SINGLE-CHANNEL REAL-TIME DROWSINESS DETECTION BASED ON ELECTROENCEPHALOGRAPHY

by
Hassan Fahad Albalawi

Department of Electrical and Computer Engineering
Duke University

Date:_____

Approved:

_____
Xin Li, Supervisor

_____
John Board

_____
Hai Li

_____
Patrick Wang

_____
Mikhail Lebedev

Thesis submitted in partial fulfillment of
the requirements for the degree of Doctor
of Philosophy in the Department of
Electrical and Computer Engineering
in the Graduate School
of Duke University

2018

ABSTRACT

SINGLE-CHANNEL REAL-TIME DROWSINESS DETECTION BASED ON ELECTROENCEPHALOGRAPHY

by
Hassan Fahad Albalawi

Department of Electrical and Computer Engineering
Duke University

Date:_____

Approved:

_____
Xin Li, Supervisor

_____
John Board

_____
Hai Li

_____
Patrick Wang

_____
Mikhail Lebedev

An abstract of a thesis submitted in partial fulfillment of
the requirements for the degree of Doctor
of Philosophy in the Department of
Electrical and Computer Engineering
in the Graduate School
of Duke University

2018

Abstract

Drowsiness is considered as a major risk factor in workplace injuries and fatalities as much as alcohol. Drowsiness-related accidents tend to be catastrophic. The need of a reliable drowsiness detection system is arising today, as drowsiness is considered as a major cause for many accidents in different sectors. In this thesis, we propose a real-time drowsiness detection system based on a single-channel electroencephalography (EEG). Towards that goal, we introduced three main contribution proposed in this thesis: (1) a real-time drowsiness detection algorithm based on EEG suitable for portable applications with low computational complexity; (2) several novel algorithms to train classifiers that can be implemented on chip with low-power fixed-point arithmetic with extremely small word length; (3) an instantaneous drowsiness detection system suitable for short-time windows of single-channel EEG signal. The proposed real-time drowsiness detection algorithm adopts a cumulative counter to extract important features from 8 different frequency bands. Our experimental results demonstrate that the proposed algorithm is capable of detecting drowsiness with superior accuracy (83.36%) over the conventional method (70.62%). The proposed fixed-point

algorithms incorporate the non-idealities (i.e., rounding and overflow) associated with fixed-point arithmetic into the offline training process so that the resulting classifiers are robust to these non-idealities. Our numerical experiments demonstrate that the proposed methods are able to achieve up to 1.67x reduction in the word length compared to the conventional approaches without surrendering any classification accuracy. The instantaneous drowsiness detection algorithm proposed in this work is based on Convolutional Neural Network (CNN). Our experimental results demonstrate that our CNN-based drowsiness detection system is capable of detecting drowsiness in short-time windows (five seconds) with higher accuracy (84.8%) compared to conventional methods (71.0%) and the counter-based method (77.2%). Finally, we briefly discuss few possible research tasks for the future: (1) wearable derive for industrial workers, (2) fixed-point implementation for CNN, and (3) multimode data fusion.

Dedication

Dedicated to my beloved parents & family.

Contents

List of Tables

List of Figures

1. Introduction

Drowsiness is considered a significant risk factor in workplace injuries and fatalities [1]. Many studies have shown a significant association between an increased level of drowsiness and higher risk of accidents and injury [2]. Drowsiness is characterized by a low level of consciousness and difficulty in staying awake [3]. Previous studies argue that drowsiness plays the most important role in human errors [4]. In hazardous environments, such as construction and underground mining, errors and mistakes can lead to casualties and massive economic losses [5]. According to the Bureau of Labor Statistics, 828 fatalities were reported in the construction sector in 2013 [6]. Moreover, drowsiness has been identified as a potential risk factor related to slip-induced falls [7].

Drowsiness is a common and chronic problem in many countries, both developing and developed [8]. Workers in a construction field or an underground mine are known for their long working hours, multiple shifts, and heavy workload requirements under challenging environmental conditions [9] [5]. These are factors that heavily contribute to drowsiness [8]. Over the past four decades, many studies have been developing various drowsiness measures. Initially, researchers depended on subjective measures that were based on self-reporting

questionnaires. But in the past two decades, there have been many applications and approaches developed to measure drowsiness more objectively. Most of the drowsiness-countermeasure techniques that have been developed focus on the application of preventing drowsy driving.

Unlike driving vehicles, working in a construction field or an underground mine is dynamic, transient, and fragmented [5]. Such challenging environments make it hard and unfeasible to adopt conventional drowsiness-detection systems. Construction and mining laborers are typically involved in physically demanding work often performed in harsh environmental conditions. This can cause fatigue and lead to poor judgment, an increased risk of accidents, and a decrease in productivity [9, 10]. Therefore, a practical drowsiness detection system suitable for the industrial environments needs to be designed and implemented to effectively improve safety and productivity.

1.1 Drowsiness Detection Systems

In recent years, drowsiness detection has become a common topic for many studies, many of which have focused on detecting drowsiness by observing different measures. All methods used in detecting drowsiness can be classified into four groups:

i) Subjective measures—Self-administered questionnaires that focus on evaluating subjective feelings of the consequences of drowsiness. The questionnaires usually take 2-3 minutes to complete, making them suitable and non-intrusive for workers to complete before or during their shift.

ii) Embedded-performance measures—Different advanced systems are adopted, including monitoring the movement of the steering wheel, braking or acceleration, and lane departure to determine drowsiness levels of drivers. The key advantage of utilizing such measures is that embedded-performance measures are non-intrusive and more acceptable by users [11]. However, the performance of detecting drowsiness using vehicle driving patterns is poor, especially that changes in vehicle driving patterns are not specific to drowsiness [12] [1]. Many studies have shown that vehicle driving patterns can be also affected by any type of impaired driving, including driving under the influence of alcohol or other drugs [12]. Moreover, such a measure is not well suited for all construction workers or underground miners; however, previous work has suggested adopting vehicle-based measures in heavy machinery, such as trucks and cranes [11, 13].

iii) Behavioral measures—These include focusing on workers' behavior for detecting yawning patterns, eye closure, overall head movement, and facial

expression. Such measures have been widely used in detecting drowsy driving. The driver fatigue recognition system aims to warn drivers when they are tired and prevent fatigue-based accidents. This system normally identifies the driver's fatigue level by the head swing, blinking interval, or blinking period. Existing literature includes a system that analyzes the visual attention of drivers, which depends on estimating color statistics and global motion to monitor the facial features and head position of a driver. In the system, a rotation is classified in every viewing direction. The system can monitor, through occlusion based on eye blinking, large-mouth movement, eye closure, and occurrences, particularly if the driver's face is blocked because of rotation [14, 15]. The system can be used for advanced monitoring of the driver's visual attention because it does not halt. Literature also includes a real-time driver fatigue system that uses attached cameras armed with active infrared illuminators to capture the driver's video images in real-time [16-18]. These illuminators systematically combine various visual signals to define the driver's fatigue level. The visual signals use eyelid movement, facial expression, and head movement to acquire the level of driver's fatigue. In the industrial sectors, behavioral measures are applied in hauling

trucks. Recently, Caterpillar's fatigue management solution incorporates technologies such as Driver Safety System (DSS) to prevent drowsy driving [19].

iv) Physiological characteristics—Various physiological signals, including Electrocardiography (ECG); Electrooculogram (EOG); and Electroencephalography (EEG), have been used by many researchers to estimate drivers' drowsiness. Physiological signals have the advantages in quantifying drowsiness levels precisely; however, the intrusive nature of acquiring physiological signals is considered a major concern that needs to be addressed to enable physiological-based real-world applications [16]. Relatively to other drowsiness measures, physiological measures are considered more reliable and as more accurate indicators of drowsiness than aforementioned drowsiness measures [12] [20].

1.2 Drowsiness Detection Algorithms

In this thesis, we mainly focus on physiological characteristics because they are most reliable and accurate among all available methods. There are many methods that have been proposed in the past few years to detect drowsiness using physiological measures. However, EEG has been considered as the golden standard to capture the changes from vigilant or alert to drowsy or asleep

promptly [21]. EEG has high temporal resolution, which makes it optimal for time-sensitive applications such as drowsiness detection. Also, one of the biggest advantages of EEG is its practicality. EEG equipment is relatively inexpensive to other devices, and is portable and easy to operate.

The related literature contains various approaches that focus on using EEG to assess mental fatigue in workers performing mental tasks [22]. Also, EEG has been utilized to assess mental workload changes in construction workers during specific and controlled tasks; particularly, participants were required to climb a ladder, select the proper nut, and fasten the nut to a bolt [23].

In a comprehensive review of the literature done by [24], it was found that the transition to drive performance, from normal drive to high mental workload and eventually to drowsiness, can be observed from the changes in EEG activities; high mental workload is described by the increase of EEG power in theta (4-8Hz) band and a decrease in alpha (8-12Hz) band, while mental fatigue or drowsiness is defined by the increase of EEG power in delta (0-4Hz) as well as theta and alpha bands.

The research in EEG-based drowsiness detection systems has been actively expanding using various signal processing and machine learning approaches for

feature extraction and classification, which was evaluated using self-collected data [14, 17, 25-31] or using a publicly available and labeled sleep dataset [15, 21, 32]. Moreover, EEG-based drowsiness detection systems need to be portable and suitable for integration with existing wearable applications. Therefore, low-cost algorithms for detecting drowsiness will enable EEG-based real-world applications.

Traditionally, many machine learning algorithms are only established and authenticated by their software applications (i.e., C++, MATLAB,) based on double-precision floating-point arithmetic. How these algorithms can be correctly mapped to a low-power circuit employed with fixed-point arithmetic remains an open question. The current study will show that rounding error acquired from fixed-point arithmetic can considerably falsify the classification output if these machine learning algorithm are incorrectly modeled and integrated into the classifier training process [11]. It, in turn, implies that there is a strong need to develop a dedicated training algorithm for fixed-point classifiers.

In order for drowsiness-detection applications to be useful in real-world applications, drowsiness needs to be detected as soon as it happens. Alerting a drowsy driver, for example, requires an instantaneous detection of drowsy states

in order to reduce the risk of drowsy driving accidents. Moreover, data shows that staged-warning feedback, depending on the level of drowsiness, is more effective than discrete alerts [33]. Thus, an instantaneous drowsiness-detection system that monitor drowsiness continuously can be employed to enable more effective drowsiness countermeasures.

1.3 Thesis Outlines

The focus of the current thesis is based on development of a real-time drowsiness detection system suitable for industrial workers. Towards that goal, the proposed work addresses: (1) how drowsiness can be detected using EEG signals in real time; (2) how to design classifiers for on-chip machine-learning classifiers that can be applied for designing low-power portable devices; and (3) how to develop an instantaneous drowsiness detection system suitable for short-time windows of single-channel EEG signal.

The remainder of this thesis is organized as follows. Chapter 2 presents a novel single-channel, real-time drowsiness detection algorithm that is suitable for portable applications with low computational complexity. The proposed algorithm adopts a cumulative counter to extract important features from 8 different frequency bands: delta (1–3 Hz), theta (4–7 Hz), low-alpha (8–9 Hz), high-

alpha (10–12 Hz), low-beta (13–17 Hz), high-beta (18–30 Hz), low-gamma (31–40 Hz), and high-gamma (41–50 Hz). These features are then processed by a support vector machine (SVM) to distinguish between drowsy and awake states. Our experimental results demonstrate that the proposed algorithm is capable of detecting drowsiness with superior accuracy (83.36%) over the conventional method (70.62%).

Chapter 3 presents several novel algorithms developed to train classifiers that can be implemented on chip with low-power fixed-point arithmetic with extremely small word length. These algorithms are based upon linear discriminant analysis (LDA), support vector machine (SVM), and logistic regression (LR), and are referred to as LDA-FP, SVM-FP, and LR-FP respectively. They incorporate the non-idealities (i.e., rounding and overflow) associated with fixed-point arithmetic into the offline training process so that the resulting classifiers are robust to these non-idealities. Mathematically, LDA-FP, SVM-FP, and LR-FP are formulated as mixed integer programming problems that can be robustly solved by the branch-and-bound methods described in this paper. Our numerical experiments demonstrate that LDA-FP, SVM-FP, and LR-FP substantially outperform the

conventional approaches for the emerging biomedical applications of brain decoding.

Chapter 4 presents a CNN-based drowsiness detection system suitable for short-time windows from a single channel of EEG. In this work, the CNN architecture is fed with two-dimensional (2D) signals of frequency-specific activity over time extracted from a single-channel EEG. We used an openly available sleep dataset from 20 healthy young adults with 5-second labeled trials and applied 5-fold cross-validation to design an optimal drowsiness detector. Our experimental results demonstrate that our CNN-based drowsiness detection system is capable of detecting drowsiness in short-time windows with higher accuracy (75%) compared to the conventional methods (65%) and counter-based methods (71%).

Finally, Chapter 5 concludes this dissertation by summarizing the major findings and insights from our proposed work. Several possible directions for future research (e.g., fixed-point implementation for CNN) are also briefly discussed.

2. Real-Time Drowsiness Detection

A variety of methods have been proposed in the literature for drowsiness detection, including vehicular measures [12], behavioral measures [27, 34] and physiological measures [24] [20, 21, 25, 26, 28, 29, 35-39]. Among these methods, physiological measures are often preferred in many real-time applications due to their high accuracy and low latency, thereby preventing drowsiness-related accidents promptly. Most physiological approaches detect drowsiness by acquiring and analyzing electrocardiogram (ECG) [24], electrooculogram (EOG) [20, 25] or electroencephalography (EEG) [21, 26, 28, 29, 35-39]. EEG is the most popular method because it can be reliably measured with low acquisition cost and high temporal resolution.

While EEG offers accurate drowsiness detection, the user comfort during EEG acquisition is often considered as a major concern that must be appropriately addressed [16]. In practice, the user experience heavily depends on the number, location, and type of EEG electrodes. To offer a comfortable experience, a minimal number of electrodes should be used, the location of electrodes should not disturb the normal user behavior, and all electrodes should be easy to set up and use.

The evaluation of drowsiness often relies on visual scoring of sleep stages [3, 40] where EEG signals are visually assessed by experts for 30-second epochs.

The first stage of sleep (i.e., *Stage 1* in [40] or *N1* in [3]) has been considered in the literature as the drowsy stage. According to [3], the drowsy stage is typically characterized by well-defined alpha rhythms for 80%-90% of the population. For the remaining subjects with little or no alpha rhythms, the presence of low-frequency activities over (2–7 Hz), vortex sharp waves, and slow eye movements are often used to identify the drowsy stage.

Other studies in the literature show that physiological activities may offer a coherent indication of drowsiness. In [24], it has been reported that the transition from normal condition to high mental workload and ultimately drowsiness can be identified by EEG activities. High mental workload is characterized by an increase of EEG power in the theta band (4–8 Hz) and a decrease in the alpha band (8–12 Hz). On the other hand, mental fatigue or drowsiness is characterized by an increase of EEG power in the delta band (0–4 Hz) as well as theta and alpha bands.

In this chapter, we propose a novel single-channel, real-time drowsiness detection algorithm that is suitable for portable applications with low computational complexity. Our proposed method measures the average power of EEG signal of multiple 1-second time windows over eight frequency bands: the delta band (1–3 Hz), the theta band (4–7 Hz), the low-alpha band (8–9 Hz), the

high-alpha band (10–12Hz), the low-beta band (13–17 Hz), the high-beta band (18–30 Hz), the low-gamma band (31–40 Hz) and the high-gamma band (41–50 Hz). Next, it extracts a set of important features by adopting a novel approach based on *adaptive counters*. These features are eventually used for accurate drowsiness detection. As will be demonstrated by our experimental results in Section 2.4, the proposed approach provides superior performance over other conventional methods in the literature.

Our proposed drowsiness detection approach is composed of three major components: (i) signal pre-processing, (ii) feature extraction, and (iii) drowsiness detection. The reminder of this chapter is organized as follows. In Section 2.1-3, we describe the proposed drowsiness-detection algorithm and then demonstrate its efficacy by experimental examples in Section 2.4.

2.1 Signal Pre-processing

In this work, the EEG data is collected from a single channel. The proposed pre-processing consists of two major steps. First, the EEG signal is filtered by using a second-order, bi-directional, zero-phase, band-pass Butterworth filter with the passband of 0.5-50Hz. The EEG data is processed in both forward and reverse directions in order to avoid phase distortion [41]. Second, the filtered EEG signal

is partitioned into 1-second segments to assure statistical stationarity required for power density estimation [42]. The average value of each 1-second segment is removed to filter out the DC offset.

2.2 Feature Extraction

Given the EEG signal after pre-processing, we measure the power spectral densities of multiple 1-second time windows over eight frequency bands: the delta band (1–3 Hz), the theta band (4–7 Hz), the low-alpha band (8–9 Hz), the high-alpha band (10–12Hz), the low-beta band (13–17 Hz), the high-beta band (18–30 Hz), the low-gamma band (31–40 Hz) and the high-gamma band (41–50 Hz). It has been reported in the literature that the EEG power over these frequency bands carries important information for drowsiness detection [24], [43] [43].

We estimate the relative power $x_{t,k}$ of the t-th time window and the k-th frequency band, where $t \in \{1, 2, \cdots, T\}$ and $k \in \{1, 2, \cdots, 8\}$. T represents the total number of 1-second time windows (i.e., the length of epoch in seconds). The relative power $x_{t,k}$ is defined as the ratio of the power spectral density within the frequency band over the total power. Based on this definition, $x_{t,k}$ must be within the interval [0, 1].

