The Pennsylvania State University
The Graduate School

MONITORING SLEEP QUALITY USING SMARTWATCHES

A Thesis in
Computer Science and Engineering
by
Yifei Xiao

© 2020 Yifei Xiao

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2020
The thesis of Yifei Xiao was reviewed and approved by the following:

Guohong Cao
Distinguished Professor of Computer Science and Engineering
Thesis Advisor

Sencun Zhu
Associate Professor of Computer Science and Engineering

Chitaranjan Das
Distinguished Professor of Computer Science and Engineering
Head of the Department of Computer Science Engineering
Abstract

Sleep plays an essential role in maintaining good physical and emotional health. Sleep study in a hospital or sleeping center is the traditional way to measure the sleep quality, which is very expensive and time consuming. Thus, there is an urgent need to design a low-cost and widely applicable sleep monitoring system for people to use in their daily lives.

In this thesis, we present a smartwatch based system that leverages the built-in accelerometer sensors to monitor sleep quality. First, we propose techniques to estimate the respiratory rate during sleep, which can be used to assess the sleep quality and to predict some sleep-related diseases. To calculate the respiratory rate, we design a filter to extract the respiratory signal from the collected weak and noisy sensing data along each axis. Fast Fourier Transform (FFT) is applied to estimate the respiratory rate along each axis, and an axes mixture approach is designed to improve the estimation accuracy. Second, we proposed techniques to enhance our system to detect sleep apnea, which is a common sleep disorder where the patient stops breathing during sleep. We identify some special characteristics of sleep apnea and extract proper features for detecting sleep apnea. Based on the extracted features, we apply machine learning algorithms to detect sleep apnea. The performance of our system is evaluated through real experiments, and the evaluation results shows that our system can estimate the respiratory rate and detect sleep apnea with high accuracy.
# Contents

List of Figures vi

List of Tables viii

1 Introduction 1
   1.1 Motivation .................................................. 1
   1.2 Challenges .................................................. 3
   1.3 Contributions .............................................. 4

2 Related Work 5
   2.1 Sleep apnea detection ............................... 5
   2.2 Machine learning in sleep study .......................... 6

3 Respiratory Rate Estimation 7
   3.1 Denoise data ............................................. 8
      3.1.1 Moving average .................................. 8
      3.1.2 Total variation .................................. 9
   3.2 Estimate breathing rate ............................... 11
   3.3 Mix three axes .......................................... 12
      3.3.1 Kalman-Mode filter ............................... 12

4 Apnea Detection 16
   4.1 Different types of sleep apnea ......................... 16
4.1.1 Obstructive sleep apnea ........................................ 16
4.1.2 Central sleep apnea ........................................... 17
4.1.3 Hypopnea ....................................................... 18
4.2 Preprocessing ...................................................... 18
4.2.1 Data Calibration ................................................. 18
4.2.2 Sleep Duration .................................................. 20
4.3 Feature extraction .................................................. 22
4.3.1 Spikes ............................................................ 23
4.3.2 Distance between peaks ....................................... 29
4.3.3 Number and amplitude of peaks ............................... 30

5 Performance Evaluation ............................................ 32
  5.1 Experimental dataset collection .................................. 33
  5.2 Respiratory rate .................................................. 33
     5.2.1 Effect of window size and filter ............................ 33
     5.2.2 Effect of regularization parameter ......................... 35
  5.3 Sleep apnea detection ............................................ 36
     5.3.1 Total sleep time estimation accuracy ....................... 36
     5.3.2 Features of sleep apnea .................................... 37
     5.3.3 Overall performance ....................................... 40

6 Conclusion and Future Work ..................................... 46
  6.1 Conclusion ....................................................... 46
  6.2 Future work ..................................................... 46

Bibliography .......................................................... 48
List of Figures

