UNIVERSITY OF CALIFORNIA,
IRVINE

Mobile Health Monitoring on Low-power Wearable Devices based on EEG and ECG Signals

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Computer Engineering

by

Berken Utku Demirel

Thesis Committee:
Associate Professor Mohammad Al Faruque, Chair
Assistant Professor Peter Tseng
Assistant Professor Salma Elmalaki

2022

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my committee chair, Professor Mohammad Abdullah Al Faruque, for constantly guiding me and believing in me. Without his support, I would not have completed this thesis.

I would also like to thank my committee members Professor Peter Tseng and Professor Salma Elmalaki, for providing me guidance and feedback.

I would also like to thank my lab-mates Nafiul Rashid, Mohanad Odema and Luke Chen for their intellectual support with my research.

ABSTRACT OF THE THESIS

Mobile Health Monitoring on Low-power Wearable Devices based on EEG and ECG Signals

By

Berken Utku Demirel

Master of Science in Computer Engineering

University of California, Irvine, 2022

Associate Professor Mohammad Al Faruque, Chair

The recent developments in signal processing, machine learning, and smart sensing technology allow real-time monitoring and recording of bio-signals, especially the electrocardiogram (ECG) and electroencephalogram (EEG) signals. However, continuous monitoring of those signals is challenging in low-power systems such as wearable devices due to energy and memory constraints. Therefore, in this work, I proposed a novel lightweight single-channel EEG-based method to estimate arousal levels, defined as an individual's degree of alertness or responsiveness to stimuli at low-power wearable devices for brain activity monitoring. Moreover, a novel resource-efficient template control based Convolutional Neural Network (CNN) architecture is presented for cardiac activity monitoring. To evaluate the proposed methodology performance for brain activity monitoring, I used scalp EEG recorded during overnight sleep and intra-operative anesthesia with technician-scored hypnogram annotations at the University of California at Irvine Medical Center. And, the well-known PTB diagnostic ECG database is used to assess the proposed methodology's performance for cardiac activity monitoring. Evaluation of real hardware shows that the proposed methodologies can be implemented for devices with a minimum RAM of 512 KB while maintaining high accuracy with low energy consumption compared to the state-of-the-art works.

Chapter 1

Introduction

Recent advances in machine learning with signal processing have opened the gate to unveil fundamental processes in the human brain and heart, such as a change in drowsiness level or the effect of the heart rate variability in cardiovascular diseases. These research efforts have become possible by recording and stimulating the human brain and heart activity in the clinical setup by clinicians with very high accuracy. Clinical ECG and EEG is the primary tool for monitoring a person's health situation. However, they can only be used for a limited time, and continuous monitoring of the patients' condition is still required outside clinical hours [6, 33]. Although some techniques were developed to monitor patients' situations continuously, they have failed to provide enough caring systems. For example, ambulatory ECG devices are traditionally used to monitor cardiac activity for a long duration to be further investigated by clinicians. For example, the Holter [14], a battery-operated portable device, is used to record and store long-term ECG signals. However, these devices cannot provide real-time feedback to users, and cardiologists need to analyze long-term recordings, which is a very time-consuming and expensive process.

These difficulties are also the same for monitoring neural activity even though EEG signals are widely used for emotion recognition, epileptic seizures, and drowsiness. The nature of

EEG signals makes it harder to process data near the sensor as it requires more channels with a high sampling rate to get information about the different neural activities. Currently, wearable device solutions for monitoring those two vital signals follow a cloud computing architecture where the raw data from wearable devices is offloaded to fog (mobile phones) or cloud (remote servers), where the processing takes place [31], using the recent developments in IoT [8]. This offloading consumes a tremendous amount of communication energy which reduces the battery life of wearable devices, as well as mobile phones [21, 23]. In addition to that, it also introduces latency which hinders the real-time monitoring and detection in healthcare applications [37]. Moreover, offloading the raw data to a mobile phone or cloud makes the users' data vulnerable to privacy breaches [10]. To overcome these limitations, 'Edge Computing' [19] architecture gained momentum in recent years where all the processing is done on the wearable devices [32, 34], and only the analyzed results are sent to the cloud for remote monitoring. Thus, it addresses the issues as mentioned earlier related to energy consumption, latency, and vulnerability of privacy breaches.

The designed signal processing and machine learning algorithms for wearable devices should be energy-efficient, memory-efficient, and provide acceptable performance within the previous two constraints. Most of the state-of-the-art works on monitoring these two signals are not wearable device compatible as they prioritize performance and do not consider the other two constraints. The state-of-the-art works can be roughly separated into two groups. The first group tried to extract some handcrafted features, which is also called feature engineering from the bio-signals manually [15, 7, 30]. For neural activity monitoring examples, authors in [15] proposed to transform time-domain signals to frequency domain and then extract 181 distinct frequency-domain features for classifying a person's arousal while watching emotional content videos. Similarly, in [22], authors extract 104 features from different frequency bands of electroencephalogram (EEG) to classify sleep stages. Also, these featre extraction based algorithms used more than one EEG channel to extract features. However, as the number of extracted features and channels increases, the algorithms' computa-

tional complexity and memory requirement increase, making them unsuitable for wearable devices with constrained memory and energy resources. The second group used complex deep learning algorithms [1, 29], such as CNN or LSTM to extract features automatically. Machine learning algorithms perform classification based on the extracted features from the data. However, the feature extraction processes are often time-consuming and require a huge amount of energy. Deep learning algorithms like Convolutional Neural Networks (CNN) do not require manual feature engineering and extraction as they automatically extract features through convolution layer. Moreover, the layered architecture of CNN provides flexibility to design a network by adding or removing layers as necessary in the training phase. Later, this trained architecture may be used to classify data during the inference phase. However, the full architecture from the training phase may not be needed at the inference phase [24], as many of the data can be correctly classified using only the first few layers of the architecture. This early exit capability of CNN should help to avoid redundant operations during the inference phase leading to energy efficiency for wearable device solutions while maintaining the performance. The early exit architectures usually make the exit decision based on the classification confidence at a particular exit (output) layer. If the classification confidence at one of the exit layers is greater than a specific threshold, it exits the network and thereby avoids further operations. Therefore, using an early exit CNN architecture where the network portions can be used at the inference phase is more appropriate for wearable device solutions than the entire architecture.

This work shows that both feature extraction and designing neural networks can be implemented in resource-constrained (low-power, low-memory) systems if the application-specific knowledge is also involved in designing the overall system. To show the resource-efficient feature extraction, the arousal level estimation using EEG signals is chosen as an application. It is shown that the proposed methodology discriminates wakefulness from reduced arousal solely based on the neurophysiological brain state with more than 80% accuracy. Therefore, our findings describe a common electrophysiological marker (feature) that tracks

reduced arousal states, which can be applied to different applications (e.g., emotion detection, driver drowsiness). Evaluation on hardware shows that the proposed methodology can be implemented for devices with a minimum RAM of 512 KB with 55 mJ average energy consumption. For showing the efficient processing of neural networks, the Myocardial Infarction (MI), also known as heart attack, detection using ECG signals is selected as another application with different bio-signal. The baseline and proposed neural network architecture outperform related works on the well-known PTB diagnostic ECG database. Both of our architectures are suitable for wearable devices requiring only 20 KB of RAM. Evaluation of real hardware shows that our methodology is faster and up to 53×more energy-efficient than the state-of-the-art works on wearable devices.

1.1 Background

1.1.1 The Electrocardiogram

The Electrocardiogram (ECG) signals are the electrical activity of the heart cells. In the heart, there are special cells that are adequate to generate their electrical impulse intrinsically. In the cardiac conduction system, the electrical stimulation is triggered by the cells with the fastest inherent firing rate. The following parts of the heart play a role in the cardiac conduction system, the sinoatrial (SA) node, the atrioventricular node, the atrioventricular bundle branches and the Purkinje cells. Cardiac rhythm is initialized by the sinoatrial (SA) node. The SA node is called the pacemaker of the heart. Electrical signals initialized from the SA node is sent to the atrioventricular node through internal pathways. The electrical signal's travel time between the atrioventricular node and the SA node is approximately 50 milliseconds [4]. During this time, the atria pump the blood into the ventricles. The electrical signal travels to the atrioventricular bundle branches and the Purkinje fibres after passing

His bundle. With the help of Purkinje fibres, the ventricles receive the electrical signal and contract. As the ventricles contract, the blood goes to the lungs from the right ventricle. Meanwhile, the left ventricle pumps the blood to the aorta, and the blood is distributed to the rest of the body.

1.1.2 Measurement of ECG

In clinical setups, the electrical activity of the heart is measured by using the standard 12-lead electrocardiography noninvasively. The output of the standard 12-lead electrocardiography is called an electrocardiogram. As seen in Figure 1.1, there are three main parts in an electrocardiogram: Depolarization in the atria is reported as a P wave. Atrial anomalies are detected by observing the P wave's duration, amplitude, and frequency. The interval between the beginning of the upslope of the P wave and the beginning of the QRS complex is called the PR interval. The duration of the PR interval is approximately between 120 and 200 ms. The QRS complex, one of the most important parts of the ECG waveform, reflects the repolarization of atria and the depolarization of ventricles. The QRS complex comprises Q wave, R wave and S wave. The first negative deflection in the QRS complex is called the Q wave, the first positive deflection is called the R wave, and the second negative deflection is called the S wave. The QRS complex has a sharper form than the P wave due to the fast conduction velocity in the bundle of His and the Purkinje cells. The change in the duration of the QRS complex may indicate arrhythmias, ventricular 9 hypertrophy, or myocardial infarction.

