class-based n-gram models of natural language,” *Computational linguistics*, vol. 18, no. 4, pp. 467–480, 1992.

[61] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[62] J. H. Lau, A. Clark, and S. Lappin, “Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge,” *Cognitive science*, vol. 41, no. 5, pp. 1202–1241, 2017.

[63] J. R. Bellegarda, “Statistical language model adaptation: review and perspectives,” *Speech communication*, vol. 42, no. 1, pp. 93–108, 2004.

[64] M. Sundermeyer, R. Schlüter, and H. Ney, “Lstm neural networks for language modeling,” in *Thirteenth annual conference of the international speech communication association*, 2012.

[65] A. Onan and M. A. Toçoğlu, “A term weighted neural language model and stacked bidirectional lstm based framework for sarcasm identification,” *IEEE Access*, vol. 9, pp. 7701–7722, 2021.

[66] E. Shareghi, D. Gerz, I. Vulić, and A. Korhonen, “Show some love to your n-grams: A bit of progress and stronger n-gram language modeling baselines,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4113–4118.

[67] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.

[68] V. Steinbiss, B.-H. Tran, and H. Ney, “Improvements in beam search,” in *Third international conference on spoken language processing*, 1994.

[69] P. S. Ow and T. E. Morton, “Filtered beam search in scheduling,” *The International Journal Of Production Research*, vol. 26, no. 1, pp. 35–62, 1988.

[70] C. Tillmann and H. Ney, “Word reordering and a dynamic programming beam search algorithm for statistical machine translation,” *Computational linguistics*, vol. 29, no. 1, pp. 97–133, 2003.

[71] K. Heafield, “Kenlm: Faster and smaller language model queries,” in *Proceedings of the sixth workshop on statistical machine translation*, 2011, pp. 187–197.

[72] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker, “Perplexity—a measure of the difficulty of speech recognition tasks,” *The Journal of the Acoustical Society of America*, vol. 62, no. S1, pp. S63–S63, 1977.

[73] S. Stern, D. Tzivoni, and Z. Stern, “Diagnostic accuracy of ambulatory ecg monitoring in ischemic heart disease.” *Circulation*, vol. 52, no. 6, pp. 1045–1049, 1975.

[74] S. Khalighi, T. Sousa, J. M. Santos, and U. Nunes, “Isruc-sleep: A comprehensive public dataset for sleep researchers,” *Computer methods and programs in biomedicine*, vol. 124, pp. 180–192, 2016.

[75] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[76] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *ICLR (Poster)*, 2015.

[77] M. K. Uçar, M. R. Bozkurt, C. Bilgin, and K. Polat, “Automatic sleep staging in obstructive sleep apnea patients using photoplethysmography, heart rate variability signal and machine learning techniques,” *Neural Computing and Applications*, vol. 29, pp. 1–16, 2018.

[78] M. M. Ghassemi, B. E. Moody, L.-W. H. Lehman, C. Song, Q. Li, H. Sun, R. G. Mark, M. B. Westover, and G. D. Clifford, “You snooze, you win: the physionet/computing in cardiology challenge 2018,” in *2018 Computing in Cardiology Conference (CinC)*, vol. 45. IEEE, 2018, pp. 1–4.

[79] D. J. Gottlieb and N. M. Punjabi, “Diagnosis and management of obstructive sleep apnea: a review,” *Jama*, vol. 323, no. 14, pp. 1389–1400, 2020.

[80] A. V. Benjafield, N. T. Ayas, P. R. Eastwood, R. Heinzer, M. S. Ip, M. J. Morrell, C. M. Nunez, S. R. Patel, T. Penzel, J.-L. Pépin *et al.*, "Estimation of the global prevalence and burden of obstructive sleep apnoea: a literature-based analysis," *The Lancet Respiratory Medicine*, vol. 7, no. 8, pp. 687–698, 2019.

[81] P. E. Peppard, T. Young, J. H. Barnet, M. Palta, E. W. Hagen, and K. M. Hla, "Increased prevalence of sleep-disordered breathing in adults," *American journal of epidemiology*, vol. 177, no. 9, pp. 1006–1014, 2013.

[82] H. Sümbül, A. H. Yüzer, and K. Şekeroğlu, "A novel portable real-time low-cost sleep apnea monitoring system based on the global system for mobile communications (gsm) network," *Medical & Biological Engineering & Computing*, vol. 60, no. 2, pp. 619–632, 2022.

[83] R. Bhattacharjee, A. Benjafield, A. Blase, G. Dever, J. Celso, J. Nation, R. Good, and A. Malhotra, "The accuracy of a portable sleep monitor to diagnose obstructive sleep apnea in adolescent patients," *Journal of Clinical Sleep Medicine*, vol. 17, no. 7, pp. 1379–1387, 2021.