1.1 Polysomnography ........................................... 2
3.1 The process of respiration. ................................. 7
3.2 Chest movement during breathing. ......................... 7
3.3 Respiratory Rate Estimation System overview. .......... 8
3.4 Moving average filter ........................................ 9
3.5 Total variation filter ......................................... 10
3.6 Total variation of acceleration data (along x-axis). ..... 11
3.7 Chest movement and acceleration data with their FFT. .... 13
3.8 Estimations of frequencies along three axes and chest movement. 14
3.9 The Kalman-Mode Filter ..................................... 15
4.1 Airflow and watch data during OSA. ....................... 17
4.2 Airflow and watch data during CSA. ....................... 18
4.3 Airflow and watch data during Hypopnea. ................ 19
4.4 Irregular acceleration data during CSA. ................... 19
4.5 Lag 1 difference of Fig 4.4(a) ............................ 20
4.6 A patient’s sleep for one night ............................. 22
4.7 Data differencing and its auto correlation. ................. 27
4.8 AIC of 1st order differencing of original data ............. 28
4.9 Outlier Detection ............................................. 29
4.10 Traditional peak detection algorithm on CSA. .......... 30
4.11 Improved peak detection algorithm on CSA. ............. 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Effect of window size and filter on estimation.</td>
<td>35</td>
</tr>
<tr>
<td>5.2</td>
<td>Effect of regularization parameter on estimation.</td>
<td>36</td>
</tr>
<tr>
<td>5.3</td>
<td>Bland-Altman plot for total sleep time measured by PSG vs ApneaDetector</td>
<td>38</td>
</tr>
<tr>
<td>5.4</td>
<td>CSA, Hypopnea, and OSA by each patient</td>
<td>38</td>
</tr>
<tr>
<td>5.5</td>
<td>Number of peaks in a 60-second window</td>
<td>40</td>
</tr>
<tr>
<td>5.6</td>
<td>Peak amplitude of different types of sleep</td>
<td>41</td>
</tr>
<tr>
<td>5.7</td>
<td>Maximum standardized residual of different types of sleep</td>
<td>42</td>
</tr>
<tr>
<td>5.8</td>
<td>CDF of the distance between end of OSA events and spikes</td>
<td>43</td>
</tr>
<tr>
<td>5.9</td>
<td>Peak vs. Residual</td>
<td>45</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Features extracted in ApneaDetector. . . . . . . . . . . . . . . 23
5.1 AHI Rating. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
5.2 Descriptive characteristics of the clinic study. . . . . . . . . . 34
5.3 Coefficients of the fitted logistic regression model. . . . . . . 37
5.4 Sleep parameters scored by PSG vs. ApneaDetector . . . . . . 39
5.5 The performance of different classification approaches. . . . . 41
5.6 Performance of different spikes detection methods . . . . . . 42
5.7 Mean, median and 97% percentile of event duration . . . . . . 43
5.8 The classification result of three classes. . . . . . . . . . . . . 44
Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor Prof. Guohong Cao for his continuous support of my master’s study and research, for his patience, motivation, enthusiasm, and immense knowledge. I also want to show my appreciation to my peer graduate student, Yeming Tang and Xianda Chen, for their comments and help. Finally, I want to thank another graduate student Chuyue Xie. Her suggestions help me improve the quality of the thesis a lot.
Chapter One

Introduction

1.1 Motivation

Sleep is a vital component of every person’s overall health and well-being; it plays an essential role in maintaining good physical and emotional health. Clinical studies show that sleep is related to many diseases, including diabetes, depression, and obesity [26, 15]. According to American Sleep Association (ASA), 50-70 million U.S. adults have a sleep disorder, including insomnia, sleep apnea, and snoring [2].

Sleep apnea is a sleep disorder where a person has pauses in breathing or periods of shallow breathing during sleep. Each pause can last for a few seconds to a few minutes, and they happen several times a night [37]. It affects 9% to 15% of women and 24% to 31% of men in the U.S. [2]. Sleep apnea is classified into two different types, Obstructive Sleep Apnea (OSA) and Central Sleep Apnea (CSA). Obstructive Sleep Apnea is more common, occurring in 2% to 4% of middle-aged adults and 1% to 3% of preschool children, and is generally caused by a collapse of the upper respiratory airway [5]. However, over 80% of moderate to severe sleep apnea cases go undiagnosed because of the inconvenience, expenses, and unavailability of testing.