12-lead ECG is the primary tool for monitoring cardiac activity; however, the number of channels with bulk electrodes makes it harder to implement in a mobile environment. Therefore, single-channel ECG systems are recently used to detect myocardial infarction and several arrhythmias. In this work, it is also shown that the myocardial infarction, which is also called a heart attack, can be successfully detected using single-lead ECG.

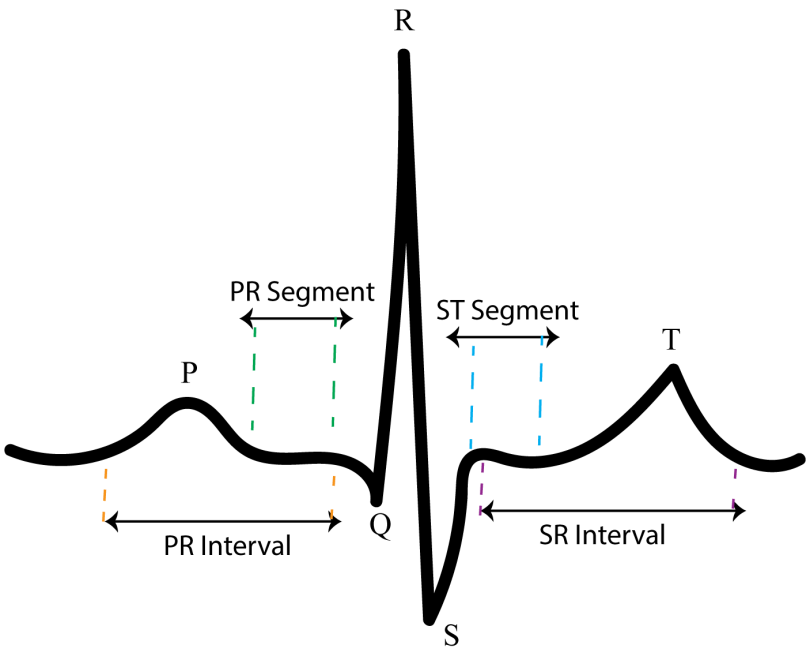

Figure 1.1: A complete heartbeat

1.1.3 The Electroencephalography

Electroencephalography (EEG) is a method for measuring electrical activity of a brain (neural activity) by placing an array of electrodes on the surface of a subject's scalp. EEG signals have been widely used in several clinical setups from its first recording by Hans Berger in 1924 [39]. For example, EEG is used as a diagnostic tool for epilepsy, brain damage and sleep disorders and is commonly used in various research areas, such as emotion recognition [16], sleep stage classification [26] or brain-computer interfaces.

Although, recently, classification and detection of different applications using EEG signals draw significant attention from academia and industry, robust and real-time analysis of EEG are challenging to achieve for many reasons. Firstly, EEG measurements are susceptible and tend to have a low signal to noise ratio. The electrical activity observed on the scalp's surface is measured on the microvolt level, requiring excessively sensitive differential amplifiers and low impedances between the electrodes and the scalp. Due to the small electrical potentials involved, EEG is also very susceptible to contamination by various types of artefacts and

noise. For example, different bio-signals from the human body, such as heart contraction or muscle contraction and eye movements, can generate artefacts that dominate the signals generated by brain activity. Moreover, the small movements of a subject can generate high voltage spikes in the EEG signal which makes it harder to collect and process data. Also, EEG suffers from low spatial resolution, as the electric fields generated by the brain are smeared by the tissues, such as the skull, situated between the sources and the sensors.
Although EEG signals have some major drawbacks, there are some advantages of using EEG signals as well. For example, EEG has an excellent temporal resolution thanks to the incredible speed at which electric fields propagate: events occurring at millisecond timescales in the brain can typically be captured using EEG signals. As menrtioned before, the spatial resolution of the EEG signals are low due to smearing by the skulls and tissues. However, as a result of that EEG channels are often highly correlated spatially, which brings additional problem types such as the source localization, or inverse problem. These topics are an active area of research in which algorithms are developed to reconstruct brain sources given EEG recordings.

1.1.4 Measurement of EEG

In clinical setups, the electrical activity of the brain is measured by using the standard 10-20 configuration noninvasively. The output of this setup is called an Electroencephalography (EEG). Unlike the ECG signals, the EEG signal has no known pattern or determined points (P, Q, R, S points). Therefore, different applications set a window length where the EEG signals are assumed to be stationary. For example, in sleep stage classification, the EEG signals are segmented as 30-second windows for processing. 10-20 setup is the primary tool for monitoring neural activity in clinical setups; however, this monitoring system suffers from preparation time (conductive gels or pastes) and large equipment (metal electrodes) and prevents the EEG monitoring to be deployed in mobile devices. However, the advances in

sensor technology allow a fully portable, wireless, long-term, flexible scalp electronic system, incorporating a set of dry electrodes [20].

Thanks to these advancements in sensing technologies and, in particular, tiny electrical charges, the EEG signals can be measured using portable commodity devices. These sensors track the brain's electrical activity by placing electrodes on the scalp. These electrodes detect the tiny electrical changes on the skin that originate from the neurons' electrophysiologic pattern of depolarization and repolarization. Therefore, single-channel EEG systems are recently used to detect and classify different EEG signals. In this work, it is also shown that the arousal level, which is defined as the level of consciousness, can be successfully estimated using a single EEG channel.

Chapter 2

Materials and Methods for Arousal Level Estimation

This section explains the data collection for validating the algorithms and methods applied to collected EEG signals for estimating the arousal level at low-power wearable devices using single-channel of EEG. In the end, the estimation results are presented with energy and memory calculations of each operation.

2.1 Materials

2.1.1 Data Collection

The scalp EEG was recorded during overnight sleep or intra-operative anesthesia at the University of California at Irvine Medical Center. All patients provided informed consent according to the local ethics committees of the University of California at Berkeley and at Irvine and gave their written consent before data collection. We analyzed recordings from

8 participants between 24 and 57 years old. The Polysomnography which is a comprehensive test used to diagnose sleep disorders, was recorded for the sleep stages during 8 hours and 5 min quiescent rest with eyes closed before and after sleep. Data were recorded on a Grass Technologies Comet XL system (Astro-Med, Inc, West Warwick, RI) with a 19-channel EEG using the standard 10–20 setup as well as three electromyography (EMG) and four electrooculography (EOG) electrodes are used as additional bio-signals to facilitate gold standard sleep staging. The EEG was referenced to the bilateral linked mastoids and digitized at 1000 Hz. Sleep staging was carried out by trained personnel and according to established guidelines.

The anesthesia EEG data were recorded from the induction of anesthesia to the recovery using a Nihon Kohden recording system (256 channel amplifier), analog filtered above 0.01 Hz, and digitally sampled at 5 kHz. General anesthesia was induced intravenous with remifentanil (100 μg) and propofol (150 mg). The awake state, which is also called the highest level of consciousness, was defined as the time before the administration of propofol, and anesthesia was defined as the time after inducing remifentanil and propofol.

2.1.2 Methods

Pre-Processing

Both sleep and anesthesia data are resampled to 200 Hz from 1000 and 5000 Hz for sleep and anesthesia respectively, using a Finite Impulse Response (FIR) antialiasing lowpass filter. Then, a 10 order low-pass Butterworth filter is used with 50 Hz cut-off frequency for denoising. After filtering, sleep data is epoched into 30-second segments. In contrast, the anesthesia is segmented as 10-second to increase the number of epochs as anesthesia's duration (1-3 hours or less) is shorter than sleep (6-10 hours). It is observed that the feature's discrimination performance is best amongst the electrodes Fz, Pz, and Cz. For this study,

we choose the Cz electrode for calculation of spectral slope and classification.

Feature Extraction

After artifact removal and segmentation, the Multitaper approach based on discrete prolate Slepian sequences is used to compute the power spectral density (PSD) estimate from 0.5 Hz to 45 Hz with 0.5 Hz smoothing. Short-time Fourier Transform or Periodogram, which utilizes the discrete Fourier transform to estimate the power spectral density (PSD) of a signal, has been widely used in literature [28, 27] to extract more features from EEG signals. However, these methods suffer from high variance and bias due to the finite duration of the observed non-stationary random signal, ending with an inconsistent estimate of power spectral density. The single taper methods are developed to reduce the bias and variance using windowing techniques such as Hanning and Welch [40], but the resulting spectrograms are still high variance due to used windows are not optimal for increasing the resolution of PSD. We found that Thomson's multitaper spectral estimation method [38] gives the best result for EEG signals by reducing the variance and bias of the spectrum by averaging the estimates from multiple tapers applied to the same window.

$$\lambda = \frac{\int_{-W}^{W} \left|X(f)\right|^2 df}{\int_{-F_s/2}^{F_s/2} \left|X(f)\right|^2 df} \qquad (2.1) \qquad \int_{-F_s/2}^{F_s/2} \left|X(f)\right|^2 df < \infty \qquad (2.2)$$

These tapers, known as Discrete Prolepian Spheroidal Sequences (DPSS), are very useful as they are not only designed to reduce bias, but they also have orthogonality property, which enables them to extract uncorrelated single-taper spectral estimates from the same data. The discrete prolate Slepian sequences (DPSS) arise from the following spectral concentration problem. The discrete Fourier transform (DTFT) ($X(f)$) of a finite time series $x[n]$, for which a sequence maximizes the ratio given in Equation 2.1, subject to the constraint that the sequence has finite power (Equation 2.2).