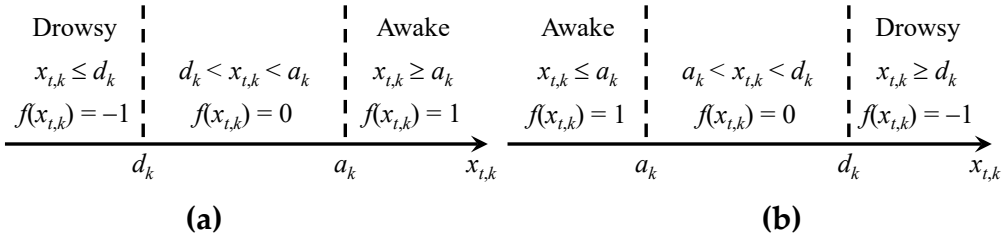

Figure 1: Nonlinear function $f_k(\bullet)$ in (1) is defined for two different cases. (a) The power spectral density $x_{t,k}$ takes a larger value at the awake state than the drowsy state. (b) The power spectral density $x_{t,k}$ takes a smaller value at the awake state than the drowsy state

Once the relative power values are known, we map them to eight features:

$$y_k = \sum_{t=1}^{T} f_k(x_{t,k}) \quad (k = 1, 2, \cdots, 8) \tag{1}$$

Where the definition of the nonlinear function $f_k(\bullet)$ is shown in Figure 1 or two different cases. At each 1-second time window, the power spectral density $x_{t,k}$ is compared against two threshold values, a_k and d_k , representing the awake and drowsy states respectively. The feature y_k in (1) is calculated by *counting* the comparison results over the entire epoch.

In practice, the threshold values a_k and d_k must be carefully determined with consideration of two important facts. First, these threshold values are expected to vary from subject to subject. Second, a_k and d_k may even vary from session to session for the same subject. Hence, because the EEG data is non-stationary, we

cannot simply set a_k and d_k to two constant values. Instead, they must be *adaptively* adjusted in real time.

To adaptively find the optimal values for a_k and d_k, we consider two training data sets for awake and drowsy states respectively associated with a given subject: $\{x_{A,t,k}^{(n)}; t = 1, 2, \cdots, T; k = 1, 2, \cdots, 8; n = 1, 2, \cdots, N_A\}$ and $\{x_{D,t,k}^{(n)}; t = 1, 2, \cdots, T; k = 1, 2, \cdots, 8; n = 1, 2, \cdots, N_D\}$, where N_A and N_D denote the number of epochs in these two data sets respectively. We calculate the average relative power values for awake and drowsy states:

$$\tilde{x}_{A,k} = \frac{1}{N_A \cdot T} \sum_{n=1}^{N_A} \sum_{t=1}^{T} x_{A,t,k} \tag{2}$$

$$\tilde{x}_{D,k} = \frac{1}{N_D \cdot T} \sum_{n=1}^{N_D} \sum_{t=1}^{T} x_{D,t,k} \tag{3}$$

The difference between $\tilde{x_{A,k}}$ and $\tilde{x_{D,k}}$ is:

$$\delta_k = \tilde{x}_{A,k} - \tilde{x}_{D,k} \tag{4}$$

Note that Eq. (2)-(4) are all calculated by using the historical training data.

For our application of drowsiness detection, we can generally assume that a subject is awake as a new session starts. Therefore, the EEG signal (say, N_S

epochs) at the beginning of a session represents the awake state and can be used to determine the session-specific threshold value a_k:

$$a_k = \frac{1}{N_S \cdot T} \sum_{n=1}^{N_S} \sum_{t=1}^{T} x_{S,t,k} \tag{5}$$

For different sessions, a_k estimated by (5) is often different, implying that we are adaptively determining the threshold value to accommodate non-stationary EEG data.

On the other hand, we do not have any training data in the drowsy state for a new session. To determine the threshold value d_k, we assume that δ_k in (4) is unchanged over different sessions, yielding:

$$d_k = a_k - \delta_k \tag{6}$$

Since a_k varies from session to session, d_k is adaptively updated over sessions.

Once all threshold values $\{(a_k, d_k); k = 1, 2, \cdots, 8\}$ are determined by using (5)-(6), we can extract the features $\{y_k; k = 1, 2, \cdots, 8\}$ for eight frequency bands based on (1). These features will be further used for drowsiness detection, as will be discussed in the following sub-section.

2.3 Drowsiness Detection

Once the features $\{y_k; k = 1, 2, \cdots, 8\}$ are extracted, we need to train an efficient classifier to distinguish awake and drowsy states. In this proposed method, we adopt the algorithm of support vector machine (SVM) [44] drowsiness detection. In particular, a linear SVM aims to learn a discriminant function:

$$g(y) = \beta^T y + c = \begin{cases} \geq 0 & (Awake) \\ < 0 & (Drowsy) \end{cases}, \tag{7}$$

where $\mathbf{y} = [y_1\ y_2\ \cdots\ y_8]^T$, $\boldsymbol{\beta} \in \Re^8$, and $c \in \Re$ denote the unknown coefficients that should be determined by the training process.

Given two training data sets $\{f_{A,k}{}^{(n)}; k = 1, 2, \cdots, 8; n = 1, 2, \cdots, N_A\}$ and $\{y_{D,k}{}^{(n)}; k = 1, 2, \cdots, 8; n = 1, 2, \cdots, N_D\}$ associated with the awake and drowsy states respectively, the unknown coefficients $\boldsymbol{\beta}$ and c in (7) can be determined by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\beta}, c, \xi_A^{(n)}, \xi_D^{(n)}} \quad & \boldsymbol{\beta}^T\boldsymbol{\beta} + \lambda \cdot \sum_{n=1}^{N_A} \xi_A^{(n)} + \lambda \cdot \sum_{n=1}^{N_D} \xi_D^{(n)} \\ \text{S.T.} \quad & \boldsymbol{\beta}^T \mathbf{y}_A^{(n)} + c \geq 1 - \xi_A^{(n)} && \left(n = 1, \cdots, N_A\right) \\ & \xi_A^{(n)} \geq 0 && \left(n = 1, \cdots, N_A\right), \\ & -\boldsymbol{\beta}^T \mathbf{y}_D^{(n)} - c \geq 1 - \xi_D^{(n)} && \left(n = 1, \cdots, N_D\right) \\ & \xi_D^{(n)} \geq 0 && \left(n = 1, \cdots, N_D\right) \end{aligned} \tag{8}$$

where $\lambda \in \Re$ is a hyper-parameter that can be determined by cross-validation [44].

With the classifier in (7), we are able to distinguish awake and drowsy states for each epoch. The efficacy of our proposed drowsiness detection will be demonstrated by the experimental results in the next section.

2.4 Experimental Results

In this section, we demonstrate the efficacy of the proposed method by using the EEG recordings from three different publicly-available sleep datasets. In the following sub-sections, we describe the dataset used to test and validate the proposed drowsiness-detection method.

2.4.1 MIT-BIH Polysomnographic Database

This database is an assortment of multiple recordings which utilizes multiple physiologic indicators during sleep. In this work, EEG recording of 16 subjects from the MIT-BIH polysomnographic database. In this database, we consider the 30-second epochs that are labeled by professionals as "*Awake Stage*" (AS) or "*Stage I*" (S1) where S1 stands for the drowsy stage. Since we are interested in detecting the transition from the awake state to the drowsy state, we only consider the adjacent AS and S1 epochs where the S1 epoch immediately follows

the AS epoch. The MIT-BIH database includes a number of AS epochs between other sleep stages and they are excluded from our experiment.

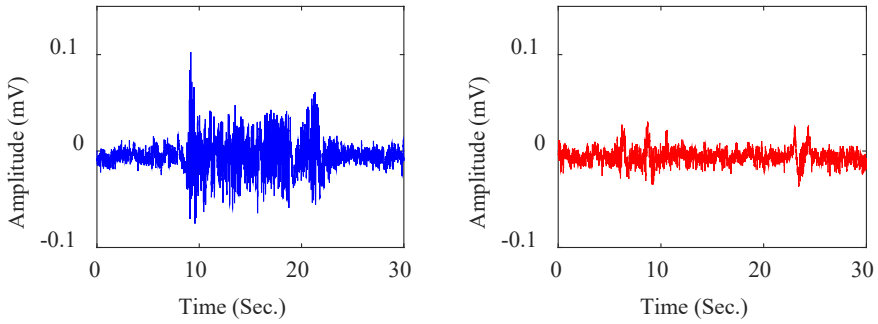

Figure 2: EEG signals before pre-processing for two 30-second epochs of the 14th subject: (a) awake and (b) drowsy.

As an example, Figure 2 shows the raw EEG signals for two 30-second epochs of the 14th subject from the data set. It is straightforward to observe that the signal power is substantially different between the awake and drowsy states.

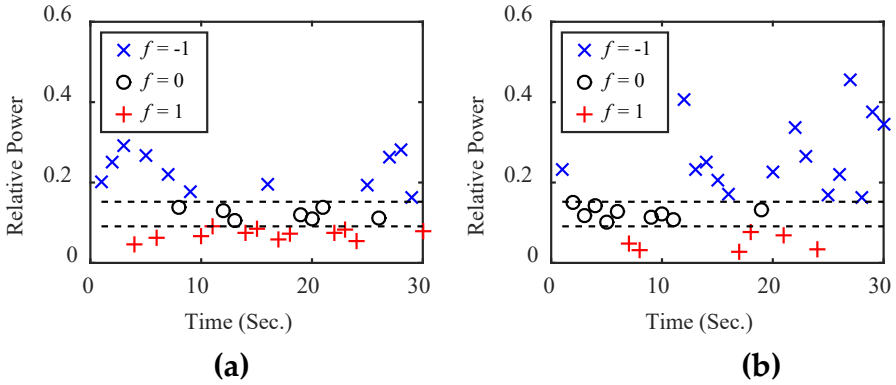

Figure 3: The relative power is shown for each 1-second time window over the theta band (4–7 Hz) of two 30-second epochs of the 14th subject: (a) awake and (b) drowsy. The dashed lines represent two threshold values ak and dk defined by (5)-(6).

While Figure 2 does not plot the signal power for each frequency band, Figure 3 shows the relative power for each 1-second time window over the theta band (4–7 Hz) of these two 30-second epochs. Note that the relative power over [4Hz, 7Hz] increases, as the awake state transits to the drowsy state. Hence, the nonlinear function $f_k(\bullet)$ in (1) follows the definition in Figure 1 for the theta band. Even though the relative power calculated from each 1-second window seems noisy, our proposed counter-based approach in (1) is expected to average out the noise and, hence, reliably extract the features of interest.

Figure 4 further shows the relative power for each 1-second time window over the low-alpha band (8–9 Hz) of the same epochs. Different from Figure 3, the relative power over [8Hz, 9Hz] decreases, as the awake state transits to the drowsy state. Hence, the nonlinear function $f_k(\bullet)$ in (1) follows the definition in Figure 1(a) for the low-alpha band.

For testing and comparison purposes, we have implemented two different drowsiness detection algorithms: (i) a conventional linear classifier [21] that takes the relative power values $\{x_{t,k};\ t = 1, 2, \cdots, 30;\ k = 1, 2, \cdots, 8\}$ of all 1-second time windows over all frequency bands as the input features, and (ii) the proposed approach that extracts the input features by using (1). Note that our proposed

approach relies on the nonlinear functions $\{f_k(\bullet); k = 1, 2, \cdots, 8\}$ for feature extraction. Hence, even though the discriminant function in (7) trained by SVM is linear, the proposed classifier is nonlinear with respect to the relative power values $\{x_{t,k}; t = 1, 2, \cdots, 30; k = 1, 2, \cdots, 8\}$.

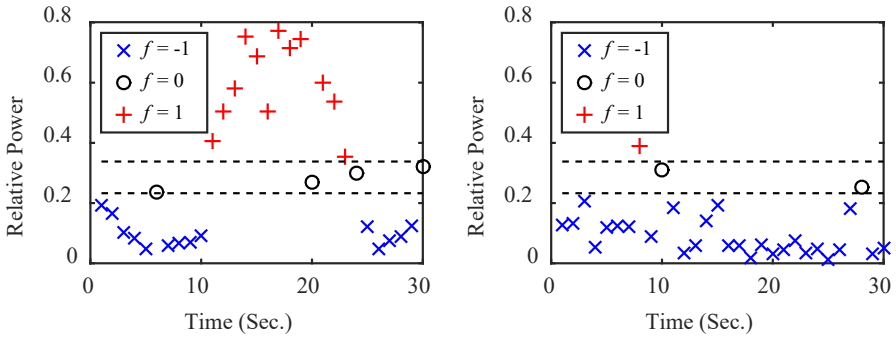

Figure 4: The relative power is shown for each 1-second time window over the low-alpha band (8–9 Hz) of two 30-second epochs of the 14th subject: (a) awake and (b) drowsy. The dashed lines represent two threshold values a_k and d_k defined by (5)-(6).

Table 1 compares the detection accuracy for the aforementioned two approaches. For all 16 subjects, the proposed method offers superior detection accuracy over the conventional approach. On average, the proposed method is able to improve the detection accuracy from 70.62% to 83.36%. In this example, the nonlinear functions $\{f_k(\bullet); k = 1, 2, \cdots, 8\}$ in (1) converts the relative power to three possible values $\{-1, 0, 1\}$, removes the random fluctuations associated with power spectral densities and, consequently, results in high detection accuracy.

Table 1: Drowsiness detection accuracy using MIT-BIH Dataset 30-seconds

Subject	Conventional	Proposed
1	73.23%	86.67%
2	70.19%	80.00%
3	68.25%	78.75%
4	74.50%	81.67%
5	65.46%	84.12%
6	74.23%	83.00%
7	78.63%	80.00%
8	62.04%	70.00%
9	72.95%	87.78%
10	67.86%	80.00%
11	68.89%	72.00%
12	84.79%	100.00%
13	69.45%	83.33%
14	65.90%	80.00%
15	74.62%	88.57%
16	69.44%	80.00%
Average	70.62%	83.36%

2.4.2 University College Dublin Sleep Apnea Database

This database is commonly known to permit automatic semantic classification of sleep apnea. The database consists of 25 full overnight polysomnograms which are simultaneously three-channel Holter ECG that is composed of adult subjects that have suspected sleep-disordered breathing. Subjects are selected over a 6-month period from the sleep disorder clinic at St Vincent's University Hospital, Dublin. The subjects were all above 18 years of age

and were all diagnosis for obstructive sleep apnea, central sleep apnea or primary snoring. No subject was selected with known cardiac disease, autonomic dysfunction, and medication known to interfere with heart rate.

Table 2: Drowsiness detection accuracy using UCDDB Dataset 30-seconds

Subject	Conventional	Proposed
1	56.67%	73.18%
2	64.52%	84.50%
3	52.83%	69.38%
4	59.70%	72.94%
5	55.16%	72.00%
6	60.02%	79.33%
7	53.29%	63.89%
8	63.23%	80.77%
9	59.63%	85.88%
10	81.06%	100.00%
11	66.31%	86.07%
12	57.16%	75.67%
13	55.03%	59.74%
14	53.75%	67.50%
15	70.00%	90.00%
16	64.40%	71.11%
17	56.87%	73.64%
18	59.26%	78.33%
19	64.59%	96.15%
20	55.38%	71.75%
21	55.61%	65.38%
22	63.25%	77.33%
23	59.81%	76.32%
24	62.31%	70.83%
25	56.64%	71.92%
Average	**59.23%**	**75.05%**

In the University College Dublin Sleep Apnea database, we consider the 30-second epochs that are labeled by professionals as "*Awake Stage*" (AS) or "*Stage I*" (S1) where S1 stands for the drowsy stage. Since we are interested in detecting the transition from the awake state to the drowsy state, we only consider the adjacent AS and S1 epochs where the S1 epoch immediately follows the AS epoch. Similarly to the MIT-BIH database, the University College Dublin Sleep Apnea database includes a number of AS epochs between other sleep stages and they are excluded from our experiment.

Table 2 compares the detection accuracy for the previously-discussed two approaches. For all the 25 subjects, the proposed method offers superior detection accuracy over the conventional approach. On average, the proposed method is able to improve the detection accuracy from 59.23% to 75.05%.

2.4.3 DREAMS Subjects Database

This database from University of MONS - TCTS Laboratory and Universit´e Libre de Bruxelles - CHU de Charleroi Sleep Laboratory contains recordings of 20 subjects. Data collected were acquired in a sleep laboratory of a Belgium hospital using a digital 32-channel polygraph. At least two EOG channels (P8-A1, P18-A1), three EEG channels (CZ-A1 or C3-A1, FP1-A1 and O1-A1) and one submental

EMG channel were recorded. In this work, we used a single EEG channel, CZ-A1 [45]. The database including sleep-staging scores for every subject according to R&K and AASM classification of sleep stages. It presents a strong and very flexible analysis that can be utilized for empirical research.

In the DREAMS database, we stack 6 consecutive 5-seconds epochs together to form 30-seconds epochs. The 5-seconds epochs are labeled by professionals as "*Awake Stage*" (AS) or "*Stage I*" (S1) where S1 stands for the drowsy stage. Since we are interested in detecting the transition from the awake state to the drowsy state, we only consider the adjacent AS and S1 epochs where the S1 epoch immediately follows the AS epoch. Similarly to the MIT-BIH database, the DREAMS database includes a number of AS epochs between other sleep stages and they are excluded from our experiment.

Table 3 compares the detection accuracy between the conventional and the proposed algorithms, which are previously discussed. For all the 20 subjects, the proposed method offers superior detection accuracy over the conventional approach. On average, the proposed method is able to improve the detection accuracy from 64.44% to 74.26%.

Table 3: Drowsiness detection accuracy using DREAMS Dataset 30-seconds

Subject	Conventional	Proposed
1	65.20%	76.67%
2	61.87%	68.82%
3	64.23%	80.00%
4	67.45%	77.78%
5	74.04%	86.00%
6	62.07%	76.00%
7	60.25%	55.00%
8	62.97%	70.67%
9	63.01%	74.00%
10	62.13%	74.00%
11	66.90%	71.67%
12	64.02%	74.38%
13	62.80%	76.84%
14	60.86%	70.00%
15	62.58%	74.55%
16	69.48%	77.78%
17	68.49%	73.33%
18	68.94%	83.33%
19	61.16%	70.00%
20	62.60%	75.00%
Average	**64.44%**	**74.26%**

3. Fixed-Point Implementation

Construction and mining industries suffer from challenging environment. One of the most challenging factors is network reliability. Recently, a number of emerging applications, such as portable wearable devices for processing electrocardiography (ECG), have posed a strong need to implement machine learning algorithms with application-specific integrated circuits (ASICs) due to the following reasons:

Small latency: The response of a machine learning engine must be sufficiently fast for many real-time applications such as vital sign monitoring [46] and real-time drowsiness detection systems [31, 47-54]. In these cases, an on-chip machine learning engine must be implemented to locally process the data to ensure fast response time, instead of transmitting the data to an external device (e.g., cloud server) for remote processing.