To date, technology has not enabled personalized, in-place sleep quality monitoring and analysis. The traditional sleep testing process is called Polysomnography (PSG), which is a comprehensive recording of the bio-
physiological changes that occur during sleep [18]. As shown in Figure 1.1, the PSG monitors many body functions, including brain activity (EEG), eye movements (EOG), muscle activity or skeletal muscle activation (EMG), and heart rhythm (ECG), during sleep. The patient must sleep at a sleep clinic with electrodes attached to the body for at least one night. Overnight Polysomnography may cost hundreds to thousands of dollars for each night. The mere dependency on PSG needs to be taken away from the laboratory for more straightforward detection and faster treatment of sleep apnea.

With the rapid development of smart devices, people are trying to use simpler and more efficient ways to monitor body health. By leveraging sensors on smart devices, a lot of smart health applications have been proposed [19, 21, 33, 34], including monitoring sleep quality and detecting sleep apnea [23, 11, 35, 17]. In recent years, wearable devices like smartwatches are becoming more and more popular; many commercial wrist-worn devices such as Fitbit or AppleWatch are able to track user’s sleep. However, they can only record some coarse-grained sleep data such as sleep duration and body movement, and none of them detect sleep apnea. In this thesis, we propose a smartwatch-based system to detect sleep apnea by using its embedded accelerometer sensors. By using time series statistical models and machine
learning methods, our system can analyze acceleration data recorded by smartwatch. Users only need to wear a smartwatch during sleep and are able to get an analysis of sleep quality such as respiration rate and sleep apneas in the next morning.

### 1.2 Challenges

We address the following challenges in our system. First, we show that acceleration data recorded by the smartwatch is highly related to chest movement data collected by PSG. During sleep, the chest has a periodic movement caused by the repeated inhalation and exhalation process during respiration. We found that acceleration data has the same pattern. To estimate the user’s respiratory rate, the window of raw data is first fed into a filter to remove noise. Since the signal variation in the filtered acceleration data is mainly caused by the periodic patterns of inhalation and exhalation while the user breathes during sleep, we apply Fast Fourier Transform (FFT) to the filtered data and use the frequency with the largest magnitude to estimate the user’s respiratory rate. The estimates from three axes are then fused as the respiratory rate at the corresponding time [35].

Second, we show that there will be an anomaly on acceleration data when the user appears to have sleep apnea. When sleep apnea starts, the anomaly on acceleration data appears and disappears quickly. Two different methods are used to detect the anomaly. The first method is by using autoregressive integrated moving average (ARIMA) model [41] to detect outliers on acceleration data within each window. ARIMA model is good at analyzing time series data and can be used as anomaly detection. By fitting various ARIMA models on different windows, we can detect outliers in each window, which is an indication of the appearance of sleep apnea. The second method is to leverage machine learning algorithms. Supervised classification algorithms are applied to help identify and classify sleep apnea events. Compare to the
ARIMA model, the machine learning approach is more efficient since there is no need to fit different models. Once a suitable model is formed, and enough data is trained, we can use this model on every acceleration data to detect sleep apnea.

1.3 Contributions

In summary, the contributions of the thesis are as follows.

- We design a smartwatch based system which can estimate respiratory rate and detect sleep apnea.
- We show that there is a strong relation between acceleration data recorded by smartwatch and chest data collected by PSG.
- We find that anomaly appearance in the collected acceleration data whenever sleep apnea occurs.
- We utilize time series analysis techniques to detect anomaly on the collected acceleration data, which helps us infer the appearance of sleep apnea.
- We take advantage of machine learning techniques to build models that can detect sleep apnea more efficiently.