Where F_s is the sampling rate of the sequence $x[n]$ and $\left|W\right| < F_s/2$. This ratio determines an index-limited sequence with the largest proportion of its energy in the band $[–W, W]$. This maximization leads to the eigenvalue problem is given in Equation 2.3.

$$\sum_{m=0}^{N-1} \frac{sin(2\pi W(n-m))}{\pi(n-m)} g_k(m) = \lambda_k(N,W) g_k(n) \tag{2.3}$$

Where λ_k is the eigenvalues, and $g_k(n)$ is the DPSS values that correspond to kth Slepian sequence. The eigenvectors of this equation, $g_k(n)$, are the DPSS values, which are mutually orthogonal to each other. Since the resulting single-taper estimates are uncorrelated with each other, they can be averaged together as if they were independent trials of the same distribution, producing a spectrum with reduced variance. The estimated power spectra, $S_k(f)$, is calculated at frequency f as in Equation 2.4.

We have used 29 tapers for 30-second segments of sleep EEG and 9 tapers for 10-second anesthesia segments, so the first 29 and 9 DPSS are used for multitaper PSD estimation. After obtaining DPSS values, the modified periodograms are calculated in Equation 2.4 using a different Slepian sequence for each window.

$$S_k(f) = \Delta t \left| \sum_{n=0}^{N-1} g_k(n) x(n) e^{-j2\pi f n \Delta t} \right|^2 \tag{2.4}$$

Here $S_k(f)$ is the modified periodograms, each obtained using a different Slepian sequence $(g_k(n))$. Finally, the multitaper PSD estimate is calculated, by averaging the modified periodograms using Equation 2.5.

$$S(f) = \frac{1}{K} \sum_{k=0}^{K-1} S_k(f) \tag{2.5}$$

After obtaining the power spectral density estimation, we calculated the spectral slope by fitting a linear regression line to the PSD in log-log space between 30 and 45 Hz. This range was proved to correlate best with arousal changes in rodents and monkeys [11]. The best line is obtained using the polynomial curve fitting method to the 30-45 Hz range of multitapered power spectral density estimation. Figure 2.1 shows the PSD estimation of three different EEG epochs obtained using curve fitting.

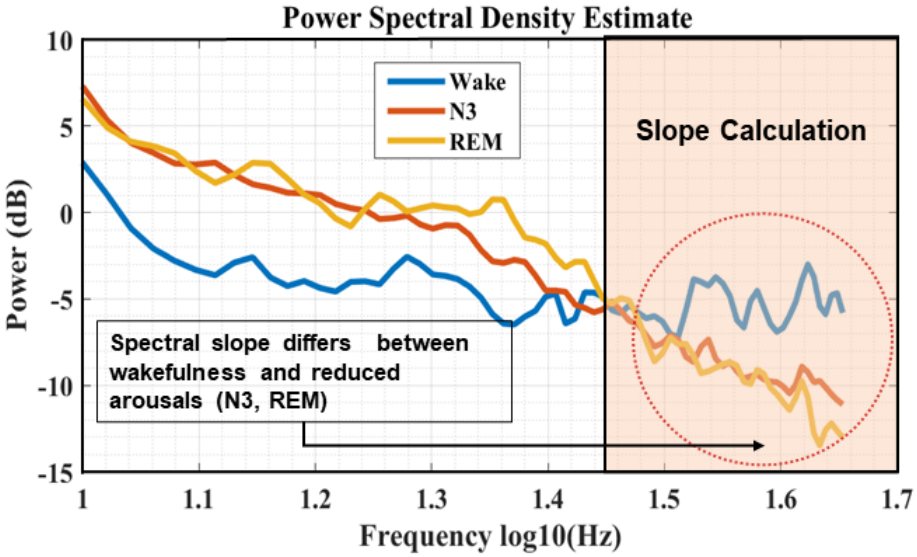

Figure 2.1: The power spectral density of three different EEG epochs

As shown in this figure, the spectral slope of the wakefulness and reduced arousals is different. The slope for the reduced arousals tends to be more negative than the wake stage. To observe these spectral slope differences better between sleep stages, the whiskey plot is shown in Figure 2.2.

Figure 2.2 shows the spectral slopes' distribution for three different sleep stages (Wake,

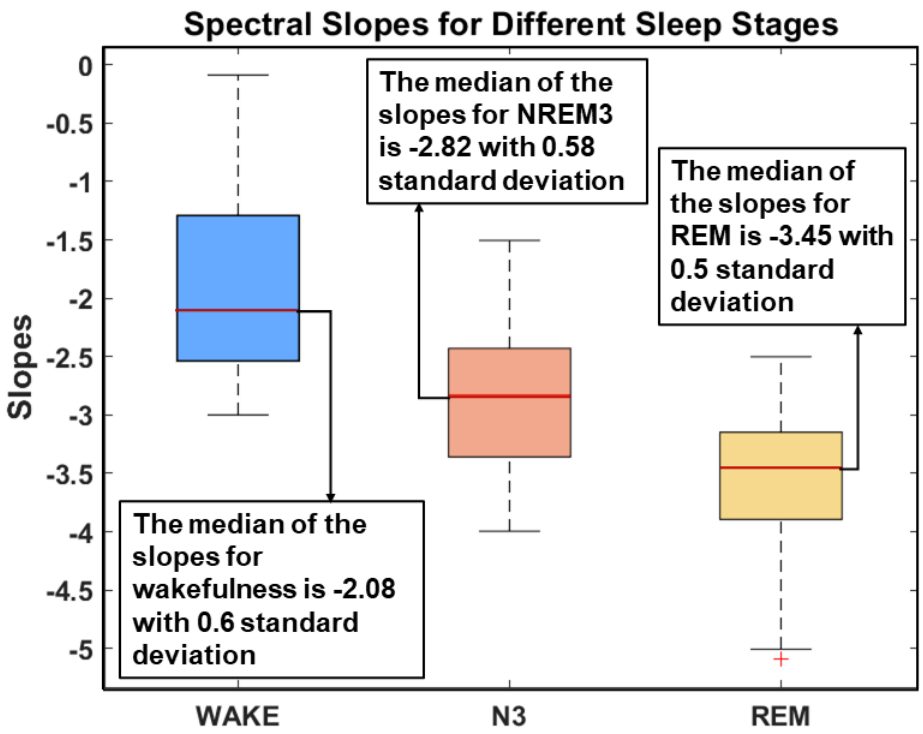

Figure 2.2: The whisker plot for slope distribution of three different stage

NREM3, and REM). We observed the wake stage slope between -0.1 and -3 with a -2.08 median value and 0.6 standard deviations. On the other hand, the median of spectral slopes for the REM stage is -3.45 with 0.5 standard deviations.

We have applied the same method to EEG data recorded under propofol anesthesia to show that the spectral slope between 30-45 Hz can track the reduced arousal levels. In these recordings, the wake stage is defined as the time before the start of propofol and remifentanil, and anesthesia was defined as the periods when the patients were unresponsive to verbal commands assessed by the attending anesthetist. Figure 2.3 shows the spectral slopes obtained during wakefulness and anesthesia. We observed that the spectral slope was higher during wakefulness (-1.9 median value with 0.5 standard deviations) than during anesthesia (-2.77 median value with 0.8 standard deviations).

These results provide evidence that the spectral slope discriminates wakefulness from reduced arousals. Moreover, it is observed that the more negative slopes of the wake stage are closer

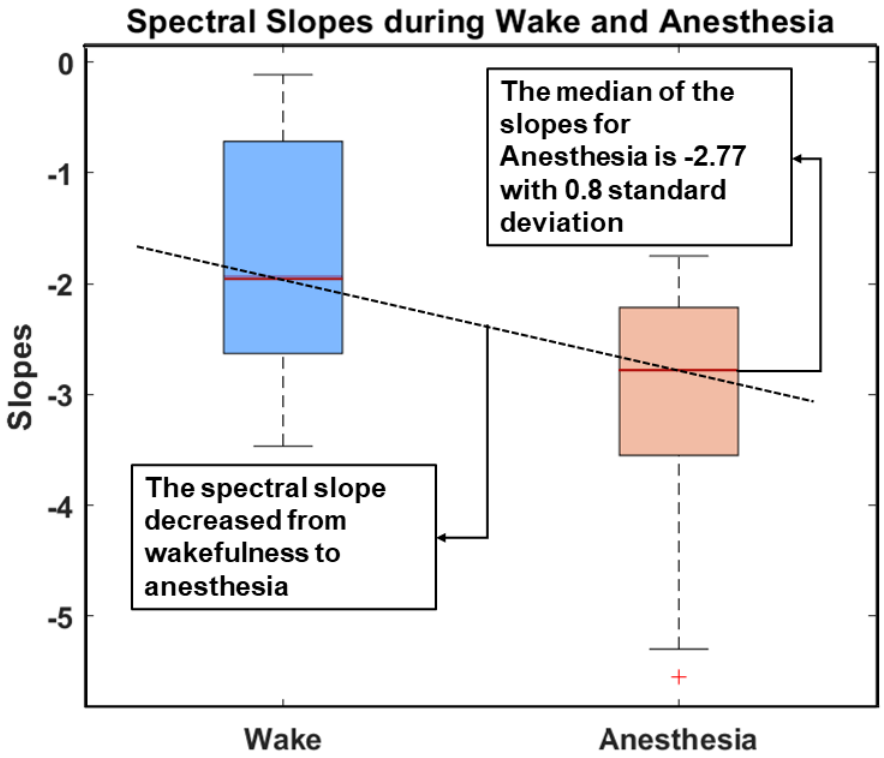

Figure 2.3: The whisker plot for slope distribution of wake and anesthesia

to state transitions, which can be explained as the subjects are still drowsy.