Low power: To facilitate a portable/wearable device to continuously operate over a long time without recharging the battery, its power consumption must be minimized. Especially for the applications where power consumption is highly constrained (e.g., less than 10 μW), it is necessary to design an ASIC circuit, instead of relying on a general-purpose microprocessor, to meet the tight power budget.

To maximally reduce the power consumption of on-chip machine learning for portable/implantable applications, fixed-point arithmetic, instead of floating-point arithmetic, must be adopted to implement machine learning algorithms and the word length for fixed-point computing must be aggressively minimized. While fixed-point arithmetic has been extensively studied for digital signal processing [55-58], it is rarely explored for emerging machine learning tasks. Historically, most machine learning algorithms are only developed and validated by their software implementations (e.g., MATLAB, C++, etc.) based on double-precision floating-point arithmetic. It remains an open question how to appropriately map these algorithms to a low-power ASIC circuit implemented with fixed-point arithmetic.

In this chapter, we consider a case study of three well-known machine learning algorithms for binary classification: linear discriminant analysis (LDA), support vector machine (SVM), and logistic regression (LR) [59]. We will demonstrate that rounding error incurred from fixed-point arithmetic can significantly distort the classification output, if they are not appropriately modeled and incorporated into the classifier training process. In other words, the conventional classification algorithms developed for double-procession floating-

point operation must be "re-designed" in order to be made "robust" to rounding error. Similar "numerical" issues have been extensively studied in many other application domains. For instance, pivoting is an important technique for Gaussian elimination that is needed to mitigate the numerical error of a linear solver [60]. The fundamental question here is how we can improve the robustness of LDA, SVM, and LR to make them suitable for on-chip low-power implementation.

Towards this goal, we propose to reformulate the training process of LDA, SVM, and LR as mixed-integer programming (MIP) problems with consideration of the non-idealities (i.e., rounding and overflow) posed by fixed-point arithmetic. Our re-designed LDA algorithm is referred to as LDA-FP in this section. A novel branch-and-bound method is developed to find the globally optimal classification boundary of LDA-FP. With our re-designed training algorithm, LDA-FP is made maximally robust to the non-idealities associated with fixed-point arithmetic and, hence, can be efficiently implemented with extremely small word length for on-chip low-power operation.

On the other hand, we refer to the re-designed SVM and LR algorithms as SVM-FP and LR-FP, respectively. Similar to LDA-FP, the branch-and-bound

method is applied to solve SVM-FP and LR-FP and find their optimal classification boundaries. In addition, several efficient heuristics (e.g., piecewise linear approximation [61]) are adopted to make the branch-and-bound solver computationally efficient.

Our proposed algorithms (i.e., LDA-FP, SVM-FP and LR-FP) are validated for the application of EEG-based drowsiness detection [21, 29]. As will be demonstrated by our experimental results in Section 3.5, the proposed approach is able to reduce the word length by up to 1.67× compared to the conventional techniques, without surrendering any classification accuracy.

3.1 Classification Algorithms

In this work, we consider a case study of three well-known machine learning algorithms for binary classification: linear discriminant analysis (LDA), support vector machine (SVM), and logistic regression (LR).

3.1.1 LDA

LDA is a machine learning algorithm that has been extensively applied to a large number of binary classification problems[59]. To illustrate the basic idea of LDA, we consider two sets of training data $\{\mathbf{x}_A^{(n)}; n = 1, 2, \ldots, N_A\}$ and $\{\mathbf{x}_B^{(n)}; n = 1, 2, \ldots, N_B\}$ corresponding to two classes, where $\mathbf{x}_A^{(n)} = [x_{A,1}^{(n)}\ x_{A,2}^{(n)}\ \ldots\ x_{A,M}^{(n)}]^T$ and $\mathbf{x}_B^{(n)} =$

$[x_{B,1}^{(n)}\ x_{B,2}^{(n)}\ \ldots\ x_{B,M}^{(n)}]^T$ are the feature vectors of the nth sampling point from the class A and B respectively, M stands for the number of features, and N_A and N_B represent the numbers of sampling points for these two classes respectively. LDA aims to find an optimal projection direction $\mathbf{w} \in \Re^M$ so that the two classes are maximally separated after projecting the feature vector $\mathbf{x}$ along the direction $\mathbf{w}$ [59].

Towards this goal, we first quantitatively calculate the between-class scatter matrix $\mathbf{S}_{B,\mathbf{x}} \in \Re^{M\times M}$ and the within-class scatter matrix $\mathbf{S}_{W,\mathbf{x}} \in \Re^{M\times M}$ for the feature vector $\mathbf{x}$ [59]:

$$\mathbf{S}_{B,\mathbf{x}} = \left(\mu_A - \mu_B\right)\cdot\left(\mu_A - \mu_B\right)^T \tag{9}$$

$$\mathbf{S}_{W,\mathbf{x}} = \frac{1}{2}\cdot\left(\Sigma_A + \Sigma_B\right) \tag{10}$$

where $\boldsymbol{\mu}_A \in \Re^M$ and $\boldsymbol{\mu}_B \in \Re^M$ stand for the mean vectors:

$$\mu_A = \frac{1}{N_A}\cdot\sum_{n=1}^{N_A}\mathbf{x}_A^{(n)} \tag{11}$$

$$\mu_B = \frac{1}{N_B}\cdot\sum_{n=1}^{N_B}\mathbf{x}_B^{(n)} \tag{12}$$

and $\boldsymbol{\Sigma}_A \in \Re^{M\times M}$ and $\boldsymbol{\Sigma}_B \in \Re^{M\times M}$ represent the covariance matrices:

$$\Sigma_A = \frac{1}{N_A}\cdot\sum_{n=1}^{N_A}\left(\mathbf{x}_A^{(n)} - \mu_A\right)\cdot\left(\mathbf{x}_A^{(n)} - \mu_A\right)^T \tag{13}$$

$$\Sigma_B = \frac{1}{N_B} \cdot \sum_{n=1}^{N_B} \left(\mathbf{x}_B^{(n)} - \mu_B \right) \cdot \left(\mathbf{x}_B^{(n)} - \mu_B \right)^T . \tag{14}$$

Projecting the feature vector **x** along the direction **w** yields:

$$y = w^T x \tag{15}$$

Since the projection result y is equal to a linear combination of all features weighted by **w**, the projection direction **w** is often referred to as the *weight vector* for LDA in the literature [59].

It is straightforward to verify that the between-class scatter $S_{B,y} \in \Re$ and the within-class scatter $S_{W,y} \in \Re$ for the projection result y can be written as [59]:

$$S_{B,y} = \mathbf{w}^T \cdot \mathbf{S}_{B,\mathbf{x}} \cdot \mathbf{w} \tag{16}$$

$$S_{W,y} = \mathbf{w}^T \cdot \mathbf{S}_{W,\mathbf{x}} \cdot \mathbf{w} . \tag{17}$$

In order to maximally separate the two classes, the between-class scatter $S_{B,y}$ should be maximized, while the within-class scatter $S_{W,y}$ should be minimized. Hence, we can formulate the following optimization to minimize the ratio between $S_{W,y}$ and $S_{B,y}$ [59]:

$$\min_{\mathbf{w}} \quad \frac{S_{W,y}}{S_{B,y}} = \frac{\mathbf{w}^T \cdot \mathbf{S}_{W,\mathbf{x}} \cdot \mathbf{w}}{\mathbf{w}^T \cdot \mathbf{S}_{B,\mathbf{x}} \cdot \mathbf{w}} = \frac{\mathbf{w}^T \cdot \mathbf{S}_{W,\mathbf{x}} \cdot \mathbf{w}}{\left[\left(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B \right)^T \cdot \mathbf{w} \right]^2} . \tag{18}$$

There are two important clarifications that should be made regarding the optimization formulation in (18). First, it is easy to verify that the optimal solution

$\mathbf{w}$ of (18) is not unique, since the cost function in (18) is dependent on the direction of $\mathbf{w}$ only and it is independent of the length of $\mathbf{w}$. Namely, if $\mathbf{w}$ is an optimal solution of (18), multiplying $\mathbf{w}$ by any non-zero scalar λ (i.e., $\lambda \cdot \mathbf{w}$) does not change the cost function and, hence, is also an optimal solution of (18). When LDA is implemented, an additional constraint is often added to specify the length of the vector $\mathbf{w}$ (e.g., $||\mathbf{w}||_2 = 1$ where $||\bullet||_2$ denotes the L_2-norm of a vector) so that the optimal solution becomes unique.

Second, even though the optimization formulation in (18) is not convex, it can be mapped to a generalized eigenvalue problem and solved both efficiently (i.e., with low computational cost) and robustly (i.e., with guaranteed global optimum) [59]. Assuming that the within-class scatter matrix $\mathbf{S}_{W,\mathbf{x}}$ is full-rank, the optimal $\mathbf{w}$ of (18) can be proven to be [59]:

$$\mathbf{w} \propto \mathbf{S}_{W,\mathbf{x}}^{-1} \cdot \left(\mu_A - \mu_B\right). \tag{19}$$

The weight vector $\mathbf{w}$ in (19) can be normalized by its length if we want to keep the final solution with unit length. Once $\mathbf{w}$ is determined, the following linear decision boundary can be constructed for binary classification:

$$\mathbf{w}^T \cdot \mathbf{x} - \mathbf{w}^T \cdot \frac{\mu_A + \mu_B}{2} = \begin{cases} \geq 0 & \left(Class\ A\right) \\ < 0 & \left(Class\ \mathrm{B}\right) \end{cases} \tag{20}$$

where **x** denotes the feature vector of a sampling point that should be classified.

3.1.2 SVM

SVM has been widely used for pattern recognition and data classification [62]. It aims to find the optimal classification boundary that maximizes the margin. Similar to LDA, we consider the training data $\{\mathbf{x}_A^{(n)}; n = 1, 2, \ldots, N_A\}$ and $\{\mathbf{x}_B^{(n)}; n = 1, 2, \ldots, N_B\}$ for two classes. When a linear SVM is applied, the classification boundary is represented by $\mathbf{w}^T \cdot \mathbf{x} + b = 0$, where **w** and b should be determined from the training data. The optimal **w** and b can be solved from the following optimization problem to maximize the classification margin [59]:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_A^{(n)},\xi_B^{(n)}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda \cdot \sum_{n=1}^{N_A} \xi_A^{(n)} + \lambda \cdot \sum_{n=1}^{N_B} \xi_B^{(n)} & \\ \text{S.T.} \quad & \mathbf{w}^T \cdot \mathbf{x}_A^{(n)} + b \geq 1 - \xi_A^{(n)} & \left(n = 1, \cdots, N_A\right) \\ & \xi_A^{(n)} \geq 0 & \left(n = 1, \cdots, N_A\right) \\ & -\mathbf{w}^T \cdot \mathbf{x}_B^{(n)} - b \geq 1 - \xi_B^{(n)} & \left(n = 1, \cdots, N_B\right) \\ & \xi_B^{(n)} \geq 0 & \left(n = 1, \cdots, N_B\right) \end{aligned} \tag{21}$$

where $\{\xi_A^{(n)}; n = 1, 2, \ldots, N_A\}$ and $\{\xi_B^{(n)}; n = 1, 2, \ldots, N_B\}$ represent the slack variables for measuring classification error, and λ is a parameter that can be determined by cross-validation [59].

3.1.3 LR

Logistic Regression (LR) has been used for classification problems in many practical applications. Considering the training data $\{\mathbf{x}_A^{(n)}; n = 1, 2, \ldots, N_A\}$ and $\{\mathbf{x}_B^{(n)}; n = 1, 2, \ldots, N_B\}$ for two classes, we define the following model to describe the probability for each sampling point $\mathbf{x}$ to belong to a given class [59]:

$$p(A \mid \mathbf{x}) = \frac{1}{1 + \exp\left(-\mathbf{w}^T \cdot \mathbf{x} - b\right)} \tag{22}$$

$$p(B \mid \mathbf{x}) = 1 - \frac{1}{1 + \exp\left(-\mathbf{w}^T \cdot \mathbf{x} - b\right)} = \frac{\exp\left(-\mathbf{w}^T \cdot \mathbf{x} - b\right)}{1 + \exp\left(-\mathbf{w}^T \cdot \mathbf{x} - b\right)} \tag{23}$$

where $\mathbf{w}$ and b define the classifier. The optimal values of $\mathbf{w}$ and b can be found by maximum likelihood estimation based on the training data:

$$\min_{\mathbf{w},b} \quad -\sum_{n=1}^{N_A} \log\left[\frac{1}{1 + \exp\left(-\mathbf{w}^T \cdot \mathbf{x}_A^{(n)} - b\right)}\right] - \sum_{n=1}^{N_B} \log\left[\frac{\exp\left(-\mathbf{w}^T \cdot \mathbf{x}_B^{(n)} - b\right)}{1 + \exp\left(-\mathbf{w}^T \cdot \mathbf{x}_B^{(n)} - b\right)}\right] + \rho \cdot \mathbf{w}^T \mathbf{w} \tag{24}$$

where $\rho \cdot \mathbf{w}^T \mathbf{w}$ is the additional regularization term to avoid over-fitting and the optimal value of ρ should be determined by cross-validation.

3.1.4 Fixed-Point Classifiers

The conventional classification algorithms are able to find the optimal decision boundary, assuming that all computations can be performed without rounding error. In practice, once the classifier is implemented with fixed-point

arithmetic, rounding error can substantially bias the decision boundary and, hence, distort the classification output. In this case, the classification boundary determined by a conventional machine learning algorithm is no longer optimal.

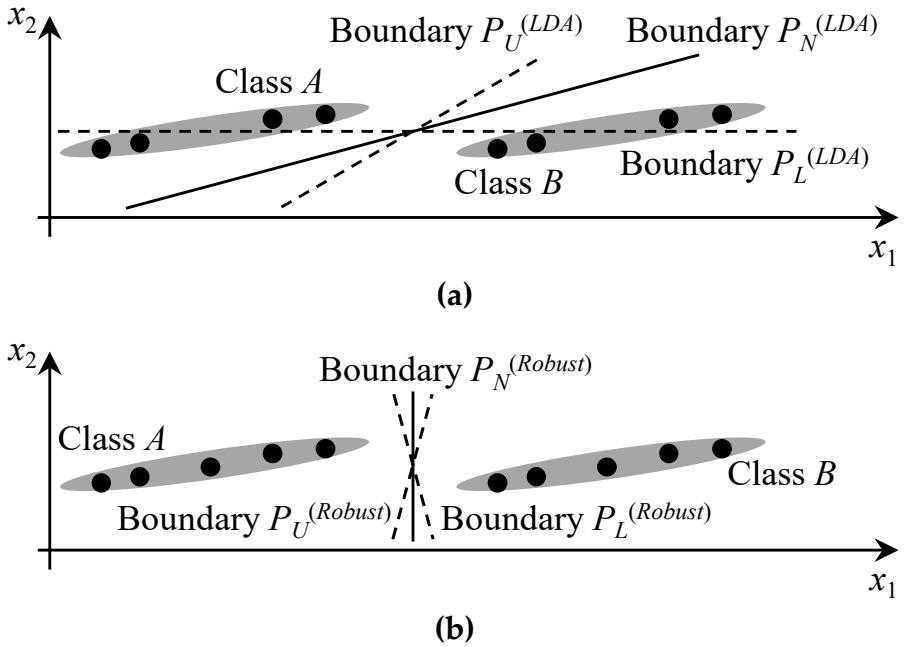

Figure 5: 2-D LDA classification example: (a) The optimal boundary $P_N^{(LDA)}$ determined by LDA is highly sensitive to rounding error. (b) A robust boundary $P_N^{(Robust)}$ can be found and it is insensitive to rounding error.

To understand the reason, we consider a simple 2-D example for LDA. The objective is to determine an optimal linear boundary to separate Class *A* and Class *B*. By applying LDA, we find the optimal boundary $P_N^{(LDA)}$ shown in Figure 5(a). In addition to this nominal boundary $P_N^{(LDA)}$, Figure 5 (a) further plots two perturbed boundaries, $P_L^{(LDA)}$ and $P_U^{(LDA)}$, due to rounding error. Note that if $P_N^{(LDA)}$ is rounded to $P_L^{(LDA)}$, a large classification error is expected. On the other hand,

Figure 5 (b) shows a robust boundary $P_N^{(Robust)}$ that is less sensitive to rounding error than $P_N^{(LDA)}$. Even if $P_N^{(Robust)}$ is perturbed to $P_L^{(Robust)}$ or $P_U^{(Robust)}$, the classification error remains negligible.

The aforementioned discussion reveals an important observation that the conventional machine learning algorithms may result in large classification error, since they do not explicitly consider the rounding error during classifier training. It, in turn, motivates us to re-design these conventional algorithms in order to generate a robust classifier that is least sensitive to rounding error.

3.2 FP-LDA

3.2.1 Problem Formulation

The non-idealities (i.e., rounding and overflow) posed by fixed-point arithmetic can be broadly classified into two different categories: (i) the non-idealities associated with the feature vector **x**, and (ii) the non-idealities associated with the weight vector **w**. For the feature vector **x**, all features in **x** can be carefully scaled to avoid overflow. In addition, the feature vector **x** should be rounded to its fixed-point representation, before the training data is used to learn the classifier. As such, the training algorithm can easily take into account the rounding error associated with **x**. In other words, the conventional LDA algorithm is able to address the non-idealities for the feature vector **x** appropriately.

On the other hand, mitigating the non-idealities of the weight vector **w** is not trivial. If we simply follow the conventional LDA algorithm, **w** is determined by (**19**) and then normalized and rounded to its fixed-point representation. In this case, large classification error may occur, as is demonstrated by the simple example in Figure 6. Hence, our focus of this section is to derive a new LDA-FP algorithm to solve the optimal **w** that is suitable for fixed-point implementation.