[84] S.-W. Cho, S. J. Jung, J. H. Shin, T.-B. Won, C.-S. Rhee, and J.-W. Kim, "Evaluating prediction models of sleep apnea from smartphone-recorded sleep breathing sounds," *JAMA Otolaryngology–Head & Neck Surgery*, vol. 148, no. 6, pp. 515–521, 2022.

[85] N. Montazeri Ghahjaverestan, S. Saha, M. Kabir, B. Gavrilovic, K. Zhu, and A. Yadollahi, "Sleep apnea severity based on estimated tidal volume and snoring features from tracheal signals," *Journal of Sleep Research*, vol. 31, no. 2, p. e13490, 2022.

[86] L. Ding, J. Peng, L. Song, and X. Zhang, "Automatically detecting apnea-hypopnea snoring signal based on vgg19+ lstm," *Biomedical Signal Processing and Control*, vol. 80, p. 104351, 2023.

[87] V. L. Le, D. Kim, E. Cho, H. Jang, R. D. Reyes, H. Kim, D. Lee, I.-Y. Yoon, J. Hong, and J.-W. Kim, "Real-time detection of sleep apnea based on breathing sounds and prediction reinforcement using home noises: Algorithm development and validation," *Journal of Medical Internet Research*, vol. 25, p. e44818, 2023.

[88] Y. Wu, X. Pang, G. Zhao, H. Yue, W. Lei, and Y. Wang, "A novel approach to diagnose sleep apnea using enhanced frequency extraction network," *Computer Methods and Programs in Biomedicine*, vol. 206, p. 106119, 2021.

[89] G. Korompili, A. Amfilochiou, L. Kokkalas, S. A. Mitilineos, N.-A. Tatlas, M. Kouvaras, E. Kastanakis, C. Maniou, and S. M. Potirakis, "Psg-audio, a scored polysomnography dataset with simultaneous audio recordings for sleep apnea studies," *Scientific Data*, vol. 8, no. 1, p. 197, 2021.

[90] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition," *Speech communication*, vol. 54, no. 4, pp. 543–565, 2012.

[91] J. Ainslie, S. Ontanon, C. Alberti, V. Cvicek, Z. Fisher, P. Pham, A. Ravula, S. Sanghai, Q. Wang, and L. Yang, "ETC: Encoding long and structured inputs in transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 268–284. [Online]. Available: https://aclanthology.org/2020.emnlp-main.19

[92] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[93] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[94] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[95] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

초 록

아동을 포함한 일반적 대중의 낮은 수면의 질은 잠재적으로 인지 능력, 학습, 기억, 심지어 우울증과 심혈관 질환 등 신체적, 정신적 질환의 원인이 된다. 따라서 편안하고 안정적인 수면은 신체적, 심리적 건강에 필수적인 요소이다. 수면 장애는 개인의 다양한 신체적 특성 및 환경적, 정신적 요인들이 복합적으로 작용하여 발생한다. 따라서 개인적 요인에 따른 불확실성을 극복하고 이용자의 편의를 고려한 수면 단계분류 모델을 만드는 것은 여러 어려움이 따른다.

본 논문은 기존 자동 음성 인식 분야의 원리와 기술을 적용하여 자동 수면 단계 분류기를 심층 신경망 아키텍처로 개발하여 이 분야의 어려움을 극복하고자 한다.

첫째로 편의성을 고려하여 개인의 불편을 초래하고 이용에 복잡한 절차가 필요한 뇌파 센서를 대체하기 위해 수면다원검사의 다양한 신호들의 가능성을 확인하고 여러 심층 신경망 아키텍처들에 적용하여 성능을 비교하였다. 이 결과를 통해 심전도 신호 하나만 사용한 자동 수면 단계 분류 모델을 개발하였다. 심전도 신호는 수면 단계 판정에 필요한 정보가 간접적으로 내재하여 그 정보를 파악하는데 어려움이 따른다. 이를 극복하기 위해 자동 음성인식에서 보편적으로 사용하는 특징 추출기와 시계열 모델을 결합한 구조를 채택하였다. 특징 추출기는 ContextNet 구조를, 시계열 모델은 Transformer 구조를 사용하여 심층 신경망 모델을 구성하였다. 특히 하룻밤 전체의 수면 신호를 한꺼번에 다루어 2, 3시간 주기로 발생하는 수면 단계 유형을 모델이 포착할 수 있도록 하였다.

제한된 수면 신호에서 오는 수면 정보 부족을 보완하기 위해 자연어 처리에서 다루는 언어 모델에서 영감을 얻어, 수면 모델을 제안하였다. 수면 모델은 이전 수면 단계의 시계열 정보를 바탕으로 다음 수면 단계를 예측한다. 따라서 수면 모델은 심

전도 신호를 바탕으로 한 수면 단계 분류 모델의 예측을 보완하여, 수면 단계 예측의 정확성을 높일 수 있었다.