2.1.3 Results

2.1.4 Performance Evaluation

To evaluate the proposed methodology's performance, we used sleep recording, specified in Section 2.1.1, which has an associated hypnogram file scored by a specialist. The hypnogram files contain labels identifying the sleep stages. The two different NREM stages (3 and 4) are combined into one group as NREM3 since these two stages are considered as deep sleep, whereas NREM1 and NREM2 are considered as light sleep [12]. Thus, the classification is performed for three different sleep stages (Wake, NREM3, and REM).

Since we extracted spectral slope from one channel of the EEG signal, simple threshold values

are used for discriminating sleep stages from each other instead of using machine learning algorithms. For this study, the threshold value is chosen as -2.45 for separating wakefulness from different sleep stages. If the calculated spectral slope of an EEG epoch is greater than -2.45, this epoch is classified as *Wake*. If the slope is less than -3.2, the period is classified as *REM*. Lastly, if the calculated spectral slope is between these two threshold values, the epoch is classified as the *NREM3* stage.

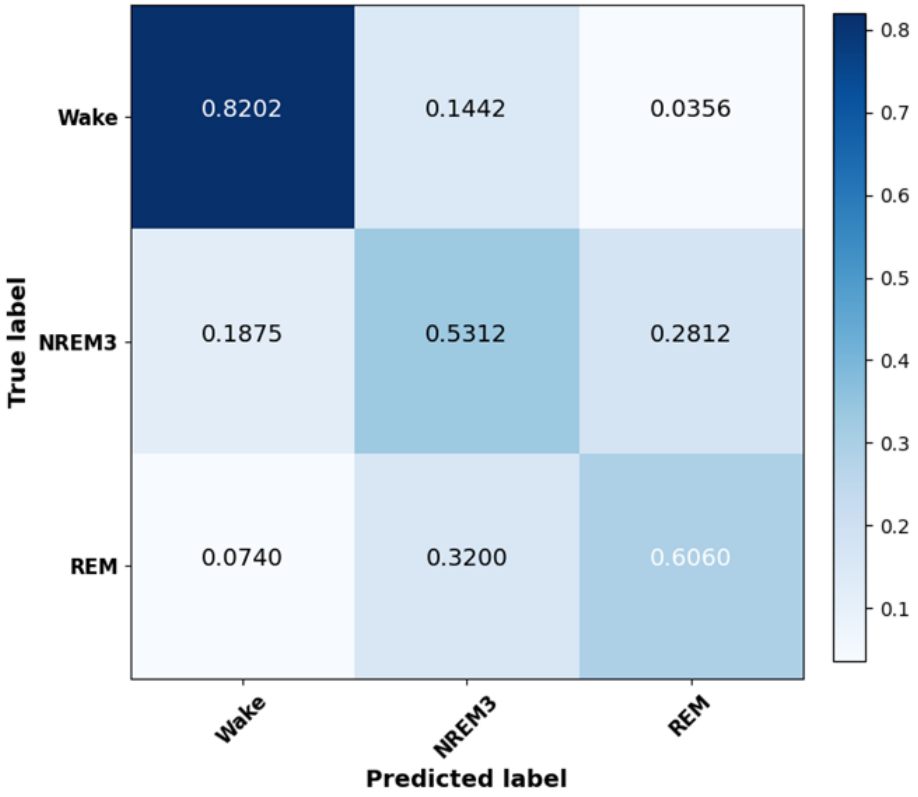

Figure 2.4: The normalized confusion matrix

Figure 2.4 shows the confusion matrix for three different sleep stages. As can be seen from the figure, the confusion of *Wake* stage with *NREM3* and *REM* stages are lower than 15%, which indicates that the spectral slope feature can discriminate the wakefulness with reduced arousal brain states. Since both *NREM3* and *REM* stages are considered as low arousal level, the algorithm's discrimination performance for these two stages is not as high compared to *Wake* classification. However, the confusion degree for reduced arousal with wakefulness is as low as 7% (Wake-REM). As seen in these results, the spectral slope

calculated from the power spectral density can estimate the human's arousal level based on neurophysiological brain state. Also, it is known that if the PSD estimates are obtained from different electrodes with high sampling frequency and averaged across all electrodes, the discrimination performance of the spectral slope increases for arousal level estimation [18] or different methods and techniques can be applied to multitaper power spectra to obtain a more accurate and fine estimation of frequency variations. However, the algorithm's memory requirement and energy consumption increase with the sampling rate and number of channels used, making it unsuitable for wearable devices. Also, most of the wearable system try to use as few channels as possible to increase the comfortability and acceptability amongst society [9].

2.1.5 Memory and Energy Consumption Evaluation

The proposed methodology's memory footprint, energy consumption and execution time are evaluated using an STM32 Nucleo-144 (STM32H743Z), 32-bit ARM Cortex-M7 core with 480 MHz maximum operating frequency, and 1 MB RAM. The energy profiling is performed using STM32 Power Shield (LPM01A), an accurate power monitoring device with a power consumption measurement range between 180 nW and 165 mW. Table 2.1 shows the execution time, energy consumption, and required memory for each operation. The energy and memory calculation is performed using a 30-second epoch of EEG signal, sampled at 200 Hz.

Table 2.1: Memory and Energy Consumption on Nucleo Board

Operations	Exe. Time (ms)	Avg. Power (mW)	Avg Energy (mJ)	Flash Memory Footprint (kB)	RAM Memory Footprint (kB)
Filtering	180	55	10	26.4	96.8
PSD Estimation	101	440	45	45	350
Overall	281	195.7	55	$\geq$ 64 KB	$\geq$ 512 KB

The overall execution time for a 30-second epoch takes 281 ms in the device with 55 mJ average energy consumption. The proposed methodology is also compatible with any devices with a minimum RAM of 512 KB. When multitaper PSD estimation is calculated, the discrete Prolepian sequences are represented as sparse matrices to save memory. Since the first sequences contain a few non-zero elements, storing them as sparse matrices are more efficient. The number of discrete Slepian sequences (29 tapers in this study) used for PSD calculation can be decreased using fewer tapers to save more memory. However, the value for frequency smoothing needs to be decreased, which would reduce PSD resolution and result in a high-variance spectrum.

Chapter 3

Materials and Methods for Real-time MI Detection

This section explains the used public data for validating the proposed algorithm and methods applied to ECG signals for detecting the myocardial infarction at low-power wearable devices. In the end, the results are presented with energy and memory calculations of each operation similar to the previous sections.

3.1 Experimental Setup

3.1.1 Database Used

To make a fair comparison with the state-of-the-art works, the well-known PTB diagnostic ECG database [5] from PhysioNet [13] is used. This database contains MI data from 148 subjects and normal healthy control data from 52 subjects. Each record includes 15 simultaneously measured signals: the conventional 12 leads (I, II, III, aVR, aVL, aVF, V1, V2, V3,

V4, V5, V6) together with the 3 Frank lead ECGs (Vx, Vy, Vz). As the proposed method is designed for the wearable device or mobile health monitoring we use the single lead ECG data from the 11^{th} lead (V5). Each signal segment is digitized at 1000 samples per second (1 kHz). Table 3.3 shows the summary of data distribution and the specific lead used in each of the related works.

3.1.2 Performance Metric

As the number of segments (heart beats) for different classes (MI and Normal) in the dataset is highly imbalanced, reporting only the classification accuracy is not appropriate and enough to measure classification performance of the proposed method. Thus, we use additional both the sensitivity and specificity metric. Also, by that way the proposed method can be compared with other related works in a fair way as shown below:

$$Se = \frac{TP}{TP + FN} \qquad (3.1) \qquad\qquad Sp = \frac{TN}{TN + FP} \qquad (3.2)$$

$$Ac = \frac{TP + TN}{TP + TN + FN + FP} \quad (3.3)$$

Where TP, TN, FP and FN refer to True Positive, True Negative, False Positive and False Negative, respectively. The respective definitions of these four standard metrics using true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are as follows: Accuracy is the ratio of the number of correctly classified beats to the total number of beats classified, Acc = (TP+TN)/(TP+TN+FP+FN); Sensitivity is the rate of correctly classified events among all events, Sen = TP/(TP+FN); Specificity is the rate of correctly classified nonevents among all nonevents, Spe = TN/(TN+FP). In other words, sensitivity is the true positive rate that measures the portion of the positive class (MI segments) that is classified correctly. Likewise, specificity metric is the true negative rate that measures the

portion of the negative class (Normal segments) that is classified correctly.

3.1.3 Training CNN Classifier

To validate the performance of proposed CNN classifier, we use data from 200 subjects (52 Normal, 148 MI) from PTBDB [5]. The total of 62306 (51880 MI, 10426 Normal) heartbeat segments are obtained after pre-processing steps from Lead 11 ECG data. As mentioned before, the dataset is highly imbalanced since the number of abnormal MI segments is approximately 5 times the normal segments. For providing a proper training on the highly imbalanced dataset, the class weights are utilized to each class during training using the following formula $d_k = \frac{1}{N_k} * \frac{N}{n_c}$. In this equation, d_k, and N_k represent the class weight and the number of segments belonging to class k, respectively. N is the total number of segments from all classes and n_c is the number of output classes which are MI and normal heartbeats in this case.