The rest of the thesis is organized as follows. In Chapter 2, we discuss the related work. In Chapter 3, we show the relationship between acceleration data and chest data. Chapter 4 introduces different ways to classify different kinds of sleep apnea. In Chapter 5, we present the system performance evaluations. In Chapter 6, we conclude the thesis and propose several future research directions.
Chapter Two

Related Work

2.1 Sleep apnea detection

Sleep apnea detection has been studied in many previous works. Traditionally, Polysomnography (PSG) is used to detect sleep apnea [22]. Polysomnography measures several sleep variables, one of which is the apnea-hypopnea index (AHI). The AHI is defined as the sum of apneas and hypopneas per hour of sleep. Until today, PSG is still considered as the most accurate way to detect sleep apnea. Different electric signals have been studied to estimate sleep apnea. In [25, 20, 9], oxygen saturation (SpO2) is considered as the metric of estimating sleep apnea. In [30, 5, 36], Electrocardiography (ECG) signal is used as a measurement to detect sleep apnea. Both signals can be extracted from PSG. However, spending one night at a sleep clinic may cost hundreds to thousands of dollars. Furthermore, sleep experts have to look at PSG results minute by minute to reach a conclusion. This process takes a lot of time and labor. Under such circumstances, many researchers are looking for a more efficient way to detect sleep apnea.

With the development of smart devices, more and more portable devices have been used to detect sleep apnea. In [25, 20, 9], the pulse oximeter is used to monitor oxygen saturation (SpO2) during sleep, and patients can use pulse oximeter at home and connect it to a smartphone or other device to record data. In [16, 24], the microphone is exploited to monitor sleep quality and
estimate sleep apnea. In [23], the smartphone is transforming into an active sonar system to detect chest movement, which can be used to determine sleep apnea. However, this method needs to put the smartphone close to the user at a particular angle. Different from all these works, we propose a smartwatch-based system to estimate sleep apnea and design different models to get high accuracy estimation.

2.2 Machine learning in sleep study

Machine learning classifications, fueled by increases in computing power and availability of large labeled datasets, have recently matched the performance of medical experts in complex medical pattern recognition tasks. Researchers have recently favored using deep learning to study sleep stages. Sleep is broken down into 5 phases: wake, N1, N2, N3, and R. Stages N1 to N3 have considered non-rapid eye movement sleep, each progressively going into a deeper sleep. Sleep is staged in sequential 30-second epochs, and each of these epochs is assigned a specific sleep stage [27]. This staging method is very suitable for machine learning and deep learning. In [4, 3], support vector machines (SVM) have been utilized to classify the sleep stage by learning Electroencephalogram (EEG) data. In [40], a one-dimensional convolutional neural network (1D-CNN) is used to train PSG signals to predict the sleep stage. Researchers combined convolutional neural network (CNN) with recurrent neural network (RNN) to achieve long-term high accuracy [28, 7]. In this thesis, we design use different supervised classification algorithms to detect sleep apnea in a faster and more efficient way.
Chapter Three

Respiratory Rate Estimation

As shown in Figure 3.1, respiration is the movement of oxygen from the outside environment to the cells within tissues, and the transport of carbon dioxide in the opposite direction, which consists of repeated cycles of inhalation and exhalation. Inhalation begins with the contraction of the muscles attached to the rib cage; this causes an expansion in the chest cavity. During exhalation, the air is moved out from the lungs, and diaphragm and muscles relax; this produces a contraction in the chest cavity. As shown in Figure 3.2, the periodic increase and decrease in the volume of the chest cavity will lead to the periodic contractions and relaxations of the corresponding muscles, which eventually lead to the periodic movement of the chest, abdomen, arms and wrists.

![Figure 3.1: The process of respiration.](image1)

![Figure 3.2: Chest movement during breathing.](image2)

Our system leverages the acceleration data collected by the smartwatch on the user’s wrist to detect his/her respiration cycles during sleep. As shown in Figure 3.3, we first remove noise from the raw acceleration data. Then, we
apply frequency analysis to estimate respiratory rate from filtered data along three axes. After that, a filter is design to improve the estimation accuracy by mixing estimations from three axes together. Each step is discussed in the following sections.