Without loss of generality, we assume that a fixed-point number is represented in the format of *QK.F* based on two's complement [58]. It has *K* integer bits (including the sign bit) and F fractional bits, as shown in Figure 6. The total word length is $K + F$. In this section, we further assume that all fixed-point operations in the classifier are implemented with the same format *QK.F*. In practice, it is possible to further optimize the word length for each individual operation. For instance, different elements $\{w_m; \mathrm{m} = 1, 2, \ldots, M\}$ of the weight vector **w** can be assigned with different word lengths. However, since we mainly focus on classifier training in this section, the problem of word length optimization will be considered as a separate topic for our future research.

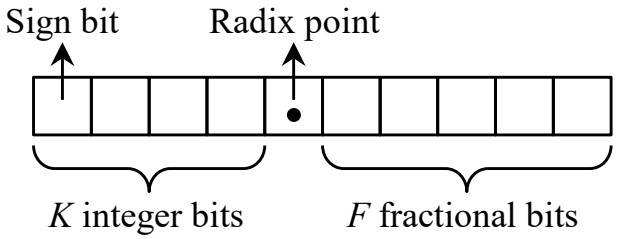

Figure 6: An example is shown for the fixed-point format of *QK.F* based on two's complement.

Given the fixed-point representation *QK.F* shown in Figure 6, we must consider a number of important constraints, when deriving our proposed optimization formulation for LDA-FP.

Rounding error: Since the weight vector **w** is represented in the format of *QK.F*, each element of **w**, w_m where $m \in \{1, 2, \ldots, M\}$, can only take a finite number of possible values:

$$w_m \in \Omega \quad (m = 1, 2, \cdots, M) \tag{25}$$

where Ω stands for the set of all possible values that can be represented by *QK.F*.

Overflow: While the overflow of the feature vector **x** can be easily prevented by appropriately scaling **x** during a pre-processing step, we must carefully constrain the weight vector **w** so that calculating the projection $y = \mathbf{w}^T \cdot \mathbf{x}$ in (**15**) does not result in an overflow. To this end, we statistically model the feature vector **x** as a multivariate Gaussian distribution:

$$\mathbf{x} \sim \begin{cases} Gauss\left(\Sigma_A, \Sigma_A\right) & \left(Class\ A\right) \\ Gauss\left(\Sigma_B, \Sigma_B\right) & \left(Class\ \mathrm{B}\right) \end{cases}. \tag{26}$$

Note that the probability distribution of **x** depends on the class that **x** belongs to. Such a Gaussian model has been widely applied in the literature by the machine-learning community [59].

Based on (26), each multiplication $w_m \cdot x_m$, where $m \in \{1, 2, \ldots, M\}$, yields a Gaussian distribution:

$$w_m \cdot x_m \sim \begin{cases} Gauss\left(w_m \cdot \mu_{A,m}, w_m^2 \cdot \sigma_{A,m}^2\right) & \left(Class\ A\right) \\ Gauss\left(w_m \cdot \mu_{B,m}, w_m^2 \cdot \sigma_{B,m}^2\right) & \left(Class\ \mathrm{B}\right) \end{cases} \tag{27}$$

where $\mu_{A,m}$ and $\mu_{B,m}$ are the m^{th} elements of $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ respectively, and $\sigma_{A,m}^2$ and $\sigma_{B,m}^2$ are the m^{th} diagonal elements of $\boldsymbol{\Sigma}_A$ and $\boldsymbol{\Sigma}_B$ respectively. With the Gaussian distributions in (27), we can statistically define a confidence interval for $w_m \cdot x_m$:

$$\beta = \Phi^{-1}\left(0.5 + 0.5 \cdot \rho\right) \tag{28}$$

$$\begin{matrix} \left[w_m \mu_{A,m} - \beta \cdot \left|w_m\right| \cdot \sigma_{A,m},\ w_m \mu_{A,m} + \beta \cdot \left|w_m\right| \cdot \sigma_{A,m}\right] & \left(Class\ A\right) \\ \left[w_m \mu_{B,m} - \beta \cdot \left|w_m\right| \cdot \sigma_{B,m},\ w_m \mu_{B,m} + \beta \cdot \left|w_m\right| \cdot \sigma_{B,m}\right] & \left(Class\ B\right) \end{matrix} \tag{29}$$

where ρ denotes the confidence level and $\Phi^{-1}(\bullet)$ stands for the inverse cumulative distribution function of a standard Gaussian distribution. The confidence level ρ measures the probability for $w_m \cdot x_m$ to take a value within the confidence interval. If ρ is sufficiently large, the confidence interval in (29)

"almost" covers the range of $w_m \cdot x_m$. Hence, the confidence interval in (29) must be within the range of the fixed-point representation *QK.F* to avoid overflow:

$$\begin{aligned} w_m \cdot \mu_{A,m} - \beta \cdot |w_m| \cdot \sigma_{A,m} &\geq -2^{K-1} \\ w_m \cdot \mu_{B,m} - \beta \cdot |w_m| \cdot \sigma_{B,m} &\geq -2^{K-1} \\ w_m \cdot \mu_{A,m} + \beta \cdot |w_m| \cdot \sigma_{A,m} &\leq 2^{K-1} - 2^{-F} \\ w_m \cdot \mu_{B,m} + \beta \cdot |w_m| \cdot \sigma_{B,m} &\leq 2^{K-1} - 2^{-F} \end{aligned} . \tag{30}$$

In addition to the multiplication wm·xm, the projection result y = wT·x is a linear combination of all features with Gaussian distributions and, hence, remains a Gaussian distribution:

$$\mathbf{w}^T \cdot \mathbf{x} \sim \begin{cases} Gauss\left(\mathbf{w}^T \cdot \mu_A, \mathbf{w}^T \cdot \Sigma_A \cdot \mathbf{w}\right) & (Class\ A) \\ Gauss\left(\mathbf{w}^T \cdot \mu_B, \mathbf{w}^T \cdot \Sigma_B \cdot \mathbf{w}\right) & (Class\ B) \end{cases} . \tag{31}$$

Similar to (27)-(30), we can statistically define the confidence interval for *y* and then set up the following constraints to prevent *y* from overflowing:

$$\begin{aligned} \mathbf{w}^T \mu_A - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_A \mathbf{w}} &\geq -2^{K-1} \\ \mathbf{w}^T \mu_B - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_B \mathbf{w}} &\geq -2^{K-1} \\ \mathbf{w}^T \mu_A + \beta \cdot \sqrt{\mathbf{w}^T \Sigma_A \mathbf{w}} &\leq 2^{K-1} - 2^{-F} \\ \mathbf{w}^T \mu_B + \beta \cdot \sqrt{\mathbf{w}^T \Sigma_B \mathbf{w}} &\leq 2^{K-1} - 2^{-F} \end{aligned} . \tag{32}$$

Two important clarifications must be made here. First, we do not need to explicitly set up any overflow constraint for the intermediate sum when calculating the weighted sum $y = \mathbf{w}^T \cdot \mathbf{x}$. As long as the final projection result *y* does

not overflow and the fixed-point numbers are implemented with two's complement based on wrapping, the overflow of the intermediate sum should not introduce any error on y. To intuitively illustrate this important property, we consider a simple example of calculating the summation $y = 3 + 3 - 4$ using $Q3.0$. Since the range of $Q3.0$ is $[-4, 3]$, calculating the intermediate sum $011 + 011 = 110$ results in an overflow. After calculating the final result, however, we get $110 + 100 = 010$. Compared to the correct value of $y = 3 + 3 - 4 = 2$, the final result represented by $Q3.0$ remains correct.

Second, rounding and overflow are closely coupled and must be simultaneously considered in our proposed training process. On one hand, if we focus on rounding without overflow, the classifier may carry extremely large weight values, represented in the format of $QK.F$, to minimize the impact of rounding. Such a classifier, however, is likely to overflow due to its large weight values. On the other hand, if we consider overflow without rounding, the classifier may use extremely small weight values to minimize the probability to overflow. However, these small weight values can be substantially distorted after rounding.

Classification accuracy: To maximize classification accuracy of LDA, the weight vector $\mathbf{w}$ must be optimized to maximally separate the two classes.

Towards this goal, we borrow the cost function in (**18**) to minimize the ratio between the within-class scatter $S_{W,y}$ and the between-class scatter $S_{B,y}$, resulting the following optimization:

$$
\begin{aligned}
\min_{\mathbf{w}} \quad & \frac{\mathbf{w}^T \cdot \mathbf{S}_{W,\mathbf{x}} \cdot \mathbf{w}}{\left[\left(\mu_A - \mu_B\right)^T \cdot \mathbf{w}\right]^2} \\
\text{S.T.} \quad & w_m \cdot \mu_{A,m} - \beta \cdot |w_m| \cdot \sigma_{A,m} \geq -2^{K-1} \qquad (m = 1,2,\cdots,M) \\
& w_m \cdot \mu_{B,m} - \beta \cdot |w_m| \cdot \sigma_{B,m} \geq -2^{K-1} \qquad (m = 1,2,\cdots,M) \\
& w_m \cdot \mu_{A,m} + \beta \cdot |w_m| \cdot \sigma_{A,m} \leq 2^{K-1} - 2^{-F} \qquad (m = 1,2,\cdots,M) \\
& w_m \cdot \mu_{B,m} + \beta \cdot |w_m| \cdot \sigma_{B,m} \leq 2^{K-1} - 2^{-F} \qquad (m = 1,2,\cdots,M) \\
& \mathbf{w}^T \mu_A - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_A \mathbf{w}} \geq -2^{K-1} \\
& \mathbf{w}^T \mu_B - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_B \mathbf{w}} \geq -2^{K-1} \\
& \mathbf{w}^T \mu_A + \beta \cdot \sqrt{\mathbf{w}^T \Sigma_A \mathbf{w}} \leq 2^{K-1} - 2^{-F} \\
& \mathbf{w}^T \mu_B - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_B \mathbf{w}} \leq 2^{K-1} - 2^{-F} \\
& w_m \in \Omega \qquad (m = 1,2,\cdots,M)
\end{aligned}
\quad . \tag{33}
$$

Note that the optimization formulation in (33) represents a mixed integer programming problem [63, 64] that is difficult to solve due to the following two reasons. First, the solution **w** is constrained to a discrete set, instead of a continuous set. Second, the cost function is represented as the ratio of two quadratic functions and, hence, is not convex. To address these challenges, we will further propose a novel branch-and-bound method with several efficient

heuristics to solve (33) and find the global optimum. The details of our proposed solver will be discussed in the next sub-section.

3.2.2 Branch-and-Bound Solver

Branch-and-bound method has been widely used to solve mixed integer programming problems [64]. The key idea is to iteratively partition the optimization variables into sub-intervals. For each sub-interval, the lower and upper bounds of the cost function are estimated in order to efficiently remove the irrelevant sub-intervals that do not contain the optimal solution from the search space. However, when the branch-and-bound method is applied to solve (33), it is not trivial to efficiently estimate the lower and upper bounds for a given sub-interval of $\mathbf{w}$, because the cost function in (33) is the ratio of two quadratic functions and, hence, not convex [63]. Next, we will propose several novel ideas to address this technical challenge so that the optimization problem in (33) can be solved by the branch-and-bound robustly (i.e., with guaranteed global optimum).

Lower bound estimation: We introduce an additional variable:

$$t = \left(\mu_A - \mu_B\right)^T \cdot \mathbf{w} \tag{34}$$

and re-write the cost function in (33) as:

$$\min_{\mathbf{w},t} \quad \frac{\mathbf{w}^T \cdot \mathbf{S}_{W,\mathbf{x}} \cdot \mathbf{w}}{t^2}. \tag{35}$$

Note that both **w** and t are now considered as optimization variables and they should be partitioned into sub-intervals when searching for the optimal solution by the branch-and-bound method.

With a given sub-interval:

$$\begin{gathered} l_{wm} \le w_m \le u_{wm} \quad (m = 1, 2, \cdots, M) \\ l_t \le t \le u_t \end{gathered} \tag{36}$$

where l_{wm} and l_t represent the lower bounds of w_m and t respectively, and u_{wm} and u_t stand for the upper bounds of w_m and t respectively. We can now estimate the lower bound of the cost function by solving the following optimization:

$$\begin{aligned} \min_{\mathbf{w},t} \quad & \frac{1}{\eta} \cdot \mathbf{w}^T \cdot \mathbf{S}_{W,\mathbf{x}} \cdot \mathbf{w} \\ \text{S.T.} \quad & w_m \cdot \mu_{A,m} - \beta \cdot |w_m| \cdot \sigma_{A,m} \ge -2^{K-1} && (m = 1, 2, \cdots, M) \\ & w_m \cdot \mu_{B,m} - \beta \cdot |w_m| \cdot \sigma_{B,m} \ge -2^{K-1} && (m = 1, 2, \cdots, M) \\ & w_m \cdot \mu_{A,m} + \beta \cdot |w_m| \cdot \sigma_{A,m} \le 2^{K-1} - 2^{-F} && (m = 1, 2, \cdots, M) \\ & w_m \cdot \mu_{B,m} + \beta \cdot |w_m| \cdot \sigma_{B,m} \le 2^{K-1} - 2^{-F} && (m = 1, 2, \cdots, M) \\ & \mathbf{w}^T \mu_A - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_A \mathbf{w}} \ge -2^{K-1} \\ & \mathbf{w}^T \mu_B - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_B \mathbf{w}} \ge -2^{K-1} \\ & \mathbf{w}^T \mu_A + \beta \cdot \sqrt{\mathbf{w}^T \Sigma_A \mathbf{w}} \le 2^{K-1} - 2^{-F} \\ & \mathbf{w}^T \mu_B - \beta \cdot \sqrt{\mathbf{w}^T \Sigma_B \mathbf{w}} \le 2^{K-1} - 2^{-F} \\ & t = (\mu_A - \mu_B)^T \cdot \mathbf{w} \\ & l_{wm} \le w_m \le u_{wm} && (m = 1, 2, \cdots, M) \\ & l_t \le t \le u_t \end{aligned} \tag{37}$$

where η is a constant defined by:

$$\eta = \sup_{l_t \le t \le u_t} t^2 . \tag{38}$$

In (38), sup(•) denotes the supremum (i.e., the least upper bound) of a set. Since t^2 is simply a quadratic function, it is easy to calculate the supremum over the interval $l_t \le t \le u_t$ and determine the value of η.

The optimization in (37) is a convex second-order cone programming problem [63] and, hence, can be solved efficiently. Compared to (33), the formulation in (37) is "relaxed" in two different ways. First, the denominator of the cost function is relaxed to the maximum possible value η over the interval $l_t \le t \le u_t$. Second, the vector **w** is no longer constrained to a discrete set; instead, it can take any real value within the sub-interval $\{l_{wm} \le w_m \le u_{wm};\ m = 1, 2, \ldots, M\}$. For this reason, solving (37) yields a lower bound of the cost function of the original mixed integer programming problem for the given sub-interval defined in (36).

Upper bound estimation: Similar to lower bound estimation, we can re-use the optimization formulation in (37) to estimate the upper bound of the cost function with the parameter η set to:

$$\eta = \inf_{l_t \le t \le u_t} t^2 \tag{39}$$

where inf(•) denotes the infimum (i.e., the greatest lower bound) of a set. In this case, the optimization in (37) is again a convex second-order cone programming problem. After solving (37), we further round the solution **w** to the discrete set defined in (25). Next, we substitute the rounded **w** to the cost function in (33). It, in turn, results in an upper bound of the cost function of (33) given the sub-interval defined in (36).

Initial interval size estimation: Since the branch-and-bound method iteratively partitions the optimization variables **w** and t into sub-intervals, we need to determine the initial interval size for **w** and t, before starting the first iteration. Since **w** is represented in the format of $QK.F$ as shown in Figure 6, it must be within the following range:

$$-2^{K-1} \le w_m \le 2^{K-1} - 2^{-F} \quad \left(m = 1, 2, \cdots, M\right). \tag{40}$$

On the other hand, since t is a linear function of **w** as defined in (34), combining (34) and (40) yields:

$$-2^{K-1} \cdot \left\| \mu_A - \mu_B \right\|_1 \le t \le \left(2^{K-1} - 2^{-F}\right) \cdot \left\| \mu_A - \mu_B \right\|_1 \tag{41}$$

where $||\bullet||_1$ denotes the L_1-norm of a vector (i.e., the summation of the absolute values of all elements in the vector).

With the aforementioned heuristics, our proposed branch-and-bound method for solving (**33**) can be summarized by **Algorithm 3.1**. Here, we iteratively shrink the search intervals until they are sufficiently small. During these iterations, the lower and upper bounds of the cost function are estimated to remove the irrelevant intervals and, hence, reduce the search space.

ALGORITHM 3.1. Branch-and-Bound Solver for LDA-FP

---

1. Start from two sets of training data $\{\mathbf{x}_A^{(n)}; n = 1, 2, \ldots, N_A\}$ and $\{\mathbf{x}_B^{(n)}; n = 1, 2, \ldots, N_B\}$ corresponding to two classes, and a given fixed-point format $Q.F$. Round the training data to their fixed-point representations.
2. Calculate $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ by (**11**)-(**12**) and $\mathbf{S}_{W,x}$ by (**10**).
3. Initialize the search interval ξ by (**40**)-(**41**). Estimate the lower bound l_f and upper bound u_f of the cost function for the given interval ξ. Set $\Xi = \{\xi\}$.
4. Select one interval from the set Ξ, and partition it into two sub-intervals. Remove the selected interval from Ξ.
5. For each of these two sub-intervals, estimate the lower bound l_f and upper bound u_f of the cost function. If l_f is no greater than the minimum upper bound over all intervals in the set Ξ, add the current sub-interval to Ξ. Find all intervals in Ξ for which the lower bounds are greater than u_f, and remove them from Ξ.
6. If the sizes of all intervals in the set Ξ are sufficiently small, stop iteration. Otherwise, go to Step 4.