The classifier is trained with a batch size of 500. The models are trained for 150 epochs and select the best model is chosen considering the minimum validation loss. We use *Binary Crossentropy* as the loss function for each output block. *Adam* optimizer is used to train the models with a learning rate of .001. Similar to the state-of-the-art works, the 5-fold cross-validation is performed on the dataset where 80% data (4 folds) is used for training and 20% is used for testing. Also, the training data is further splitted by 20% for validation during training.

3.2 Proposed Method

The following sections explain the details of the proposed methodology. The overview of our proposed methodology is demonstrated in Figure 3.1.

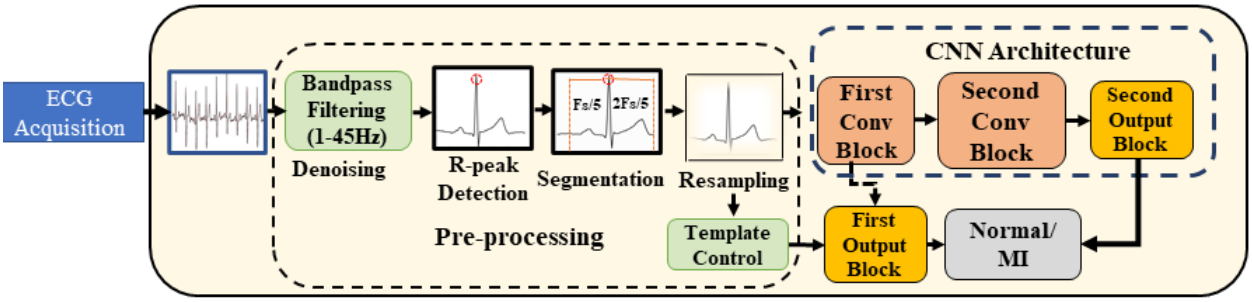

Figure 3.1: Overview of Our Proposed Methodology

3.2.1 Pre-processing Steps

ECG Lead Selection This study only used the modified 11^{th} lead (v5) from PTB dataset [5]. Automatic cardiac activity classification based on ECG is beneficial for portable or wearable devices, and it is known that few channel numbers (even single-channel) would be found in these devices. Hence, we developed our algorithm to run with a single channel of ECGs.

Filtering After selecting the lead, the raw ECG signals are filtered using a tenth-order Butterworth bandpass filter with cut-off frequencies f_1=1Hz and f_2=45Hz. It is observed that since the butterworth filter is designed to have a frequency response that is as flat as possible in the passband range, the morphological waveform of the filtered ECG heartbeats are better compared to other filters.

R-peak Detection There are several robust available methods for R peak detection in ECG signals such as Pan-Tompkins algorithm [25] based on signal processing or CNN-LSTM architectures [41] based on deep learning. However, in this study, the Pan-Tompkins algorithm is used to detect the R peaks, and the algorithm for improving R-peak detection is beyond the scope of this manuscript.

Segmentation Similar to other related works [2], the ECG signals are segmented to beats by taking 250 points before the peak (R-point) and 400 points after the peak.

Downsampling This processing step in our proposed method saves huge amount of energy consumption by reducing the number of samples to be processed by the CNN architecture for each heartbeat. The real motivation behind decreasing the number of samples in the ECG beats is that there are many published works investigates the role of sampling rate in classifying ECG signals into different cardiac diseases. For instance, to determine the effects of input size or resolution on the performance, Zhai et al. [42] compared six different input sizes of CNN classifiers for detecting arrhythmic beats. It is found from that study that the classification performance for some specific abnormal beats (Supraventricular ectopic beats) is the lowest degree at a small input size (downsampling) which is likely due to the low resolution of input to capture the necessary information of the original ECG signal. However, detection performance for V-type (Ventricular ectopic beats) beat stays relatively high, probably because that kind of beats are usually well distinguished from other beat types thanks to their waveform morphology. While determining how much downsampling is possible without decreasing the classification performance, the power spectral density estimation (PSD) is used to observe the important frequencies of the heartbeat segments. Figure 3.2 given in below shows the PSD of a filtered heartbeat segment, the frequency components which carry more information compared to the other frequency ranges or bands are included the resampled signal. As can be seen from this Figure all the necessary frequency

components are present within 125 Hz of the signal which means a sampling frequency of 250 Hz should be enough based on Nyquist theorem [17]. This sampling frequency corresponds to one-fourth of the frequency used in the PTBDB which is 1 kHz. Therefore, each heartbeat segment which consists of 600 samples is resampled by one-fourth which is 150 samples. The resampling is done by applying an anti-aliasing low pass filter to each segment using Kaiser window method. Then the segments are downsampled by 4 times. We use the resample function which is available in MATLAB (MATLAB and Signal Processing Toolbox Release R2020b,The MathWorks, Inc, USA).

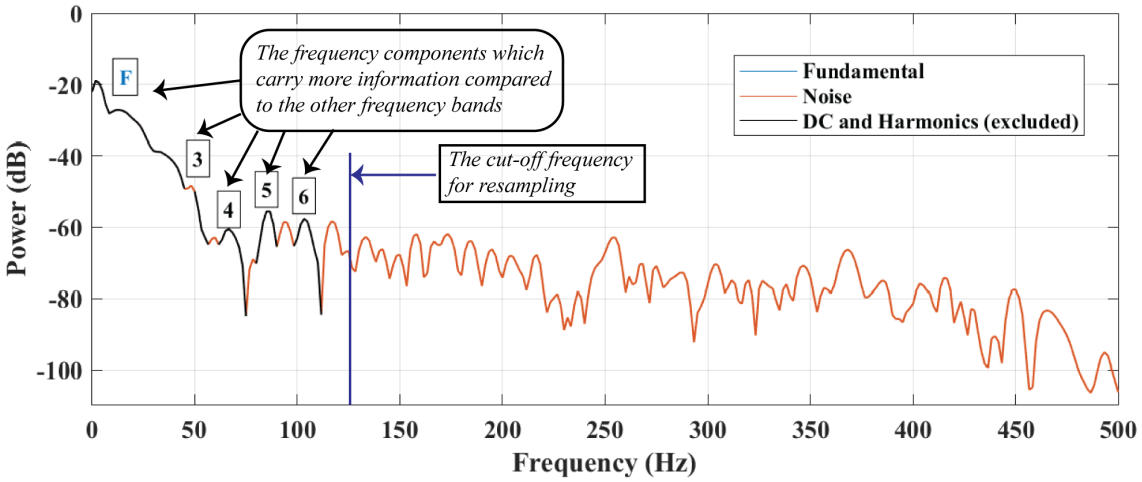

Figure 3.2: The Power Spectrum of Filtered ECG Heartbeat

3.2.2 Proposed CNN Architecture

As shown in Figure 3.1, the proposed CNN architecture has three main parts. The first part is the multi-output CNN architecture that classifies the heartbeat segments into MI or normal, the second is the template control that decides which exit should be used based on the cross-correlation of each heartbeat segment against the template beat, the final part of the proposed system is the decision algorithm that uses the correlation coefficient from the template control as an additional exit condition along with the classification confidence of the first output block.

Table 3.1: Multi-output CNN Architecture Details

Layer Name	**Kernel Size**	**Stride Size**	**Activation Function**	**Output Shape**	**# of Param.**
Input	-	-	-	159x1	0
Conv 1	31	1	ReLU	120x3	96
Pooling 1	2	2	-	60x3	0
Batch Norm.	-	-	-	60x3	12
Fc 1	-	-	Sigmoid	2x1	181
Total Number of Parameters Before the First Output					**289**
Conv 2	7	1	ReLU	54x8	176
Pooling 2	2	2	-	27x8	0
Batch Norm	-	-	-	27x8	32
Dense 2	-	-	Sigmoid	2x1	217
Total Number of Parameters for the Architecture					**533**

Multi-output CNN Architecture

While designing the CNN architecture, the constraints of resource-impoverished wearable devices are considered. Therefore, It has 2 convolution blocks and 2 classification (exits) blocks which are placed after each convolution block. Figure 3.1 shows the overall architecture of the designed CNN architecture. Each convolution block consists of one convolution layer which is passed through ReLU activation, one max-pooling layer, and one batch normalization layer. Each output block consists of one flattening layer, and one linear (dense) layer which is passed through sigmoid activation for final decision. As given in Table 3.1, the total number of parameters required to classify a heartbeat segment after the first, and second output block is 289, and 533, respectively. The baseline classifier is considered as the whole architecture only with the last output.

Template Control

Traditionally, the early exit deep learning architectures utilize brute force search to find which exit should be used. However, in this work, the Template Control is used to avoid this.

This control block utilizes the template matching mechanism using the Pearson correlation coefficient to determine whether the CNN architecture should exit from first output or not. While creating a template, we have used 20 random MI beats. These beats are aligned according to their R-peak and averaged, then the correlation is calculated [3] using Equation 3.4.

$$\rho(T,x) = \frac{1}{N-1}\sum_{n=1}^{N}\left(\frac{T[n]-\mu_T}{\sigma_T}\right)\left(\frac{x[n]-\mu_x}{\sigma_x}\right) \tag{3.4}$$

Where σ_T and μ_T are the standard deviation and mean of the template beat (T) and σ_x and μ_x are the standard deviation and mean of the incoming beat, respectively.