![Figure 3.3: Respiratory Rate Estimation System overview.](image)

**3.1 Denoise data**

During sleep, especially during deep sleep, wrist vibrations caused by breathing are very weak. The acceleration data collected by the smartwatch is easily filled with external noise. To extract the respiratory signal from noisy raw acceleration data, we need to use a filter to remove noise.

**3.1.1 Moving average**

A straightforward way is to use a moving average filter. The moving average filter operates by averaging a number of points from the input signal to produce each point in the output signal. Let \( \mathbf{d} \in \mathbb{R}^{n \times 1} \) denote a series of
acceleration data in one sampling window, \( d_i \in d \) denotes the \( i^{th} \) point in \( d \). Let \( M \) be the length of the moving window and \( \hat{d}_i \in \hat{d} \) means the filtered data, the moving average can be written as:

\[
\hat{d}_i = \frac{1}{M} \sum_{j=0}^{M-1} d_{i+j}
\]  

(3.1)

Although this method is easy to achieve, it has several drawbacks. First, the filtered data will lose information on the first part of the original data since it needs first \( M d_i \) to calculate the first \( \hat{d}_i \). Second, the moving average will weaken rare events such as rapid shocks or other anomalies. As shown in Figure 3.4, the filtered data does not capture the anomaly near 30 seconds very well. This is because the anomaly appears in a very short time and returns to normal, so the average within that moving window weakens the impact of that anomaly. However, this anomaly is the key factor in detecting sleep apnea, as discussed in later chapter.

![Figure 3.4: Moving average of acceleration data (along x-axis). The information of first several seconds is missing, and anomaly near 30 seconds has been weaken.](image)

**3.1.2 Total variation**

In ApneaDetector, the total change filter (TV filter) is used for noise reduction. The TV filter is based on the principle that the noise signal has a higher total variation. According to this principle, we can reduce the total change of the signal to make it closer to the original signal. As shown in 3.5, unlike
moving averages, TV filters are significantly effective in reducing noise while retaining important details such as bursts and anomalies.

Figure 3.5: Total variation filter (red line) can remove noise as well as keep anomaly near 30 seconds.

By using the same denotation in 3.1.1, we can define the total variation as

$$V(\hat{d}) = \sum_{i=1}^{n-1} |\hat{d}_{i+1} - \hat{d}_i|$$

(3.2)

We define $E(d, \hat{d})$ as the sum of square errors, it can be used to measure how "close" $\hat{d}$ to $d$. It can be written as:

$$E(d, \hat{d}) = \frac{1}{n} \sum_{i=1}^{n-1} (d_i - \hat{d}_i)^2$$

(3.3)

So the total variation problem can be written as

$$\hat{d} = \arg \min_{\hat{d}} \left[ E(d, \hat{d}) + \lambda V(\hat{d}) \right]$$

(3.4)

where $\lambda$ is the regularization parameter and, in our case, is set to 0.2.

We can use Majorization-minimization (MM) algorithm [31] to solve the optimization problem of Equation 3.4. $\hat{d}$ can be minimized by the following iterations:

$$\begin{cases} 
\hat{d}^{(i+1)} &= d - A^t z^{(i)} \\
\hat{z}^{(i+1)} &= clip \left( z^{(i)} + \frac{1}{\alpha} A \hat{d}^{(i+1)}, \frac{\lambda}{2} \right) 
\end{cases}$$

(3.5)
for $i \geq 0$ with $z^0 = 0$ and $\alpha \geq \max\text{eig}(AA^t)$. The matrix $A$ is an identity matrix with $-1$ on its subdiagonal, and the clip function is defined as

$$clip(b, T) = \begin{cases} 
    b & |b| \leq T \\
    T \times \text{sign}(b) & |b| \geq T 
\end{cases}$$

(3.6)

As shown in [31], the maximum eigenvalue of $AA^t$ is less than 4 regardless of the size of $A$, so for TV denoising we can set $\alpha = 4$. As shown in Figure 3.6, once we apply enough iterations, we can remove noise from raw acceleration data.