3.3 FP-SVM

Following the idea of LDA-FP described in the previous section, we can derive the mathematical formulation for SVM-FP. As mentioned in Section 3.1.2, SVM aims to find the optimal classification boundary with maximum margin. Combining the SVM formulation in (**21**), the rounding error in (25) and the

overflow constraints in (30) and (32) yields the following constrained optimization problem for SVM-FP:

$$
\begin{aligned}
\min_{\mathbf{w},b,\xi_A^{(n)},\xi_B^{(n)}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda\cdot\sum_{n=1}^{N_A}\xi_A^{(n)} + \lambda\cdot\sum_{n=1}^{N_B}\xi_B^{(n)} & \\
\text{S.T.} \quad & \mathbf{w}^T\cdot\mathbf{x}_A^{(n)} + b \ge 1-\xi_A^{(n)} & \left(n=1,\cdots,N_A\right)\\
& \xi_A^{(n)} \ge 0 & \left(n=1,\cdots,N_A\right)\\
& -\mathbf{w}^T\cdot\mathbf{x}_B^{(n)} - b \ge 1-\xi_B^{(n)} & \left(n=1,\cdots,N_B\right)\\
& \xi_B^{(n)} \ge 0 & \left(n=1,\cdots,N_B\right)\\
& w_m\cdot\mu_{A,m} - \beta\cdot\left|w_m\right|\cdot\sigma_{A,m} \ge -2^{K-1} & \left(m=1,2,\cdots,M\right)\\
& w_m\cdot\mu_{B,m} - \beta\cdot\left|w_m\right|\cdot\sigma_{B,m} \ge -2^{K-1} & \left(m=1,2,\cdots,M\right)\\
& w_m\cdot\mu_{A,m} + \beta\cdot\left|w_m\right|\cdot\sigma_{A,m} \le 2^{K-1}-2^{-F} & \left(m=1,2,\cdots,M\right)\\
& w_m\cdot\mu_{B,m} + \beta\cdot\left|w_m\right|\cdot\sigma_{B,m} \le 2^{K-1}-2^{-F} & \left(m=1,2,\cdots,M\right)\\
& \mathbf{w}^T\mu_A - \beta\cdot\sqrt{\mathbf{w}^T\Sigma_A\mathbf{w}} \ge -2^{K-1} & \\
& \mathbf{w}^T\mu_B - \beta\cdot\sqrt{\mathbf{w}^T\Sigma_B\mathbf{w}} \ge -2^{K-1} & \\
& \mathbf{w}^T\mu_A + \beta\cdot\sqrt{\mathbf{w}^T\Sigma_A\mathbf{w}} \le 2^{K-1}-2^{-F} & \\
& \mathbf{w}^T\mu_B - \beta\cdot\sqrt{\mathbf{w}^T\Sigma_B\mathbf{w}} \le 2^{K-1}-2^{-F} & \\
& w_m \in \Omega & \left(m=1,2,\cdots,M\right)
\end{aligned}
\tag{42}
$$

Eq. (42) represents a mixed integer programming problem. Unlike the problem in (33) where the cost function is non-convex, the cost and constraint functions in (42) are all convex except the last integer constraint. In this case, we can solve (42) by using the standard branch-and-bound algorithm implemented by an existing toolbox such as MOSEK [65].

3.4 FP-LR

The goal of LR is to find the parameters **w** and b that has the smallest possible deviance between the training data and the predicted values. We re-write the objective function in (**24**) by using the log-sum-exp (LSE) function $\Re^K \rightarrow \Re$:

$$\mathrm{lse}\left(x_1,\cdots,x_K\right)=\log\left(\mathrm{e}^{x_1}+\cdots+\mathrm{e}^{x_K}\right). \tag{43}$$

The function lse(•) is convex and can be interpreted as an analytic-differentiable approximation of the max function [63]. Using the bivariate LSE function, the objective function in (**24**) can be reformulated as:

$$\min_{\mathbf{w},b}\ \sum_{n=1}^{N_A}\left[\mathrm{lse}\left(0,-\mathbf{w}^T\cdot\mathbf{x}_A^{(n)}-b\right)\right]+\sum_{n=1}^{N_B}\left[\mathrm{lse}\left(0,-\mathbf{w}^T\cdot\mathbf{x}_B^{(n)}-b\right)-\left(-\mathbf{w}^T\cdot\mathbf{x}_B^{(n)}-b\right)\right]+\rho\cdot\mathbf{w}^T\mathbf{w} \tag{44}$$

Since the function lse(•) is convex, it can be approximated by a piecewise-linear (PWL) function to any degree of accuracy if the PWL function contains a sufficiently large number of pieces. For a bivariate function lse(y_1, y_2), a constructive algorithm has been proposed in the literature to find the optimal PWL approximation [61]. Given an integer $R \geq 2$, we solve the following optimization problem to find the optimal R-term PWL function to approximate the bivariate function lse(y_1, y_2):

$$\begin{aligned}\min_{u_{r1},u_{r2},h_r}\quad & \sup_{y_1,y_2}\left\{\mathrm{lse}\left(y_1,y_2\right)-\max_{r\in\{1,\cdots,R\}}\left\{u_{r1}y_1+u_{r2}y_2+h_r\right\}\right\}\\ \mathrm{S.T.}\quad & \mathrm{lse}\left(y_1,y_2\right)>\max_{r\in\{1,\cdots,R\}}\left\{u_{r1}y_1+u_{r2}y_2+h_r\right\}\qquad \forall y_1,\forall y_2\end{aligned} \tag{45}$$

where the optimization variables are $\{(u_{r1}, u_{r2}, h_r); r = 1, 2, ..., R\}$. The problem in (**45**) can be efficiently solved by an iterative algorithm described in [61]. Once the parameters $\{(u_{r1}, u_{r2}, h_r); r = 1, 2, ..., R\}$ are determined, the optimal R-term PWL approximation is:

$$\mathrm{pwl}(y_1, y_2) = \max\{u_{11}y_1 + u_{12}y_2 + h_1, \cdots, u_{R1}y_1 + u_{R2}y_2 + h_R\}. \tag{46}$$

Now, we can substitute the LSE function in (44) with the optimal R-term PWL approximation in (46):

$$\begin{aligned}\min_{\mathbf{w},b} \quad & \sum_{n=1}^{N_A}\left[\max\left\{-u_{12}\mathbf{w}^T\mathbf{x}_A^{(n)} - u_{12}b + h_1, \cdots, -u_{R2}\mathbf{w}^T\mathbf{x}_A^{(n)} - u_{R2}b + h_R\right\}\right] \\ & + \sum_{n=1}^{N_B}\left[\max\left\{-u_{12}\mathbf{w}^T\mathbf{x}_B^{(n)} - u_{12}b + h_1, \cdots, -u_{R2}\mathbf{w}^T\mathbf{x}_B^{(n)} - u_{R2}b + h_R\right\} - \left(-\mathbf{w}^T\cdot\mathbf{x}_B^{(n)} - b\right)\right] + \rho\cdot\mathbf{w}^T\mathbf{w}\end{aligned} \tag{47}$$

Finally, combining the LR formulation in (47), the rounding error in (25) and the overflow constraints in (30) and (32) yields the following constrained optimization problem for LR-FP:

$$
\begin{aligned}
&\min_{\mathbf{w},b} \quad \sum_{n=1}^{N_A}\left[\max\left\{-u_{12}\mathbf{w}^T\mathbf{x}_A^{(n)}-u_{12}b+h_1,\cdots,-u_{R2}\mathbf{w}^T\mathbf{x}_A^{(n)}-u_{R2}b+h_R\right\}\right] \\
&\qquad +\sum_{n=1}^{N_B}\left[\max\left\{-u_{12}\mathbf{w}^T\mathbf{x}_B^{(n)}-u_{12}b+h_1,\cdots,-u_{R2}\mathbf{w}^T\mathbf{x}_B^{(n)}-u_{R2}b+h_R\right\}-\left(-\mathbf{w}^T\cdot\mathbf{x}_B^{(n)}-b\right)\right]+\rho\cdot\mathbf{w}^T\mathbf{w} \\
&\qquad w_m\cdot\mu_{A,m}-\beta\cdot|w_m|\cdot\sigma_{A,m}\geq -2^{K-1} \qquad (m=1,2,\cdots,M) \\
&\qquad w_m\cdot\mu_{B,m}-\beta\cdot|w_m|\cdot\sigma_{B,m}\geq -2^{K-1} \qquad (m=1,2,\cdots,M) \\
&\qquad w_m\cdot\mu_{A,m}+\beta\cdot|w_m|\cdot\sigma_{A,m}\leq 2^{K-1}-2^{-F} \qquad (m=1,2,\cdots,M) \\
&\qquad w_m\cdot\mu_{B,m}+\beta\cdot|w_m|\cdot\sigma_{B,m}\leq 2^{K-1}-2^{-F} \qquad (m=1,2,\cdots,M) \\
&\text{S.T.} \quad \mathbf{w}^T\mu_A-\beta\cdot\sqrt{\mathbf{w}^T\Sigma_A\mathbf{w}}\geq -2^{K-1} \\
&\qquad \mathbf{w}^T\mu_B-\beta\cdot\sqrt{\mathbf{w}^T\Sigma_B\mathbf{w}}\geq -2^{K-1} \\
&\qquad \mathbf{w}^T\mu_A+\beta\cdot\sqrt{\mathbf{w}^T\Sigma_A\mathbf{w}}\leq 2^{K-1}-2^{-F} \\
&\qquad \mathbf{w}^T\mu_B-\beta\cdot\sqrt{\mathbf{w}^T\Sigma_B\mathbf{w}}\leq 2^{K-1}-2^{-F} \\
&\qquad w_m\in\Omega \qquad (m=1,2,\cdots,M)
\end{aligned}
\qquad .(48)
$$

Eq. (48) represents a mixed integer programming problem where the cost and constraint functions are all convex except the last integer constraint. It can be solved using the standard branch-and-bound algorithm implemented by an existing toolbox such as MOSEK [65]. In our implementation, CVX, a MATLAB-based convex solver [66], is used to first formulate the optimization problem in (48) and then the formulation is passed to the MOSEK solver.

3.5 Experimental Results

In this section, a drowsiness-detection application is presented to demonstrate the efficacy of our proposed algorithms. When training fixed-point classifiers, both the input data and the weight values are appropriately scaled and

rounded as fixed-point numbers. For testing and comparison purposes, the conventional algorithms, namely LDA-R, SVM-R and LR-R, are implemented where the weight vector **w** is first solved by (19), (21), and (24) respectively, and then rounded to its fixed-point representation. On the other hand, the proposed LDA-FP, SVM-FP and LR-FP methods solve the weight vector **w** from (33), (42) and (48) respectively. In addition, we show the accuracy of LDA, SVM and LR with double-precision floating-point arithmetic. All numerical values, including both the input data and the weight values, are represented as floating-point numbers, when training floating-point LDA, SVM and LR. All numerical experiments run on a 2.9GHz Linux server with 8GB memory.

The need of a reliable detection system of drowsiness is arising every day and it is considered as the leading objective in the development of new Advanced Driver Assistance systems [21]. There are several measures that can be used to assess the vigilance level, including behavioral measures, vehicle-based measures, and physiological measures [12]. In the literature, physiological measures are considered as the most reliable metrics for drowsiness detection, especially, across different individuals.

In this example, EEG recording of 16 human subjects from the MIT-BIH Polysomnographic Database is used [67]. All participants are male and range between 32 and 56 years old. The database in total contains over 80 hours of polysomnographic recording for ECG, EEG and respiration signals. The sleep stages are scored by experts. The epochs labeled as "Awake Stage" (AS) and "Stage I" (S1) have been used in this example where S1 corresponds to the drowsy stage. The AS epochs used in this section are only those precede S1, and the S1 epochs used are only those subsequent to AS. All other epochs are excluded in our experiment. There are 5435 trials where each trail contains a 10-second EEG segment. In total, 35 features are extracted from these EEG signals [21, 29] and used to distinguish the two classes: AS and S1.

Table 4, Table 5, and Table 6 show the classification error and runtime for different methods. Similar to the previous example, the classification error is estimated by using 5-fold cross-validation. Several important observations can be made from these results. First, LDA-FP is more accurate than LDA-R with the same word length. To achieve the same classification accuracy, the required word length is 5-bit for LDA-R and 3-bit for LDA-FP.

Second, when the word length is sufficiently large, both the fixed-point and floating-point classifiers achieve similar classification error. In other words, further increasing the word length for our fixed-point classifiers would not improve the classification accuracy any more. The detection accuracy in this section is similar to that reported in the literature [21, 29].

Third, LDA, SVM and LR offer similar classification accuracy in this example. However, the runtime of LDA-FP is substantially less than that of SVM-FP and LR-FP, when the word length is sufficiently large. Here, the optimization formulations of both SVM-FP and LR-FP involve a large number of constraint functions and, therefore, are more computationally expensive to solve than LDA-FP, similar to what we observed in the previous example.

Table 4: Classification error and runtime using LDA-R, LDA-FP and LDA

Word Length (QK.F)		LDA-R		LDA-FP		LDA
K	F	Error	Runtime (Sec)	Error	Runtime (Sec)	Error
1	2	49.78%	0.06	34.83%	91	31.37%
1	3	49.36%	0.06	32.84%	207	
1	4	34.80%	0.06	31.37%	419	
1	5	32.21%	0.06	31.53%	18	
1	6	32.22%	0.06	30.98%	28	
1	7	30.48%	0.06	31.08%	19	

Table 5: Classification error and runtime using SVM-R, SVM-FP and SVM

Word Length (QK.F)		SVM-R		SVM-FP		SVM
K	F	Error	Runtime (Sec)	Error	Runtime (Sec)	Error
1	2	50.00%	1.24	50.00%	13	31.51%
1	3	50.00%	1.21	36.30%	1857	
1	4	34.41%	1.44	32.19%	1485	
1	5	32.53%	1.33	32.30%	1448	
1	6	32.01%	1.24	32.00%	1056	
1	7	31.85%	1.19	31.76%	662	

Table 6: Classification error and runtime using LR-R, LR-FP and LR

Word Length (QK.F)		LR-R		LR-FP		LR
K	F	Error	Runtime (Sec)	Error	Runtime (Sec)	Error
1	2	50.00%	49.05	34.23%	9971	31.77%
1	3	50.00%	45.90	32.64%	79152	
1	4	34.69%	48.29	32.11%	185551	
1	5	32.31%	50.29	32.72%	199892	
1	6	31.67%	44.85	32.47%	75917	
1	7	31.74%	44.73	32.47%	19818	

4. Instantaneous Drowsiness Detection

In most round the clock industries such as military, construction, maritime, mining, manufacturing, oil & gas, and transportation, work in shifts is popular to maintain operations over 24 hours [68]. The pressure to have continuous operations demands that workers maintain wakefulness throughout the night, which may lead to drowsiness-related accidents. Implementing instantaneous drowsiness-detection systems may help reduce the risk of such accidents by alerting workers in a timely manner. Moreover, additional safety measures, which can be activated automatically when a drowsy state is detected, can be employed to prevent any dangerous situation through ensuring more alertness during work. Fatigue amongst industrial workers has sometimes led to inaccuracy judgment, causing significant injuries amongst these workers [69]. The environment and community have also been victims of industrial workers' fatigue. For example, the oil & gas industry has caused much damage to the environment due to numerous instances in fatigue amongst workers. Long working hours by most drivers in the oil & gas industry is responsible for most accidents. It is the responsibility of these companies to ensure they implement effective fatigue control measures to guarantee health and safety standards for all their workers.

An instantaneous drowsiness detection system has numerous advantages, which include the ability of the system to facilitate quick response time. Despite the fact that there is significant variability in the response time between the drivers, the fact that the warning alerts are instantaneous improves the performance of the driver. The purpose of real-time drowsiness detection is to provide information to drivers that is compatible with their ability to safely operate the vehicle at that particular time; therefore, providing immediate alertness is essential for driving capacity. Instantaneous drowsiness detection system provides better feedback and performance in terms of improved alertness and reduced drowsiness.

In Chapter 2, we proposed a real-time drowsiness detection algorithm. The proposed algorithm adopts a cumulative counter to extract important features from 8 different frequency bands: the delta band (1–3 Hz), the theta band (4–7 Hz), the low-alpha band (8–9 Hz), the high-alpha band (10–12Hz), the low-beta band (13–17 Hz), the high-beta band (18–30 Hz), the low-gamma band (31–40 Hz), and the high-gamma band (41–50 Hz). The proposed method provided superior performance over other conventional methods in the literature, when evaluated on sleep dataset composed of 30-seconds trials. However, to fulfill the strong need

for an instantaneous drowsiness detection system, a drowsiness detection system needs to detect drowsiness using short-time windows (shorter than 30 seconds).

In this chapter, we propose an instantaneous EEG-based drowsiness detection algorithm trained and evaluated at 5-second trials. There exists a gap in past studies since no prior work in the literature has accomplished the goal of how to define the right features for instantaneous detection. Based on this observation, we propose to use CNN to automatically "discover" the important features for instantaneous drowsiness detection. The proposed method employs CNN-based classifier using frequency-specific activity over time extracted from a single-channel EEG signal.

We used an openly available sleep dataset from 20 healthy young adults with 5-seconds labeled trials. The average power of EEG signal is extracted using 1-second time windows over eight frequency bands: the delta band (1–3 Hz), the theta band (4–7 Hz), the low-alpha band (8–9 Hz), the high-alpha band (10–12Hz), the low-beta band (13–17 Hz), the high-beta band (18–30 Hz), the low-gamma band (31–40 Hz), and the high-gamma band (41–50 Hz). Then, we construct two-dimensional (2D) input signals of frequency-specific activity over 1-second time windows. Next, we feed the constructed input signals to CNN-based classifier

with a custom and efficient architecture optimized to detect drowsiness using 5-second trials with 5-fold cross-validation approach. In this work, we employ a unique CNN architecture composed of four convolution layers, each followed by a rectified linear unit (ReLU), a fully-connected layer, and a softmax layer. Our experimental results demonstrate that our CNN-based drowsiness detection system is capable of detecting drowsiness in short-time windows with higher accuracy (75%) compared to conventional methods (65%) and counter-based method (71%).

Our proposed drowsiness detection approach is composed of three major components: (i) signal pre-processing and feature extraction, (ii) constructing input signal, and (iii) CNN architecture. The reminder of this chapter is organized as follows. Section 4.1 presents an introduction to CNN, while Section 4.2 describes the proposed CNN-based drowsiness-detection algorithm and then demonstrates its efficacy by experimental examples in Section 4.3.

4.1 Convolutional Neural Network (CNN)

Human brain neurons abstraction occurs as a result of feedforward neural network (FNN). A neutral network is a system of unified synthetic "neurons" that exchange messages together. Single hidden layer feedforward neural networks

perform universal approximators function due to their precise measurements [70]. Weighted connections provide multiple inputs and outputs in feedforward neural networks nodes connection. The relationships have numeric weights that are adjusted during the training process for the correctly trained network and will respond when presented with a recognized a pattern or image.