If the absolute value of the Equation 3.4, $|\rho(T,x)|$, is greater than the correlation threshold, $Th_{Correlation}$, which implies the morphological waveform of the segment is similar to template beat and have a higher chance to be correctly classified by the early exit. Therefore, the architecture utilizes the early exit to classify the heartbeat instead of using whole architecture. However, if the $|\rho(T,x)|$ is lower than threshold, it directly uses the baseline architecture to classify a particular segment. However, it should be emphasized that even if the template control decides to use the early exit, it does not guarantee early exit. The early exit decision is made by the algorithm as discussed in next Section 3.2.2.

Early Exit Algorithm

In order to decide whether the early exit should be used or not, an additional algorithm is developed and used. Figure 3.3 shows the overall flowchart of the developed algorithm. As the extracted template beat is created using MI heartbeats, $1-|\rho(T,x)|$ can be used to measure the similarity of the coming beat with the normal heartbeats. Also, as the sigmoid

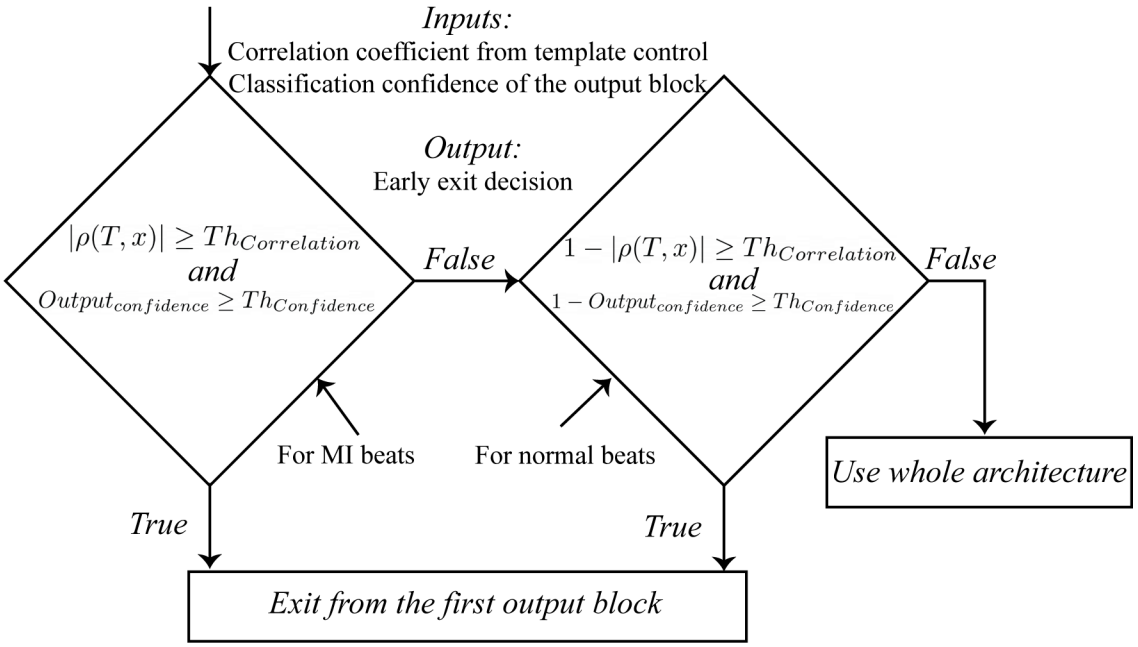

Figure 3.3: Flowchart of the proposed algorithm

function is used at the output, $Output_{confidence}$ represents the classification confidence for MI beats. Therefore, $1 - Output_{confidence}$ can be represented as the classification confidence for normal beats. Using these both correlation coefficient and confidences, the proposed algorithm makes the final decision. The reason why the confidence and correlation coefficient are used at the same time is that this allows us to overcome the limitations of architectures which purely depend on classification confidence. For example, when the exit decision is solely based on confidence and output block misclassifies a hearbeat with high confidence it will exit without investigating the beat more. However, in the proposed design, there is an additional control mechanism.

Table 3.2: Correlation and Confidence Threshold Analysis of the Architecture

$Th_{Correlation}$	$Th_{Confidence}$	Performance (%)			Output Block Used		
Template control	**Early exit**	**Accuracy**	**Sensitivity**	**Specificity**	**First**	**Second**	**Both**
0.5	0.5	98.84	99.06	97.78	39740	0	22566
	0.6	98.88	99.06	97.98	39494	0	22812
	0.7	98.92	99.06	98.20	39238	0	23068
	0.8	98.95	99.08	98.31	38855	0	23451
	0.9	98.99	99.10	98.44	38140	0	24166
0.6	0.5	98.88	99.10	97.78	36097	7271	18938
	0.6	98.91	99.10	97.98	35866	7271	19169
	0.7	98.95	99.10	98.20	35618	7271	19417
	0.8	98.99	99.12	98.33	35264	7271	19771
	0.9	99.02	99.13	98.46	34599	7271	20436
0.7	0.5	99.05	99.15	98.59	30006	20548	11752
	0.6	99.08	99.14	98.77	29814	20548	11944
	0.7	99.11	99.14	98.94	29607	20548	12151
	0.8	99.13	99.15	99.05	29304	20548	12454
	0.9	99.16	99.16	99.15	28727	20548	13031
0.8	0.5	99.19	99.17	99.32	17723	39500	5083
	0.6	99.20	99.16	99.39	17621	39500	5185
	0.7	99.21	99.16	99.44	17509	39500	5297
	0.8	99.22	99.17	99.52	17326	39500	5480
	0.9	**99.24**	**99.18**	**99.54**	**16953**	**39500**	**5853**
0.9	0.5	99.22	99.12	99.71	1260	59670	1376
	0.6	99.22	99.12	99.71	1256	59670	1380
	0.7	99.23	99.13	99.72	1251	59670	1385
	0.8	99.24	99.14	99.74	1241	59670	1395
	0.9	99.25	99.16	99.74	1229	59670	1407

3.3 Experimental Results and Analysis

3.3.1 Performance Evaluation of CNN Architecture

The accuracy for each output block of our multi-output CNN classifier is given in Figure 3.4, which is 94.82% and 99.33% for the early exit and whole architecture respectively. Furthermore, the early exit reaches a sensitivity, and specificity of 94.21%, and 97.83%, respectively while the complete architecture achieves higher sensitivity, and specificity of 99.25%, and 99.74% respectively. The results prove and motivate that each output block has similar performance for all three metrics that are used in this work. Therefore, it guarantees

that the trained CNN model is not biased to any specific class, MI or normal.

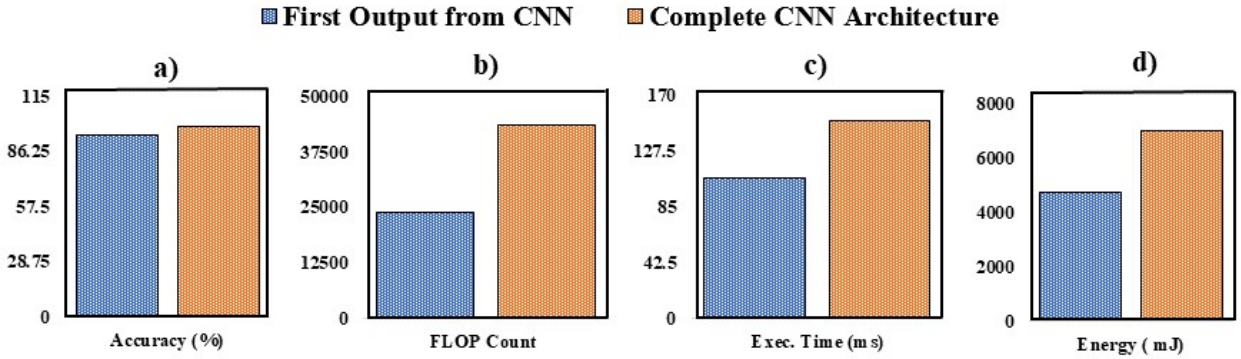

Figure 3.4: Exit-wise Statistics of Multi-output CNN Architecture

3.3.2 Correlation and Confidence Threshold Analysis

The correlation threshold, $Th_{Correlation}$ of the template control, and the confidence threshold, $Th_{Confidence}$ of the early exit play a significant role in the proposed CNN architecture. Therefore, the different combinations of these two threshold values and their effect on the classification performance with the early exit decision are investigated. The threshold value of the last output of the CNN architecture is set to 0.5 as it is general value of the confidence threshold for the complete CNN architectures. Table 3.2 shows the performance and the different output blocks used to classify the beats. The beats that are initially chosen for the early exit by the template control but do not exit after first output and end up using both the output block. Table 3.2 shows that the classification performance of the architecture improves with the increase of $Th_{Correlation}$. This can be explained by investigating the behaviour of the template control. When it is observed, it can be easily seen that the template control picks more beats to be classified by the last output block which has better performance. For each $Th_{Correlation}$ value the performance also increases with increase of the $Th_{Confidence}$ value. This event occurs because the developed algorithm does not allow early exit after first output as it examines the classification confidence and if it is lower than the threshold value, it rejects the usage of early exit and the beats are sent to the second output

block which has better performance. This is the reason why the number of beats which are classified by early exit decreases and increases the value for both output blocks in Table 3.2. If the majority of the segments ends up using both output block, our proposed architecture may require to spend more energy than the baseline model which has no early exit. Thus, the purpose of this threshold value optimization is to find the best best mixture that provides a similar performance of the developed method with the baseline architecture while keeping energy consumption low. As shown in Table 3.2, $Th_{Correlation} = 0.8$ and $Th_{Confidence} = 0.9$ achieves the best performance considering all three metrics and it gives a good balance among different output blocks used to classify the segments. Therefore, we use those two values for the correlation threshold of template control and the confidence threshold of early exit in our proposed architecture. Moreover, the proposed architecture provides a huge stretchability to designers as the two threshold values can be lowered to save energy more while achieving better performance.