또한, 음성인식과 관련성을 높여, 수면 중 녹음된 오디오 신호로부터 수면 무호흡증을 분류하는 모델을 설계하였다. 이 모델은 수면 오디오에서 네 가지 유형의 수면 무호흡증을 분류하는 것을 목표로 한다. 그중 뇌의 신경 신호 문제로 발생하는 두 가지 수면 무호흡은 소리만으로 구분이 어려워 정확도가 낮지만, 무호흡, 저 호흡, 정상 호흡 구별이 가능한 모델을 개발하였다. 추가적으로 예측 결과를 바탕으로 수면의 질을 가늠하는 지표 중 하나인 무호흡-저호흡 지수(AHI)를 추정하여 수면 다원검사에서 판정한 지수와 비교하였다. 특히 1차 식으로 추정치를 보정하여 판정 지수와의 차이를 감소시켰다.

종합하여 본 논문은 수면과 관련된 두 가지 작업(무호흡 분류와 수면 단계분류)을 수면 오디오와 심전도 신호를 통해 수행할 수 있는 심층 신경망 모델을 제안하였다. 이러한 모델은 개인이 집에서 쉽게 사용하면서 본인의 수면문제나 무호흡에 관한 정보를 얻을 수 있도록 돕기 때문에 일반 개인의 수면 건강 개선에 이바지할 수 있다. 우리는 특히 제한적이고 부족한 정보를 제공하지만, 사용의 편의성과 접근성이 좋은 신호에 초점을 맞춘 심층 신경망 모델을 개발하였기 때문에 모바일 기기 적용에 높은 가능성을 가진다.

주요어: 수면다원검사, 수면 단계, 수면 무호흡증, 시계열 모델

학번: 2014-21751

ACKNOWLEGEMENT

긴 시간이 걸린 학업을 마치며 제 성장에 도움을 주신 많은 분을 되새기며 글을 남깁니다. 대학교에 들어서면서 부 터 저는 점점 작아지기만 했습니다. 뚜렷한 목표도, 스스로에 대한 확신도 없었기에 성실하게 성취를 이루어나가지 못했습니다. 대학원에 진학하게 된 이유도 학업에 대한 흥미보다는 '아직 모르겠다'라는 생각이 더 컸던 것 같습니다. 이런 마음가짐으로 임한 대학원이었기에 제자리에 멈춰있던 시간이 길었던 저를 이끌어 주고 지원해 준 모든 분께 감사의 인사를 전합니다.

인연 없이 불쑥 부탁드린 학위 심사위원 초청에 응해주신 조남익 교수님, 정교민 교수님. 짧은 인연에도 심사에 참여해 주신 최정욱 교수님. 잠시나마 제 지도교수를 맡아주시고 물심양면으로 지원해 주신 심병효 교수님께 감사드립니다. 발전이 보이지 않았던 저를 포기하지 않고 앞으로 나갈 힘과 방향을 보여 주신 성원용 교수님. 정신적 어려움에서 직접 끌어올려 주셔서 말과 글로는 표현 못 할 깊은 감사와 존경을 표합니다.

연구실 생활의 시작을 함께 해준 준희 형, 민제 형, 창헌이 형, 동윤이 형 규연이. 덕분에 큰 어려움 없이 연구실 생활을 시작할 수 있었습니다. 친절히 복잡한 업무를 처리해 주신 이미순 비서님. 짧게나마 함께했던 승열이, 성욱이, 윤진이, 루카스, 그리고 영민이 가 있어 연구실의 변화와 동력을 얻게 되었던 것 같습니다. 연구실에서 가장 긴 시간을 함께했던 윤호와 진환이. 마지막까지 함께 했던 석현이와 많은 도움을 준 규홍이에게도 감사의 마음을 전합니다. 마지막으로 답답해 보였을 저를 가까이에서 도와주고 걱정해 준 성호에게 미안함과 함께 고맙다는 말을 전합니다.

제 학업의 마무리에 있어 새로운 식구로 받아주신 김성우 교수님과 ARIL 연구실 모든 분. 새로운 시각을 열어주고 마지막을 외롭지 않게 옆에 있어 줘서 큰 힘이 되었습니다. 대학 생활에 대한 즐거운 추억을 만들어준 R반 동기들에게 감사함과 동시에 먼저 자리를 피해 얼굴 보기 어려워진 것에 대한 미안함을 전합니다. 학업을

진행하며 얻게 된 수많은 인연을 끝까지 잊지 않겠다는 약속드리며 모두의 건강과 행복을 기원합니다.

마지막으로 정말 긴 시간이 걸린 졸업을 기다려준, 부담을 주지 않으려 걱정을 속으로 인내해준 부모님과 형에게 부족하게나마 제 마음과 이 결과물을 헌정합니다. 항상 여러분의 지원을 기억하며 여러분과 같은 더 큰 사람이 되도록 노력하겠습니다. 감사합니다.

2023년 8월

최 익 수