![Figure 3.6: Total variation of acceleration data (along x-axis).](image)

### 3.2 Estimate breathing rate

After applying the total change filter, the noise is eliminated and the acceleration fluctuations caused by respiration are retained. Since during sleep, acceleration fluctuations are mainly caused by repeated inhalation and exhalation, the breathing frequency can be estimated by the frequency of fluctuations. In order to calculate such frequencies, we apply Fast Fourier Transform (FFT) to the filtered data.

Since the FFT converts the signal from its original time domain to the frequency domain, for a periodic time domain signal, there will be a strong frequency component in the corresponding FFT, which represents the frequency of the time domain signal. In ApneaDetector, we apply FFT to chest
motion data recorded by PSG and filtered acceleration data collected by smart watches. We delete the DC component and set it to 0 to eliminate the influence of gravity on the measurement results. In addition, we do not think that the frequency is greater than 0.5Hz (30bpm) or less than 0.15Hz (9bpm), because the normal breathing frequency of healthy adults is 12 to 20 breaths per minute [14].

As shown in Figure 3.7, filtered acceleration data and chest movement have the same highest frequency, indicates that we can use acceleration data collected by smartwatch on the wrist to estimate breathing rate.

3.3 Mix three axes

Although by applying FFT can get the frequency of breathing, there is still a potential problem. The accelerometer sensor on smartwatch can record data from three axes. Ideally all three axes should return the same frequency, but this does not always happen. The smartwatch on the patient’s wrist may have different position during sleep. It is possible that only some of the axes can get the clear breathing pattern, while the others might be full of white noise. In this situation, we need to find a solution to mix all three axes to get a better result.

One straightforward solution is to average frequencies of three axes. However, this may result in a large error in cases that extreme inaccurate estimates along some axes. For example, in 3.8, estimations of frequencies along the x-axis and z-axis match estimation of frequency of chest movement while large errors occur along the y-axis. The average of three axes may cause large inaccuracy.

3.3.1 Kalman-Mode filter

Instead of considering estimation at some particular time $t$, we also consider all historical data. Since the breathing rate at time $t$ is not very likely to
Figure 3.7: Chest movement data recorded by PSG (a) and raw acceleration data along three axes (in blue) and their filtered data (in red) (c,e,g). After applying FFT, the highest frequency of chest movement is 0.234Hz (14bpm) (b). The highest frequencies (in red) along three axes are 0.234Hz, 0.234Hz, 0.234Hz, respectively (d,f,h), which is the same as the frequency of chest movement data.
significantly change compared to the breathing rate at time $t - 1$ during sleep. In ApneaDetector, we apply the Kalman Filter to each axis before averaging them. The Kalman Filter produces estimates of hidden variables based on inaccurate and uncertain measurements. Also, the Kalman Filter provides a prediction of the future system state, based on the past estimations [38].

Let $z_t$ be the measurement value of respiratory rate at time $t$. Let $x_{t,t}$ be the estimate of respiratory rate at time $t$, where the estimate is made after taking the measurement $z_t$; $x_{t,t-1}$ be the estimate that was made at time $t - 1$; $x_{t+1,t}$ be the estimate that is made at the time $t$, right after the measurement $z_t$. The Kalman Filter can be written as follows:

$$x_{t,t} = x_{t,t-1} + K_t (z_t - x_{t,t-1})$$  \hspace{1cm} (3.7)

where $K_t$ is called the Kalman Gain, which is the relative weight given to the measurements and current state estimate. It can be written as follows:

$$K_t = \frac{p_{t,t-1}}{p_{t,t-1} + r}$$  \hspace{1cm} (3.8)

In Equation 3.8, $r$ is the measurement uncertainty and $p_{t,t-1}$ is the estimate uncertainty at time $t - 1$. The measurement uncertainty $r$ in our model is set to constant 0.01 and the estimate uncertainty $p_{t,t-1}$ can be written as follows:

$$p_{t,t-1} = p_{t-1,t-1} + q$$  \hspace{1cm} (3.9)
where $q$ is the process noise variance and is set to 0.0001 in our model.