3.3.3 Overall Performance Evaluation

As shown in Table 3.3, our baseline architecture that uses all convolutional blocks reaches an accuracy of **99.33%**, sensitivity of **99.25%**, and specificity of **99.74%**. Our proposed CNN architecture (for $Th_{Correlation} = 0.8$, $Th_{Confidence} = 0.9$) provides a similar performance as the baseline with an accuracy, sensitivity, and specificity of 99.24%, 94.18%, 99.54%, respectively. Also, both CNN architectures which are introduced in this work significantly outperform the other related works [2, 35, 36, 1, 29] in all three metrics, except the sensitivity of the [2] is the best amongst all works. Furthermore, SVM and RF models, in general, are not suitable for edge devices (wearable in general) in terms of memory footprint which is discussed in the next Section **??**. The work proposed in [1] using deep CNN achieves comparatively better performance compared to other solutions where the target platform is wearable [35, 36, 29]. However, architectures from our proposed solutions still provides better performances. The

Table 3.3: Performance Comparison of Related Works

Works	**Classifier Type**	**Performance** (%)		
		Accuracy	**Sensitivity**	**Specificity**
[2]	k-NN	98.80	99.45	96.27
[35]	Full SVM	95	–	–
	2-level SVM	90	–	–
[36]	Full RF	83.26	87.95	78.82
	5-level RF	80.32	81.02	79.63
[1]	CNN	95.22	95.49	94.19
[29]	BCNN	90.29	90.41	90.16
Ours	**Baseline CNN**	**99.33**	**99.25**	**99.74**
	Proposed CNN	**99.24**	**99.18**	**99.54**

work [1] which is developed for clinical setups. The work [2] reaches the highest performance amongst the related works with accuracy, sensitivity, and specificity of 98.80% and 99.45%, and 96.27% respectively. However, the classifier type of this work, k-NN, is not a feasible solution for the wearable devices since during the inference (classification of MI and normal beats), all the training data should be stored in the memory, which makes it impossible for deploying the model in the resource constrained devices.

3.3.4 Memory Footprint Evaluation on Real Hardware

The memory footprint of all works given in Table 3.3 except for the work [2] that uses k-NN classifier are evaluated. Because as it was mentioned in the previous section the k-

NN classifier is not suitable for wearable devices as no effort has been spent to implement algorithm for evaluating memory. For the traditional machine learning approaches in [35, 36] the evaluated memory requirements is for the feature extraction and classification process. For the deep learning approaches using CNN [1, 29], the memory footprint of the classification is evaluated as the deep learning models automatically extracts features during classification, there is no separate feature extraction process.

The SVM [35] and RF [36] works use multi-level classifiers with 2-level SVM and a 5-level RF classifier respectively. Both works have three major drawbacks in terms of memory. Firstly, when the number of samples in the training dataset increases, the model size of the SVM and RF increases as well. Secondly, both methods require to deploy all level of classifiers to device during the runtime (2 for SVM and 5 for RF). Thirdly, as the SVM and RF are used for inference, they have an additional feature extraction step which requires extra memory and energy consumption.
Moreover, the work [35] requires 2 of the SVM models and the work requires 5 of the RF models to be loaded into the memory which gives us the perspective of the *second* drawback. Besides, both the approaches in [35, 36] require a huge amount of memory for the feature extraction process of their full level classifiers which brings us to the *third* drawback. We demonstrate the *third* drawback using the RAM footprint for each level of the SVM and RF classifier on the EFM32 Giant Gecko microcontroller which has 128 KB of RAM and 1 MB of flash memory. As our goal is to evaluate the memory requirement for the feature extraction of different levels of SVM and RF, we train the SVM with only 20 training samples and RF with only 100 training samples so that they can fit within the 128 KB of RAM. Also, for the RF model, we use only 10 weak learners with a fixed split of 10. As shown in Table 3.4, both the full level classifier in [35] and [36] requires almost 83 MB of RAM making them incompatible for wearable devices with lower memory.

A significant advantage of the deep learning approaches [1, 29] over the machine learning ones

Table 3.4: Memory Footprint and Energy Consumption Evaluation on EFM32 Giant Gecko Development Board

Works	Classifier Level	RAM Footprint (KB)	Exe. Time (ms)	Avg. Power (mW)	Energy (μJ)
SVM[35]	First (5 Features)	78.8	347.03	46.57	16161
	Full (47 Features)	Not Compatible: RAM Overflowed			
RF[36]	First (5 Features)	78.8	345.35	46.57	16082
	Second (10 Features)	85.9	3556.37	46.58	165655
	Third (15 Features)	87.3	7669.7	46.81	359018
	Fourth (20 Features)	88.1	8188.21	46.48	380588
	Full (72 Features)	Not Compatible: RAM Overflowed			
CNN[1]	No feature extraction	114.1	2036.82	46.97	95669
BCNN[29]	No feature extraction	3.5	253.73	44.45	11278
CNN[Ours]	Template control	2.6	3.18	44.98	143
	Early Exit	11.8	107.32	44.54	4780
	Baseline (w/o Early exit)	**20.1**	**150.77**	**46.75**	**7048**
	Proposed	**20.1**	**142.12**	**45.64**	**6486**

[35, 36] is that the number of the parameter in the CNN architecture is constant and does not affect by the number of training samples. However, the memory requirement of those models still depends on the architecture size, parameters, and input size. For example, the proposed CNN architecture presented in [1] has 11 layers and requires 114 KB of RAM. In contrast, the work in [29] targets low memory wearable devices and requires only 3.5 KB of RAM. The approach in [29] focused on memory and energy efficiency while sacrificing performance. Although this behavior, decreasing memory requirement while sacrificing classification performance, achieved better performance compared to other machine learning works [35, 36] for resource-constrained devices, it is still problematic for the myocardial infarction, which requires more accurate detection. This paper emphasizes the performance while still being energy and memory-efficient compared to all the state-of-the-artwork [35, 36, 1] except for memory efficiency against [29].

Table 3.4 shows the RAM footprint our baseline architecture is 20 KB. The template control requires only 2.61 KB of RAM without adding extra memory overhead to the proposed algorithm. The RAM footprint of our complete architecture, which is given as **Proposed**, is also 20 KB which is the maximum of RAM footprints of the template control and each

of the output blocks. Thus, our proposed algorithm is compatible with any device with a minimum RAM of 32 KB. As shown in Table 3.4, our proposed architecture has much less memory requirement than all the state-of-the-art works [35, 36, 1] except [29].

3.3.5 Energy Consumption Evaluation on Real Hardware

While evaluating the energy consumption of proposed method, the same EFM32 Giant Gecko microcontroller is utilizied which is used for memory evaluation. For the traditional machine learning approaches [35, 36], the energy for the feature extraction and classification are calculated. For deep learning ones using CNN, we evaluate the classification energy only as they automatically extract features during classification. The detailed analysis of execution time, power, and energy for one heartbeat of the data using the 48 MHz clock speed of the microcontroller are given in Table 3.4.

For the energy evaluation of work [35], the SVM classifier is trained with only 20 training samples of heartbeats to fit the SVM model in the memory. Similarly, the RF classifier in [36] is trained with 100 training samples. While training the RF classifier, the number of weak learners are set to 10 with a fixed split of 10 to fit the model into the RAM.

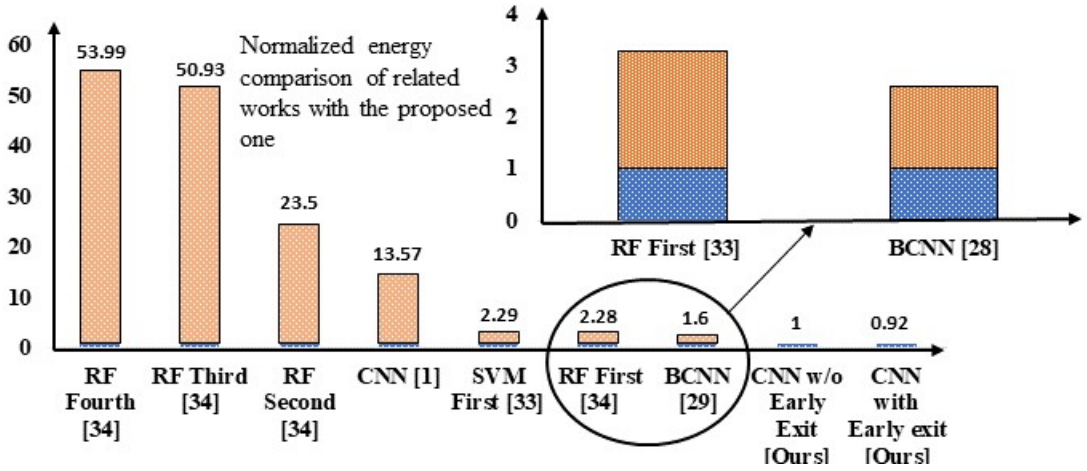

Figure 3.5: Comparison of Energy Consumption with respect to Baseline Classifier

Figure 3.5 shows the energy consumption of various works reported in Table 3.4. Also, to demonstrate the contribution of the proposed method, the results are given by normalizing with respect to the energy consumption of our proposed complete CNN architecture without early exit. As shown in Figure 3.5, RF first, second, third, and fourth level classifiers in [36] require 1.28×, 22.50×, 49.93×, and 52.99× more energy compared to our CNN classifier. All the 20 features calculated in RF fourth level are the same as the first 20 features out of the 47 features in full SVM classifier of [35]. This indicates that the full SVM and RF classifiers will consume much more energy compared to our baseline classifier. Even the first level SVM classifier in [35] consumes 1.29× more energy compared to our baseline classifier. Moreover, it should be noted that while training the SVM and RF models, random training samples are chosen instead of whole dataset to fit the models in the wearable device. Althoygh, the BCNN [29] which consumes the lowest energy compared to the related works also consumes 60% more energy than the our CNN classifier. However, the proposed architecture with early exit consumes 8% less energy compared to complete architecture as shown in Figure 3.5 which proves the energy efficiency of proposed method.