Equation 49 illustrates feedforward neural networks model. Network weights are represented as w^T, whereas x and b represent network weights, inputs and biases consecutively.

$$y = f(w^T x + b) \tag{49}$$

Equation 10 shows the forward propagation in a single node where x_i and y_i represent the input and the output respectively.

$$y_i = \sum_i x_i w_i + b \tag{50}$$

The inputs in a standard feedforward neural network propagate through neurons that are fully connected to all other neurons in the previous hidden layers, and each neuron from the previous hidden layer is also connected to all the neurons from the preceding hidden layers. The neurons of the same layer function independently from each other, and no connection exists between the neurons in

the hidden layer. Figure 7 illustrates the structure of a feedforward neural network.

Using the fully-connected feedforward neural network can sometimes prove inefficient and difficult when used with signals of large inputs like images due to its intrinsic properties. For example, a small image of 200x200x3 would require training 120,000 weights for each neuron in the second layer. Obviously, this fully-connected approach is not only computationally demanding, but it would be prone to overfitting.

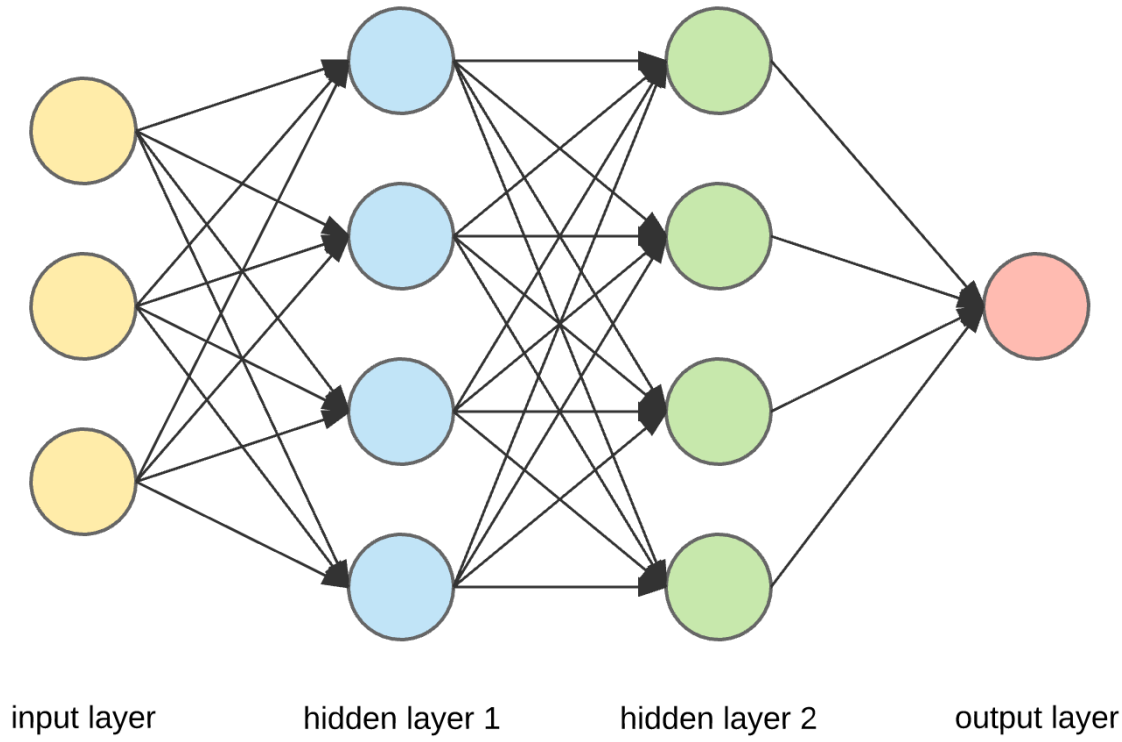

Figure 7: An example of fully connected feedforward neural networks

Convolutional neural network is another type of feedforward neural network that is well suited for computer vision applications. The convolution network limits the architecture of the deep network to reduce the number of parameters to learn, which was inspired by how human vision works [71]. CNN contains some special features that are easily not found in any other type of network. Biological visual cortex contains simple and complex cells necessary in shaping neural networks. Visual fields sub-regions are responsible for activation of these cells. Receptive fields are the term used to define these sub-regions [72]. The connection of these neural networks enables machine learning models characterized with unique coated architectural design.

CNN is an effective recognition algorithm used broadly to process images and recognize patterns with numerous features like easy structures, fewer training restrictions, and flexibility when compared to any other network with similar concealed units, making it the best detection system with numerous benefits [73]. Another major benefits of CNN is the ability to avoid hand-designed input features, but at the same time CNN makes it possible to include, inside the network, high-level knowledge that is directly related to the problem [74].

Although CNN does not require hand-designed filters, the topology of the overall network is still an empirical choice and highly depends on the application.

CNN contains numerous convolutional layers and is designed to utilize 2D structural image input or any other aspect with its input such as speech or EEG signals [73]. Unlike the neurons in a fully-connected layer, the neurons in a convolutional layer are connected to a small and local region of the preceding layers. The convolutional layers arrange neurons in three dimensions, and each layer has a height, width, and depth as shown in Figure 8.

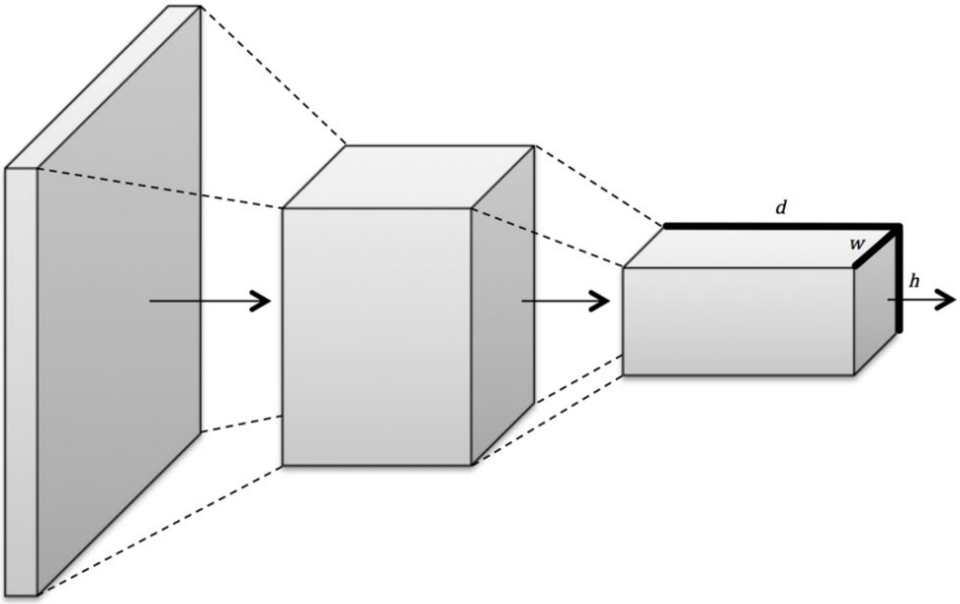

Figure 8: Convolutional layers arrange neurons in three dimensions, so layers have width, height, and depth [71]

Including multiple hidden layers into a convolutional network allows it to learn complex patterns from the data. In CNN, filters are used as feature detectors.

Each convoluted layer is responsible for learning the features detected in the preceding layer. The filters convolve around the input image and shift depending on the stride size. The kernels of the convoluted layers are learnt from the training data. Figure 9 shows two filters – one for detecting horizontal lines, and the other for detecting vertical lines – as they convolve around an input image. The result of the convolution process is considered feature maps. Figure 9 shows the feature maps on the right side.

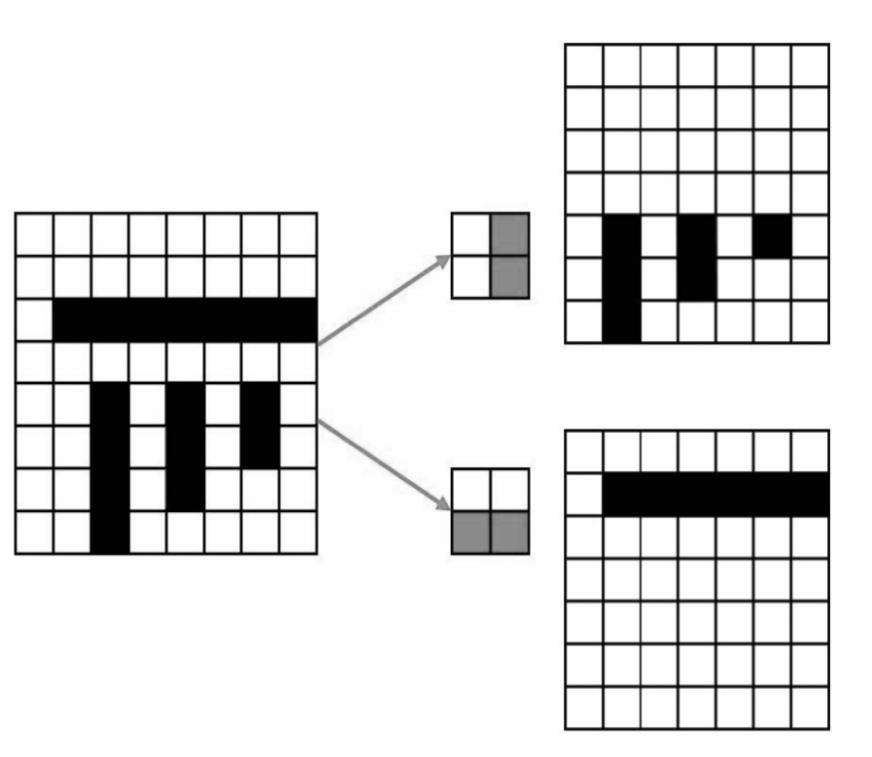

Figure 9: An example of filters used for detecting horizontal and vertical lines

Feature maps can be described mathematically by denoting the k^{th} feature map in layer m as m_k:

$$m_{ij}^{k} = f\left(\left(W * x\right)_{ij} + b^{k}\right) \tag{51}$$

where W represents the filter weights value and b_k is the bias of the neurons in the feature map [71].

Equation 51 describes how the filter operates in a single feature map. However, CNN filters are applied on all feature maps generated by a specific layer. CNN detect patterns based on multiple feature maps. For example, to detect faces at a particular layer in CNN, we need to accumulate three feature maps: one for eyes, one for noses, and one for mouths. Therefore, the CNN makes a decision about the existence of a face by combining the evidence collected from multiple feature maps with the appropriate features (two eyes, a nose, and a mouth). Additionally, if the input image has multiple channels, e.g. an image with red, blue, and green (RGB) color channels, then multiple slices need to be considered in the input volume as shown in Figure 10.

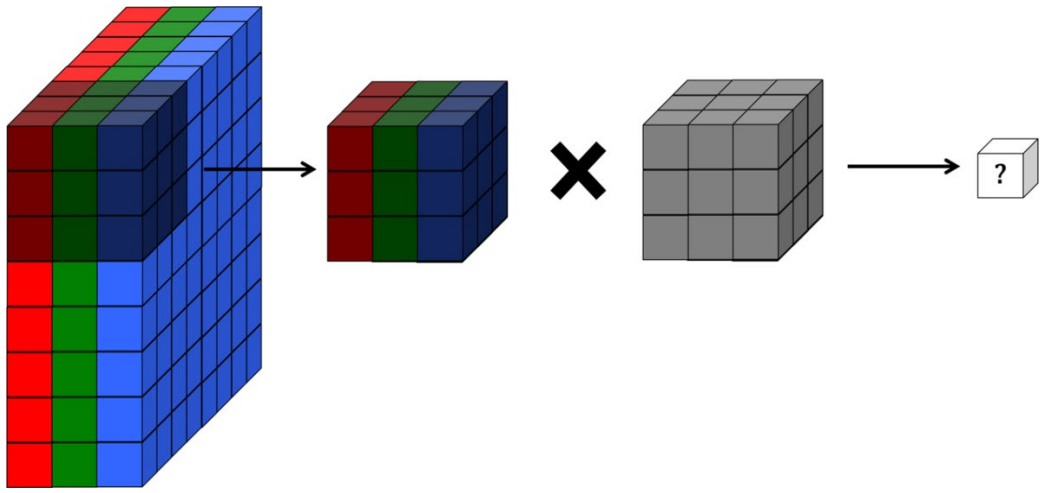

Figure 10: Representing a full-color RGB image as a volume and applying a volumetric convolutional filter [71]

In deep CNN, the first convolution layer detects simple features such as edges and lines. Complex features such as faces, animals, or any other features are extracted from high-level layers. Convolution layers are followed by activation layers, which are used to explore the non-linearities in patterns. Various activation functions have been used in CNN, such as: sigmoid, tanh, and rectified linear unit (ReLU). Selecting the type of the activation functions depends on the application and is usually done empirically.

The sigmoid neuron: This neuron calculates the output between the range [0, 1]. The sigmoid neuron outputs 0 when the logit is small and increases to 1 as the logit get larger. As shown in Figure 11 below, the sigmoid neuron assumes an S-shape and is expressed as [71]:

$$f(z) = \frac{1}{1+e^{-z}} \tag{52}$$

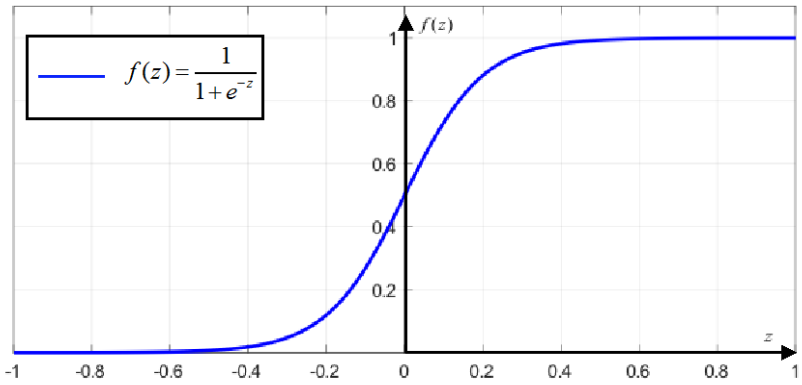

Figure 11: Plot of the sigmoid activation function

The tanh neuron: This neuron is comparable to the sigmoid neuron. It also used a similar s-shaped non-linearity as shown in Figure 12 below. However, tanh neurons are bound to range [−1, 1]. Many applications preferred tanh neurons over the sigmoid neurons because the output of tanh neurons is zero-centered. The tanh neurons are written as follows [71]:

$$f(z) = \tanh(z) \tag{53}$$

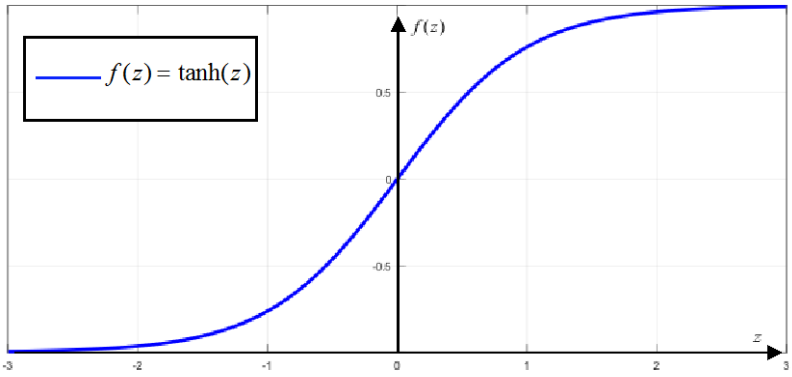

Figure 12: Plot of the tanh activation function

The rectified linear unit: ReLU utilizes a dissimilar type of non-linearity when compared to sigmoid and tanh neurons. The ReLU outputs is unbounded [0, ∞], and non-zero centered. Figure 13 below shows a plot of the rectifier activation function described by:

$$f(z) = \max(0, z) \tag{54}$$

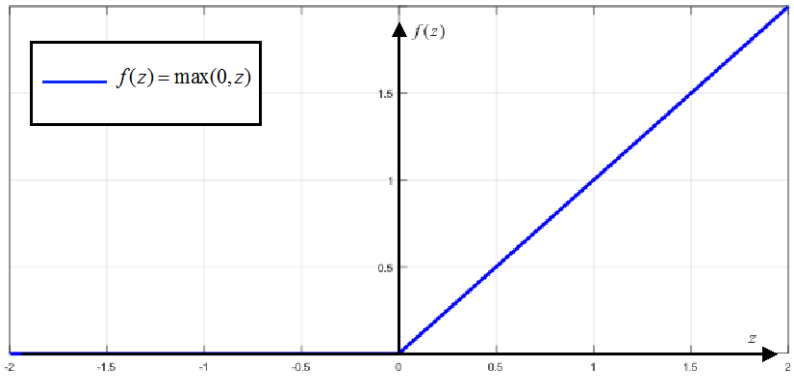

Figure 13: Plot of the ReLU activation function

The ReLU has grown to become a common option of neuron stimulation in various applications, especially computer-vision applications. The ReLU has numerous benefits and needs less optimization compared to other activation functions since it is more resilient to saturation, ignition, and disappearing gradients. ReLU neurons also have a tendency to congregate more quickly, and calculate more rapidly when compared to tanh units. The ReLU are also scale-invariant as $\max(0, ax) = a \max(0, x)$ and do not require normalizing the input

data [75]. The ReLU is naturally plausible as it is one-sided in comparison to the anti-symmetry of the tanh neurons. Additionally, ReLU can output a value of 0, which helps in implementing sparse activation functions [76].

The last layer of CNN is often a softmax activation function, particularly the classification applications. The softmax activation is usually referred to as a normalized exponential and represents the general form of the logistic sigmoid function for multi-class problems [77]. The softmax activation function can be expressed as follows [71]:

$$y_i = \frac{e^{z_i}}{\sum_j e^{z_j}}. \tag{55}$$

In what follows, we will develop a CNN-based approach for instantaneous drowsiness detection.