Chapter 4

Conclusion

In this study, two different algorithms are presented to provide the continuous real-time monitoring of human health based on electroencephalogram (EEG) and electrocardiogram (ECG) signals. A novel lightweight single-channel EEG-based method is presented to estimate the arousal level, defined as an individual's degree of alertness or responsiveness, at wearable devices. The validation of the presented method is done using the scalp EEG recorded during overnight sleep and intra-operative anesthesia with technician-scored hypnogram annotations at the University of California at Irvine Medical Center. Also, a novel resource-efficient template control-based Convolutional Neural Network (CNN) architecture is presented for detecting myocardial infarction using the ECG signals. To validate the CNN architecture, the well-known PTB diagnostic ECG database is used. Evaluation of real hardware shows that the proposed methodologies can be implemented for devices with a minimum RAM of 512 KB while maintaining high accuracy with low energy consumption compared to the state-of-the-art works.

Bibliography

[1] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam. Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals. *Information Sciences*, 415:190–198, 2017.

[2] U. R. Acharya, H. Fujita, V. K. Sudarshan, S. L. Oh, M. Adam, J. E. Koh, J. H. Tan, D. N. Ghista, R. J. Martis, C. K. Chua, et al. Automated detection and localization of myocardial infarction using electrocardiogram: a comparative study of different leads. *Knowledge-Based Systems*, 99:146–156, 2016.

[3] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[4] J. G. Betts, P. Desaix, and E. Johnson. *Anatomy & physiology*. Houston, Texas : OpenStax College, Rice University.

[5] R. Bousseljot, D. Kreiseler, and A. Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. *Biomedizinische Technik/Biomedical Engineering*, 40(s1):317–318, 1995.

[6] B. U. Demirel, I. A. Bayoumy, and M. A. A. Faruque. Energy-efficient real-time heart monitoring on edge-fog-cloud internet-of-medical-things. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[7] B. U. Demirel, I. Skelin, H. Zhang, J. J. Lin, and M. Abdullah Al Faruque. Single-channel eeg based arousal level estimation using multitaper spectrum estimation at low-power wearable devices. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 542–545, 2021.

[8] S. Elmalaki. Fair-iot: Fairness-aware human-in-the-loop reinforcement learning for harnessing human variability in personalized iot. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, IoTDI '21, page 119–132, New York, NY, USA, 2021. Association for Computing Machinery.

[9] S. Elmalaki, B. U. Demirel, M. Taherisadr, S. Stern-Nezer, J. J. Lin, and M. A. A. Faruque. Towards internet-of-things for wearable neurotechnology. In *2021 22nd International Symposium on Quality Electronic Design (ISQED)*, pages 559–565, 2021.

[10] S. Elmalaki, B.-J. Ho, M. Alzantot, Y. Shoukry, and M. Srivastava. Spycon: Adaptation based spyware in human-in-the-loop iot. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 163–168, 2019.

[11] R. Gao, E. J. Peterson, and B. Voytek. Inferring synaptic excitation/inhibition balance from field potentials. *NeuroImage*, 158:70 – 78, 2017.

[12] L. Genzel, M. C. Kroes, M. Dresler, and F. P. Battaglia. Light sleep versus slow wave sleep in memory consolidation: a question of global versus local processes? *Trends in Neurosciences*, 37(1):10 – 19, 2014.

[13] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220, 2000.

[14] D. Jabaudon, J. Sztajzel, K. Sievert, T. Landis, and R. Sztajzel. Usefulness of ambulatory 7-day ecg monitoring for the detection of atrial fibrillation and flutter after acute stroke and transient ischemic attack. *Stroke*, 35(7):1647–1651, 2004.

[15] J. Kortelainen and T. Seppänen. Eeg-based recognition of video-induced emotions: Selecting subject-independent feature set. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4287–4290, 2013.

[16] J. Kortelainen and T. Seppänen. Eeg-based recognition of video-induced emotions: Selecting subject-independent feature set. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4287–4290, 2013.

[17] H. Landau. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706, 1967.

[18] J. D. Lendner, R. F. Helfrich, B. A. Mander, L. Romundstad, J. J. Lin, M. P. Walker, P. G. Larsson, and R. T. Knight. An electrophysiological marker of arousal level in humans. *eLife*, 9:e55092, jul 2020.

[19] H. Li, K. Ota, and M. Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE network*, 32(1):96–101, 2018.

[20] M. Mahmood, D. Mzurikwao, Y. S. Kim, Y. Lee, S. Mishra, R. Herbert, A. Duarte, C. S. Ang, and W. H. Yeo. Fully portable and wireless universal brain–machine interfaces enabled by flexible scalp electronics and deep learning algorithm. *Nature Machine Intelligence*, 1(9):412–422, Sept. 2019.

[21] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. *IEEE Transactions on Biomedical Engineering*, 58(9):2456–2466, 2011.

[22] P. Memar and F. Faradji. A novel multi-class eeg-based sleep stage classification system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(1):84–95, 2018.

[23] M. Odema, N. Rashid, B. U. Demirel, and M. A. A. Faruque. Lens: Layer distribution enabled neural architecture search in edge-cloud hierarchies. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 403–408, 2021.

[24] M. Odema, N. Rashid, and M. A. A. Faruque. Eexnas: Early-exit neural architecture search solutions for low-power wearable devices. In *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6, 2021.

[25] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236, 1985.

[26] M. Perslev, M. H. Jensen, S. Darkner, P. J. Jennum, and C. Igel. U-time: A fully convolutional network for time series segmentation applied to sleep staging. *CoRR*, abs/1910.11162, 2019.

[27] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos. Joint classification and prediction cnn framework for automatic sleep stage classification. *IEEE Transactions on Biomedical Engineering*, 66(5):1285–1296, 2019.

[28] H. Phan, O. Y. Chén, M. C. Tran, P. Koch, A. Mertins, and M. D. Vos. Xsleepnet: Multi-view sequential model for automatic sleep staging, 2021.

[29] N. Rashid and M. A. Al Faruque. Energy-efficient real-time myocardial infarction detection on wearable devices. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 4648–4651, 2020.

[30] N. Rashid, L. Chen, M. Dautta, A. Jimenez, P. Tseng, and M. A. Al Faruque. Feature augmented hybrid cnn for stress recognition using wrist-based photoplethysmography sensor. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 2374–2377, 2021.

[31] N. Rashid, M. Dautta, P. Tseng, and M. A. A. Faruque. Hear: Fog-enabled energy aware online human eating activity recognition. *IEEE Internet of Things Journal, doi: 10.1109/JIOT.2020.3008842.*, 2020.

[32] N. Rashid, B. U. Demirel, and M. A. A. Faruque. Ahar: Adaptive cnn for energy-efficient human activity recognition in low-power edge devices. *IEEE Internet of Things Journal*, pages 1–1, 2022.

[33] N. Rashid, B. U. Demirel, M. Odema, and M. A. Al Faruque. Template matching based early exit cnn for energy-efficient myocardial infarction detection on low-power wearable devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2), 2022.

[34] N. Rashid, T. Mortlock, and M. A. Al Faruque. Self-care: Selective fusion with context-aware low-power edge computing for stress detection. In *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2022.

[35] D. Sopic, A. Aminifar, A. Aminifar, and D. Atienza. Real-time classification technique for early detection and prevention of myocardial infarction on wearable devices. In *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2017.

[36] D. Sopic, A. Aminifar, A. Aminifar, and D. Atienza. Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems. *IEEE transactions on biomedical circuits and systems*, 12(5):982–992, 2018.

[37] K. Tammemäe, A. Jantsch, A. Kuusik, J.-S. Preden, and E. Õunapuu. Self-aware fog computing in private and secure spheres. In *Fog Computing in the Internet of Things*, pages 71–99. Springer, 2018.

[38] D. J. Thomson. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9):1055–1096, 1982.

[39] M. Tudor, L. Tudor, and K. I. Tudor. [Hans Berger (1873-1941)–the history of electroencephalography]. *Acta Medica Croatica: Casopis Hravatske Akademije Medicinskih Znanosti*, 59(4):307–313, 2005.

[40] P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

[41] B. Yuen, X. Dong, and T. Lu. Inter-patient cnn-lstm for qrs complex detection in noisy ecg signals. *IEEE Access*, 7:169359–169370, 2019.

[42] X. Zhai and C. Tin. Automated ecg classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access*, 6:27465–27472, 2018.