4.2 CNN-based Drowsiness Detection Algorithm

Our proposed CNN-based drowsiness detection approach is composed of three major components: (i) signal pre-processing, (ii) feature extraction, and (iii) CNN training.

4.2.1 Signal Pre-processing

In this work, the EEG data is collected from a single channel. The proposed pre-processing consists of two major steps. First, the EEG signal is filtered by using a second-order, bi-directional, zero-phase, band-pass Butterworth filter with the passband of 0.5-50Hz. The EEG data is processed in both forward and reverse directions in order to avoid phase distortion [41]. Second, the filtered EEG signal is partitioned into 1-second segments to assure the statistical stationarity required for power density estimation [42]. The average value of each 1-second segment is removed to filter out the DC offset.

4.2.2 Feature Extraction

Given the EEG signal after pre-processing, we measure the power spectral densities of multiple 1-second time windows over eight frequency bands: the delta band (1–3 Hz), the theta band (4–7 Hz), the low-alpha band (8–9 Hz), the high-alpha band (10–12Hz), the low-beta band (13–17 Hz), the high-beta band (18–30 Hz), the low-gamma band (31–40 Hz), and the high-gamma band (41–50 Hz). Past studies findings show that the EEG power over these frequency bands transmits vital information for drowsiness detection [24], [43] [43].

We estimate the relative power $x_{t,k}$ of the t-th time window and the k-th frequency band, where $t \in \{1, 2, \cdots, T\}$ and $k \in \{1, 2, \cdots, 8\}$. T represents the total

number of 1-second time windows (i.e., the length of epoch in seconds). The relative power $x_{t,k}$ is defined as the ratio of the power spectral density within the frequency band over the total power. Based on this definition, $x_{t,k}$ must be within the interval [0, 1].

Given two training data sets for awake and drowsy states respectively associated with a given subject: $\{x_{A,t,k}^{(n)}; t = 1, 2, \cdots, T; k = 1, 2, \cdots, 8; n = 1, 2, \cdots, N_A\}$ and $\{x_{D,t,k}^{(n)}; t = 1, 2, \cdots, T; k = 1, 2, \cdots, 8; n = 1, 2, \cdots, N_D\}$, where N_A and N_D denote the number of epochs in these two data sets respectively, we calculate the average relative power values for awake and drowsy states:

$$\tilde{x}_{A,k} = \frac{1}{N_A \cdot T} \sum_{n=1}^{N_A} \sum_{t=1}^{T} x_{A,t,k} \tag{56}$$

$$\tilde{x}_{D,k} = \frac{1}{N_D \cdot T} \sum_{n=1}^{N_D} \sum_{t=1}^{T} x_{D,t,k} \ . \tag{57}$$

Once the relative power values are acquired, we re-structure the inputs as two-dimensional (2D) signals of frequency-specific activity over time. The architecture of the employed CNN will be discussed next.

4.2.3 CNN Architecture and Training

As shown in Figure 14, we employ a unique CNN architecture composed of four types of layers: input layer, four convolution layers, fully-connected layer, and output layer.

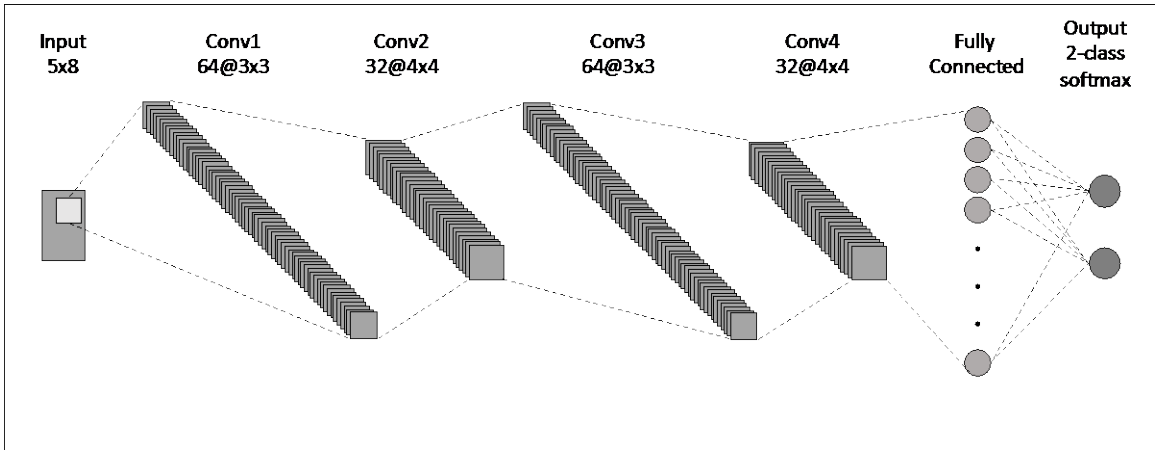

Figure 14: CNN Architecture

The input layer is a 2D stack of frequency-specific activity over time. In particular, we stack five 1-second time windows over eight frequency bands to construct the CNN input as a 5-by-8 matrix. Typical examples of this approach can also be found in recent work in signal processing for speech and acoustics [78-81].

In a convolutional layer, each neuron considers inputs from a small rectangular section of the previous layer and then multiplies the inputs against a localized filter. The same process will be repeated to cover the entire input space to detect a specific pattern based on the localized filter. All neurons sharing the same localized filter compose a feature map. A full convolutional layer is made of multiple feature maps, composed from different localized filters, to detect different kinds of patterns over the entire input space.

In our architecture, we implemented four convolutional layers. The first convolution layer (Conv1) and third convolution layer (Conv3) are composed of 64 filters, each of which has a size of 3. The second convolution layer (Conv2) and fourth convolution layer (Conv4) are composed of 64 filters, each of which has a size of 4. Each convolution layer is followed by a rectified linear unit activation function. All the convolutional layers have zero-padding of size 1 and a stride of 1. Finally, we followed the convolutional layers with a fully-connected layer to take the output of all the neurons from previous layers and feed it to the output layer. A softmax function is used in the output layer with two units to perform the classification.

A convolutional CNN often adopts a number of pooling layers. Each pooling layer takes inputs from a small region of the previous convolutional layer and produces a single output from that region. This pooling layer down-samples the inputs and reduces the training parameters. While it is possible to add a pooling layer after each convolutional layer, our experiments found that adding pooling layers reduces the detection accuracy in our application. The input size used in this work is smaller than the raw signal data. Thus, utilizing pooling layer

for dimension reduction may yield to loss of vital information for detecting drowsiness. Hence, our proposed CNN architecture does not adopt pooling layers.

Various training protocols have been proposed to train a neural network. Stochastic Gradient Descent (SGD) has been proven to be effective in training neural networks. SGD is used to find the weights of the neural network that yield to the minimum of an error function on random mini-batches of the training data. One of the most important parameters of gradient descent is the learning rate, which describes the step size used in gradient descent before recalculating its direction. When learning rate is too small, the training process of the neural network takes longer. On the other hand, when learning rate is too big, we might diverge away from the actual minimum.

In this work, instead of training CNNs with a classical SGD approach, which uses a single fixed learning rate for all the parameters, CNNs are trained using iterative optimization with the adaptive moment estimation (Adam) algorithm [82]. Adam uses different learning rates for different parameters as well as adapting different learning rates during the training process. In this work, the network is initialized with a learning rate of 0.001, the maximum number of

epochs used in training is 30, and the size of the mini-batch for each training iteration is 128.

4.3 Experimental Results

In this section, we present the results of the proposed instantaneous drowsiness-detection method using three publicly-available sleep datasets. In the following sub-sections, we describe the dataset used to test and validate the proposed instantaneous drowsiness-detection method.

For testing and comparison purposes, we have implemented three different drowsiness detection algorithms: (i) a conventional linear classifier [21] that takes the relative power values $\{x_{t,k};\ t = 1, 2, \cdots, 30;\ k = 1, 2, \cdots, 8\}$ of all 1-second time windows over all frequency bands as the input features, (ii) the counter-based approach that was previously proposed and discussed in Chapter 2, and (iii) the proposed CNN-based approach.

4.3.1 MIT-BIH Polysomnographic Database

This database is an assortment of multiple recordings that utilizes multiple physiologic indicators during sleep. EEG recordings of 16 subjects from the MIT-BIH polysomnographic database were used in this work. In this database, we consider the 30-second epochs that are labeled by professionals as "Awake Stage"

(AS) or "Stage I" (S1) where S1 stands for the drowsy stage. Since we are interested in detecting the transition from the awake state to the drowsy state, we only consider the adjacent AS and S1 epochs where the S1 epoch immediately follows the AS epoch. The MIT-BIH database includes a number of AS epochs between other sleep stages, and they are excluded from our experiment.

Table 7: Drowsiness-detection accuracy using MIT-BIH Dataset 5-seconds

Subject	Conventional	Counter-based	Proposed
1	72.0%	78.4%	84.2%
2	70.6%	78.0%	85.0%
3	68.1%	73.9%	81.6%
4	74.2%	79.2%	81.9%
5	65.3%	71.0%	75.2%
6	74.2%	79.5%	85.0%
7	78.1%	82.5%	90.0%
8	60.6%	65.0%	80.0%
9	73.0%	80.5%	89.1%
10	67.3%	73.9%	86.8%
11	68.8%	71.9%	85.9%
12	83.5%	93.3%	100.0%
13	69.2%	77.5%	87.0%
14	65.9%	72.7%	75.8%
15	75.1%	83.9%	83.9%
16	69.4%	74.3%	85.7%
Average	**71.0%**	**77.2%**	**84.8%**

Table 7 compares the detection accuracy for the aforementioned three approaches. The CNN-based approach provides better detection performance

than the other conventional methods. On average, the proposed method is able to improve the detection accuracy from 77.22% to 84.84%.

4.3.2 University College Dublin Sleep Apnea Database

This database is commonly known to permit automatic semantic classification of sleep apnea. The database consists of 25 full overnight, three-channel Holter ECG polysomnograms collected from adult subjects suspected of sleep-disorded breathing. Subjects were selected over a 6-month period from the sleep disorder clinic at St Vincent's University Hospital, Dublin. The subjects were all above 18 years of age and were all diagnosed with obstructive sleep apnea, central sleep apnea, or primary snoring. No subjects in the study had known cardiac disease, autonomic dysfunction, and/or medication known to interfere with heart rate.

In the University College Dublin Sleep Apnea database, we consider the 30-second epochs that are labeled by professionals as "Awake Stage" (AS) or "Stage I" (S1) where S1 stands for the drowsy stage. Since we are interested in detecting the transition from the awake state to the drowsy state, we only consider the adjacent AS and S1 epochs, where the S1 epoch immediately follows the AS epoch. Similarly to the MIT-BIH database, the University College Dublin Sleep Apnea

database includes a number of AS epochs between other sleep stages, and they are excluded from our experiment.

Table 8: Drowsiness-detection accuracy using UCDDB Dataset 5-seconds

Subject	Conventional	Counter-based	Proposed
1	54.7%	56.4%	68.2%
2	58.9%	67.0%	67.5%
3	52.9%	56.9%	50.0%
4	65.0%	70.6%	79.4%
5	63.2%	69.0%	80.0%
6	66.2%	73.3%	86.7%
7	51.1%	58.9%	63.9%
8	60.2%	70.8%	69.2%
9	56.2%	54.1%	67.6%
10	85.2%	90.0%	95.0%
11	67.7%	73.6%	55.4%
12	59.9%	68.7%	55.0%
13	57.5%	59.5%	61.5%
14	57.9%	62.5%	75.0%
15	73.2%	88.2%	90.9%
16	61.1%	73.3%	72.2%
17	55.7%	61.8%	66.7%
18	57.8%	70.0%	68.8%
19	67.8%	76.2%	80.8%
20	50.7%	53.3%	56.3%
21	55.6%	59.2%	61.5%
22	64.9%	70.7%	75.0%
23	60.6%	72.1%	73.7%
24	64.5%	74.2%	75.0%
25	57.4%	61.9%	76.9%
Average	**61.0%**	**67.7%**	**70.9%**

Table 8 compares the detection accuracy for the aforementioned three approaches. The CNN-based approach provides better detection performance

than the other conventional methods. On average, the proposed method is able to improve the detection accuracy from 67.68% to 70.89%.

4.3.3 DREAMS Subjects Database

DREAMS Subject Database is used to test and validate the proposed method. Data collected were acquired in a sleep laboratory of a Belgium hospital using a digital 32-channel polygraph. At least two EOG channels (P8-A1, P18-A1), three EEG channels (CZ-A1 or C3-A1, FP1-A1, and O1-A1) and one submental EMG channel were recorded [45]. In this work, we used a single EEG channel, CZ-A1. The database including sleep-staging scores for every subject according to R&K and AASM classification of sleep stages. In this work, sleep scoring according to AASM is used.

Table 9 compares the detection accuracy for the aforementioned three approaches. The CNN-based approach provides better detection performance than the other conventional methods. On average, the proposed method is able to improve the detection accuracy from 70.9% to 74.9%.

Table 9: Drowsiness-detection accuracy using DREAMS Dataset 5-seconds

Subject	Conventional	Counter-based	Proposed
1	65.8%	71.3%	74.4%
2	61.8%	66.1%	70.8%
3	64.4%	72.3%	71.0%
4	68.2%	73.4%	77.7%
5	74.1%	81.7%	85.2%
6	63.0%	69.3%	75.0%
7	59.9%	64.5%	72.5%
8	62.5%	68.4%	72.6%
9	64.0%	70.2%	80.3%
10	61.4%	72.1%	69.1%
11	66.6%	71.6%	74.7%
12	63.6%	69.8%	74.7%
13	62.8%	71.0%	71.7%
14	61.3%	64.4%	75.6%
15	63.0%	67.4%	74.2%
16	69.2%	75.3%	82.5%
17	68.6%	79.2%	84.2%
18	69.1%	75.1%	80.0%
19	61.6%	65.2%	50.0%
20	63.2%	69.2%	82.7%
Average	64.7%	70.9%	74.9%

5. Conclusion

The need for a reliable drowsiness detection system is arising today, as drowsiness is considered to be a major risk factor in workplace injuries and fatalities as much as alcohol. Drowsiness can be detected by observing different types of measures. Detecting drowsiness using the physiological characteristics has been proven to be the most reliable and accurate measure. In this thesis, we focused on developing EEG-based drowsiness detection system because EEG signals are considered to be the golden standard in sleep stage assessment and can be acquired easily relative to other physiological signals.

In this thesis, we aim to develop EEG-based drowsiness detection algorithms that are suitable for portable and low-power devices. Towards that goal, we introduced three main contribution proposed in this thesis: (1) a real-time drowsiness detection algorithm based on EEG suitable for portable applications with low computational complexity; (2) several novel algorithms to train classifiers that can be implemented on a chip with low-power fixed-point arithmetic with extremely small word length; (3) an instantaneous drowsiness detection system suitable for short-time windows of single-channel EEG signal.

Described in Chapter 2, a single-channel real-time drowsiness detection algorithm based on EEG. The proposed approach is suitable for portable

applications with low computational complexity. It calculates the relative power from 8 different frequency bands: the delta band (1−3 Hz), the theta band (4−7 Hz), the low-alpha band (8−9 Hz), the high-alpha band (10−12Hz), the low-beta band (13−17 Hz), the high-beta band (18−30 Hz), the low-gamma band (31−40 Hz) and the high-gamma band (41−50 Hz). Next, a novel counter-based approach is proposed to extract the important features of interest. Finally, a linear SVM is trained to accurately distinguish the awakening and drowsiness state. Our experimental results that were based on three publicly available data sets demonstrate that the proposed method offers superior detection accuracy (MIT-BIH: 83.36%, UCDDB: 75.05%, DREAMS: 74.26%) compared to the conventional method (MIT-BIH: 70.62%, UCDDB: 59.23%, DREAMS: 64.44%).

In Chapter 3, we introduce several novel machine learning algorithms (e.g., LDA-FP, SVM-FP and LR-FP) to train robust classifiers that are suitable for on-chip low-power implementation with a fixed-point arithmetic. These training algorithms formulate the corresponding MIP problems that take into account the non-idealities (i.e., rounding and overflow) associated with a fixed-point arithmetic. In addition, branch-and-bound methods with various heuristics are used to solve the MIP problems robustly (i.e., with guaranteed global optimum).

Our numerical experiments demonstrate that the proposed methods are able to achieve up to 1.67x reduction in the word length when compared to the conventional approaches without surrendering any classification accuracy. Our proposed techniques can be applied to a broad range of emerging applications. In this thesis, we demonstrate its efficacy when applied on EEG-based drowsiness detection applications.

Chapter 4 presents an instantaneous drowsiness detection algorithm based on CNN applicable for short-time EEG epochs. In this work, we employ a unique CNN architecture composed of four convolution layers where each is followed by a rectified linear unit, a fully-connected layer, and a softmax layer. The proposed CNN architecture is fed with two-dimensional signals of frequency-specific activity over time extracted from a single-channel EEG. Our experimental results demonstrate that our CNN-based drowsiness detection system is capable of detecting drowsiness in a short time (five seconds) with higher accuracy (MIT-BIH: 84.8%, UCDDB: 70.9%, DREAMS: 74.9%) when compared to conventional methods (MIT-BIH: 71.0%, UCDDB: 61.0%, DREAMS: 64.7%) and the counter-based method (MIT-BIH: 77.2%, UCDDB: 67.7%, DREAMS: 70.9%).

In the future, we will further pursue three possible research tasks: (1) wearable derive for industrial workers, (2) fixed-point implementation for CNN, and (3) multimode data fusion. In the following, we will briefly discuss each of these tasks.

5.1 Wearable Device for Industrial Workers

The proposed work in Chapter 2 has been carried out to build a practical real-time EEG-based drowsiness detection system that is suitable for industrial workers. The counter-based method described in Chapter 2 was implemented into a proof-of-concept prototype for a wearable device for the purpose of detecting drowsiness among industrial workers; this prototype was called *WakeCap Helmet*. In our future work, a wearable real-time drowsiness-detection system for industrial workers will be designed to carry the following important "features":

- High accuracy: It is important to have high levels of precision when detecting drowsiness with minimal noises and errors. At the industrial sector, missing a true drowsiness-related incident could lead to an accident. However, many false positives might lead to lower productivity. Thus, it is essential to have both high sensitivity and specificity for the application in order for it to be practical in the industrial sector.

- Small latency: The entire detection design should be adequately fast to ascertain that the industrial workers are alerted instantly in order to evade drowsiness-related accidents. The response time must be reliable and communicated to the channel for transmission of the data, that is, a dependable wired or wireless communication channel should easily transmit the data to an external device (e.g., smart phone, cloud server, etc.) to facilitate processing.
- Low power: It is important that the wearable device operates for a long period of time without recharging the battery and therefore ensuring that the consumption of power is reduced. It is crucial to have a design that is specific to the application's implementation hardware in order to fit in the constricted power budget. For an industrial application, a practical system would be able to operate continuously on battery for more than 24 hours.

Figure 15: Helmet integration with EEG-based drowsiness-detection system

The proposed work in this thesis enables the development of a practical drowsiness-detection device, which is suitable for the industrial workers. In addition to the proposed algorithms, we will suggest a hardware integration design to build a practical drowsiness detection system for the industrial workers. Figure 15 illustrate the proposed approach, which features the following features:

- Robustness: EEG electrodes are placed on the back side of the helmet's harness. When workers tighten the helmet harness, EEG electrode are fixated robustly without much effort. This would allow for less noise, a better signal-to-noise ratio, and better EEG signal acquisition overall.
- Rigidity: The industrial environment is tough, and suitable wearables require the device to be strong, waterproof, and dustproof. In this design, the enclosure is designed to protect the electronic circuit.
- Convenience: Workers are required to wear their safety helmets at all times. The proposed design allows users to easily use this device by simply wearing the helmet. This helps improve the adaptability and acceptability in addition to reducing training costs.

- All-in-one integration: The device is composed of an electrode for EEG-signal acquisition, a low-power microcontroller, sounder, battery, and a battery-charging circuit.
- Local processing: Data is processed locally to alert workers in the case of drowsiness; especially in challenging environments where connectivity is unreliable; such as, underground mining, or in construction fields.
- Harness integration: To integrate with a harness means to be applicable in multiple helmets. In terms of regulation compliance, harnesses can be certified to be used with multiple shells.

5.2 Fixed-Point Implementation for CNN

The preliminary results obtained from the CNN-based drowsiness detection algorithm described in Chapter 4 is promising. Further development can be done by implementing the CNN-based drowsiness detection algorithm in a fixed point, which may help in reducing memory bandwidth, lowering computation time, and reducing power consumption along with the storage requirements for the CNN. A fixed-point implementation of the CNN-based approach will make it a good candidate for the development of a practical low-power wearable system for detecting drowsiness instantaneously among

industrial workers, and other portable applications.

5.3 Multimode Data Fusion

The detection accuracy of EEG-based systems can be further improved by considering multimode data sensing (e.g., EEG, EOG, etc.). As reported in the literature, even though EEG is a valuable measure for drowsiness detection, it has failed to properly work for about 10-20% of subjects in an experimental study [3]. Hence, it is essential to incorporate multimodal physiological signals in order to make the proposed detection system generic. In particular, we will study the data fusion for EEG and EOG, as slow rolling eye movement is often observed for drowsy subjects [3]. and it can be captured by EOG measurement.

References

[1] D. Dawson, A. K. Searle, and J. L. Paterson, "Look before you (s) leep: evaluating the use of fatigue detection technologies within a fatigue risk management system for the road transport industry," *Sleep medicine reviews,* vol. 18, no. 2, pp. 141-152, 2014.

[2] A. Williamson, D. A. Lombardi, S. Folkard, J. Stutts, T. K. Courtney, and J. L. Connor, "The link between fatigue and safety," *Accident Analysis & Prevention,* vol. 43, no. 2, pp. 498-515, 2011.

[3] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, and B. Vaughn, "The AASM manual for the scoring of sleep and associated events," *Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine*, 2012.

[4] S. Folkard, and P. Tucker, "Shift work, safety and productivity," *Occupational medicine,* vol. 53, no. 2, pp. 95-101, 2003.

[5] U. Techera, M. Hallowell, E. Marks, and N. Stambaugh, "Measuring Occupational Fatigue: A Comprehensive Review and Comparison of Subjective and Objective Methods." pp. 2905-2915.

[6] W. K. Viscusi, "Estimating the value of a statistical life using census of fatal occupational injuries (CFOI) data," 2013.

[7] P. Parijat, and T. E. Lockhart, "Effects of lower extremity muscle fatigue on the outcomes of slip-induced falls," *Ergonomics,* vol. 51, no. 12, pp. 1873-1884, 2008.

[8] E. A. Koningsveld, "Worker Fatigue: Managing concerns in rapid renewal highway construction projects."

[9] T. S. Abdelhamid, and J. G. Everett, "Physiological demands during construction work," *Journal of construction engineering and management,* vol. 128, no. 5, pp. 427-437, 2002.

[10] T. Cheng, G. C. Migliaccio, J. Teizer, and U. C. Gatti, "Data fusion of real-time location sensing and physiological status monitoring for ergonomics

analysis of construction workers," *Journal of Computing in Civil engineering,* vol. 27, no. 3, pp. 320-335, 2012.

[11] D. J. Edwards, B. Sirois, T. Dawson, A. Aguirre, B. Davis, and U. Trutschel, "Evaluation of fatigue management technologies using weighted feature matrix method," 2007.

[12] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting driver drowsiness based on sensors: a review," *Sensors,* vol. 12, no. 12, pp. 16937-16953, 2012.

[13] C. M. Rudin-Brown, A. Williamson, and M. Lenne, "Can driving simulation be used to predict changes in real-world crash risk?[Monash University Accident Research Centre-Report 299]," *http://www.monash.edu.au/miri/research/reports/muarc299.html http://www.monash.edu.au/miri/research/reports/muarc299. pdf*, pp. 1-40, 2009.

[14] D. Wu, V. J. Lawhern, S. Gordon, B. J. Lance, and C.-T. Lin, "Driver Drowsiness Estimation from EEG Signals Using Online Weighted Adaptation Regularization for Regression (OwARR)," *IEEE Transactions on Fuzzy Systems*, 2016.

[15] K. C. Patrick, S. A. Imtiaz, S. Bowyer, and E. Rodriguez-Villegas, "An algorithm for automatic detection of drowsiness for use in wearable EEG systems." pp. 3523-3526.

[16] Y. M. Chi, Y.-T. Wang, Y. Wang, C. Maier, T.-P. Jung, and G. Cauwenberghs, "Dry and noncontact EEG sensors for mobile brain–computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 20, no. 2, pp. 228-235, 2012.

[17] A. Picot, S. Charbonnier, and A. Caplier, "On-Line Detection of Drowsiness Using Brain and Visual Information," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* vol. 42, no. 3, pp. 764-775, 2012.

[18] Z. Mardi, S. N. M. Ashtiani, and M. Mikaili, "EEG-based drowsiness detection for safe driving using chaotic features and statistical tests," *Journal of medical signals and sensors,* vol. 1, no. 2, pp. 130, 2011.

[19] R. A. Carter, "Mine Safety: Working Within the System," *Engineering and Mining Journal,* vol. 217, no. 3, pp. 50, 2016.

[20] T. Akerstedt, A. Anund, J. Axelsson, and G. Kecklund, "Subjective sleepiness is a sensitive indicator of insufficient sleep and impaired waking function," *J Sleep Res,* vol. 23, no. 3, pp. 240-52, Jun, 2014.

[21] A. G. Correa, L. Orosco, and E. Laciar, "Automatic detection of drowsiness in EEG records based on multimodal analysis," *Medical engineering & physics,* vol. 36, no. 2, pp. 244-249, 2014.

[22] O. G. Okogbaa, R. L. Shell, and D. Filipusic, "On the investigation of the neurophysiological correlates of knowledge worker mental fatigue using the EEG signal," *Applied Ergonomics,* vol. 25, no. 6, pp. 355-365, 1994.

[23] J. Chen, X. Song, and Z. Lin, "Revealing the "Invisible Gorilla" in construction: Estimating construction safety through mental workload assessment," *Automation in Construction,* vol. 63, pp. 173-183, 2016.

[24] G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," *Neuroscience & Biobehavioral Reviews,* vol. 44, pp. 58-75, 2014.

[25] J. Arnin, D. Anopas, M. Horapong, P. Triponyuwasi, T. Yamsa-Ard, S. Iampetch, and Y. Wongsawat, "Wireless-based portable EEG-EOG monitoring for real time drowsiness detection." pp. 4977-4980.

[26] L. Chin-Teng, W. Ruei-Cheng, L. Sheng-Fu, C. Wen-Hung, C. Yu-Jie, and J. Tzyy-Ping, "EEG-based drowsiness estimation for safety driving using independent component analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 52, no. 12, pp. 2726-2738, 2005.

[27] J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications,* vol. 41, no. 4, pp. 1139-1152, 2014.

[28] F.-C. Lin, L.-W. Ko, C.-H. Chuang, T.-P. Su, and C.-T. Lin, "Generalized EEG-Based Drowsiness Prediction System by Using a Self-Organizing

Neural Fuzzy System," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 9, pp. 2044-2055, 2012.

[29] M. V. Yeo, X. Li, K. Shen, and E. P. Wilder-Smith, "Can SVM be used for automatic EEG detection of drowsiness during car driving?," *Safety Science*, vol. 47, no. 1, pp. 115-124, 2009.

[30] C. T. Lin, Y. C. Chen, T. Y. Huang, T. T. Chiu, L. W. Ko, S. F. Liang, H. Y. Hsieh, S. H. Hsu, and J. R. Duann, "Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver's drowsiness detection and warning," *IEEE Trans Biomed Eng*, vol. 55, no. 5, pp. 1582-91, May, 2008.

[31] K. Li-Wei, L. Wei-Kai, L. Wei-Gang, C. Chun-Hsiang, L. Shao-Wei, L. Yi-Chen, H. Tien-Yang, W. Hsu-Hsuan, and L. Chin-Teng, "Single channel wireless EEG device for real-time fatigue level detection." pp. 1-5.

[32] S. Yu, P. Li, H. Lin, E. Rohani, G. Choi, B. Shao, and Q. Wang, "Support Vector Machine Based Detection of Drowsiness Using Minimum EEG Features," pp. 827-835, 2013.

[33] J. G. Gaspar, T. L. Brown, C. W. Schwarz, J. D. Lee, J. Kang, and J. S. Higgins, "Evaluating driver drowsiness countermeasures," *Traffic injury prevention*, vol. 18, no. sup1, pp. S58-S63, 2017.

[34] G. Niu, and C. Wang, "Driver fatigue features extraction," *Mathematical problems in engineering*, vol. 2014, 2014.

[35] N. R. Pal, C.-Y. Chuang, L.-W. Ko, C.-F. Chao, T.-P. Jung, S.-F. Liang, and C.-T. Lin, "EEG-Based Subject- and Session-independent Drowsiness Detection: An Unsupervised Approach," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 519480, 2008.

[36] S. Yu, P. Li, H. Lin, E. Rohani, G. Choi, B. Shao, and Q. Wang, "Support Vector Machine Based Detection of Drowsiness Using Minimum EEG Features." pp. 827-835.

[37] B. V. Hal, S. Rhodes, B. Dunne, and R. Bossemeyer, "Low-cost EEG-based sleep detection." pp. 4571-4574.

[38] T. J. Morrow, and K. L. Casey, "A microprocessor device for the real-time detection of synchronized alpha and spindle activity in the EEG," *Brain research bulletin*, vol. 16, no. 3, pp. 439-442, 1986.

[39] S. K. Lal, A. Craig, P. Boord, L. Kirkup, and H. Nguyen, "Development of an algorithm for an EEG-based driver fatigue countermeasure," *Journal of Safety Research*, vol. 34, no. 3, pp. 321-328, 2003.

[40] A. Rechtschaffen, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," *Public health service*, 1968.

[41] L. Marple, "Computing the discrete-time" analytic" signal via FFT," *IEEE Transactions on signal processing*, vol. 47, no. 9, pp. 2600-2603, 1999.

[42] K. F. K. Wong, A. Galka, O. Yamashita, and T. Ozaki, "Modelling non-stationary variance in EEG time series by state space GARCH model," *Computers in biology and medicine*, vol. 36, no. 12, pp. 1327-1335, 2006.

[43] M. H. Silber, S. Ancoli-Israel, M. H. Bonnet, S. Chokroverty, M. M. Grigg-Damberger, M. Hirshkowitz, S. Kapen, S. A. Keenan, M. H. Kryger, and T. Penzel, "The visual scoring of sleep in adults," *Journal of clinical sleep medicine*, vol. 3, no. 02, pp. 22-22, 2007.

[44] C. Bishop, "Pattern Recognition and Machine Learning (Information Science and Statistics), chapter 3, pages 138--147," Springer-Verlag New York, Inc, 2006.

[45] "The DREAMS Subjects Database.," *http://www.tcts.fpms.ac.be/~devuyst/Databases/DatabaseSubjects/*, University of MONS - TCTS Laboratory, 2015.

[46] L. Clifton, D. A. Clifton, M. A. Pimentel, P. J. Watkinson, and L. Tarassenko, "Gaussian process regression in vital-sign early warning systems." pp. 6161-6164.

[47] B. Alshaqaqi, A. S. Baquhaizel, M. E. A. Ouis, M. Boumehed, A. Ouamri, and M. Keche, "Driver drowsiness detection system." pp. 151-155.

[48] K. U. Anjali, A. K. Thampi, A. Vijayaraman, M. F. Francis, N. J. James, and B. K. Rajan, "Real-time nonintrusive monitoring and detection of eye blinking in view of accident prevention due to drowsiness." pp. 1-6.

[49] B. Cyganek, and S. Gruszczynski, "Eye recognition in near-infrared images for driver's drowsiness monitoring." pp. 397-402.

[50] S. Darshana, D. Fernando, S. Jayawardena, S. Wickramanayake, and C. DeSilva, "Efficient PERCLOS and Gaze Measurement Methodologies to Estimate Driver Attention in Real Time." pp. 289-294.

[51] M. J. Flores, J. M. Armingol, and A. D. L. Escalera, "Driver drowsiness detection system under infrared illumination for an intelligent vehicle," *IET Intelligent Transport Systems,* vol. 5, no. 4, pp. 241-251, 2011.

[52] M. J. Flores, J. M. Armingol, and A. d. l. Escalera, "Real-time drowsiness detection system for an intelligent vehicle." pp. 637-642.

[53] C. In-Ho, H. Sung Kyung, and K. Yong-Guk, "Real-time categorization of driver's gaze zone using the deep learning techniques." pp. 143-148.

[54] Y. Ishii, T. Ogitsu, H. Takemura, and H. Mizoguchi, "Real-time eyelid open/closed state recognition based on HLAC towards driver drowsiness detection." pp. 2449-2454.

[55] G. A. Constantinides, P. Y. Cheung, and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 22, no. 10, pp. 1432-1442, 2003.

[56] A. B. Kinsman, and N. Nicolici, "Automated range and precision bit-width allocation for iterative computations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 30, no. 9, pp. 1265-1278, 2011.

[57] A. Mallik, D. Sinha, P. Banerjee, and H. Zhou, "Low-power optimization by smart bit-width allocation in a SystemC-based ASIC design environment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 26, no. 3, pp. 447-455, 2007.

[58] W. T. Padgett, and D. V. Anderson, "Fixed-point signal processing," *Synthesis Lectures on Signal Processing,* vol. 4, no. 1, pp. 1-133, 2009.

[59] C. M. Bishop, "Pattern recognition and machine learning, 2006," *@@@@@@@*, vol. 60, no. 1, pp. 78-78, 2012.

[60] G. H. Golub, and C. F. Van Loan, *Matrix computations*: JHU Press, 2012.

[61] K.-L. Hsiung, S.-J. Kim, and S. Boyd, "Tractable approximate robust geometric programming," *Optimization and Engineering,* vol. 9, no. 2, pp. 95-118, 2008.

[62] V. Vapnik, *Statistical learning theory. 1998*: Wiley, New York, 1998.

[63] S. Boyd, and L. Vandenberghe, *Convex optimization*: Cambridge university press, 2004.

[64] S. Ceria, C. Cordier, H. Marchand, and L. A. Wolsey, "Cutting planes for integer programs with general integer variables," *Mathematical programming,* vol. 81, no. 2, pp. 201-214, 1998.

[65] M. ApS, "The MOSEK optimization software (2014)," *URL http://www.mosek.com*, 2012.

[66] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008.

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

[68] R. P. Balandong, R. F. Ahmad, M. N. M. Saad, and A. S. Malik, "A Review on EEG-Based Automatic Sleepiness Detection Systems for Driver," *IEEE Access,* vol. 6, pp. 22908-22919, 2018.

[69] R.-J. Dzeng, Y.-C. Fang, and I.-C. Chen, "A feasibility study of using smartphone built-in accelerometers to detect fall portents," *Automation in Construction,* vol. 38, pp. 74-86, 2014.

[70] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks,* vol. 4, no. 2, pp. 251-257, 1991.

[71] N. Buduma, and N. Locascio, *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*: " O'Reilly Media, Inc.", 2017.

[72] D. H. Hubel, and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology,* vol. 148, no. 3, pp. 574-591, 1959.

[73] M. Hajinoroozi, Z. Mao, T.-P. Jung, C.-T. Lin, and Y. Huang, "EEG-based prediction of driver's cognitive performance by deep convolutional neural network," *Signal Processing: Image Communication,* vol. 47, pp. 549-555, 2016.

[74] H. Cecotti, and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE transactions on pattern analysis and machine intelligence,* vol. 33, no. 3, pp. 433-445, 2011.

[75] J. Latvala, "Applying neural networks for improving the MEG inverse solution," 2017.

[76] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." pp. 315-323.

[77] C. M. Bishop, "Machine learning and pattern recognition," *Information Science and Statistics. Springer, Heidelberg*, 2006.

[78] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks." pp. 559-563.

[79] J.-T. Huang, J. Li, and Y. Gong, "An analysis of convolutional neural networks for speech recognition." pp. 4989-4993.

[80] S. Dieleman, and B. Schrauwen, "End-to-end learning for music audio." pp. 6964-6968.

[81] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR." pp. 315-320.

[82] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.