We set the initial guess of respiratory rate $x_{1,0}$ as 16, as it is the average of regular respiratory rate during sleep, and set the initial estimate uncertainty $p_{1,0}$ as 10,000. After the initialization, the Kalman Gain can be calculated through Equation 3.9 and Equation 3.8. After that, an estimation can be obtained based on the Kalman Gain and measurement of current time by Equation 3.7. Although the initial guess of respiratory rate $x_{1,0}$ is arbitrary, the Kalman Filter will be able to converge close the real value. Furthermore, the Kalman Filter prevents sudden changes along the axis due to inaccurate measurements.

In addition to applying the Kalman Filter on each axis before averaging them, we also apply a mode filter in ApneaDetector. More specifically, if two of the three axes have the same estimations, we ignore the estimation from the third axes. Compared to applying the Kalman Filter on every single data point, which may lead to heavy computations, the mode filter can first pass some data very efficiently. As shown in Figure 3.9, the Kalman-Mode Filter has a higher accuracy than the average filter.

![Figure 3.9: The Kalman-Mode Filter is more accurate than the average filter.](image_url)
Chapter Four

Apnea Detection

In the previous chapter, we show that we can use acceleration data collected by the smartwatch on the user’s wrist to estimate the respiratory rate. We now show that by digging into details, we can use this data to detect sleep apnea during sleep, including Obstructive Sleep Apnea (OSA), Central Sleep Apnea (CSA), and Hypopnea.

We will use supervised machine learning algorithms for classification. One of the essential parts of training supervised machine learning models is extracting proper features. In the following, we will introduce different features for different kinds of sleep apnea.

4.1 Different types of sleep apnea

4.1.1 Obstructive sleep apnea

OSA is the most common type of sleep apnea; it is characterized by recurrent sleep that entirely or partially obstructs the upper airway for over ten seconds[6]. One of the most apparent features of OSA is that the patient’s breathing is feeble during OSA. And when sleep apnea is over, because there is not enough oxygen in the body, the patient is likely to produce several intense breaths before returning to normal breathing. For example, in Figure 4.1(a), an OSA occurs between the 20th and 41st seconds; obviously
that there is no airflow during that time. Furthermore, the vibrations of the wrist become powerless compared to fluctuations during normal sleep. After that, violent breathing was produced for about 5 seconds before returning to normal breathing. The acceleration data along three axes change drastically when breathing is intense. This is because the fierce breathing will cause a significant vibration on the wrist.

### 4.1.2 Central sleep apnea

CSA occurs when the patient holds his/her breath for a period of time. The airflow during CSA also reduces to zero, but it is not the same as OSA. CSA occurs because the patient’s brain doesn’t send proper signals to the muscles that control his/her breathing. This condition is different from obstructive sleep apnea, in which the patient can’t breathe normally because of upper airway obstruction. Figure 4.2 shows airflow as well as corresponding acceleration data along three axes during CSA. The figure shows that acceleration data are flat between the 12th and 36th seconds, indicating the absence of breathing effort. If this flat persists for more than 10 seconds, it is marked...
4.1.3 Hypopnea

Hypopnea is very similar to sleep apnea; the main difference between these two is explained by the degree of blockage in the patient’s airway. In sleep apnea, the airway is completely blocked while in hypopnea, the airway is only partially blocked. Figure 4.3 shows a hypopnea event. The airflow becomes weaker during the hypopnea. However, we found that there are spikes at the end of hypopnea in the acceleration data from the smartwatch.

4.2 Preprocessing

4.2.1 Data Calibration

In general, the acceleration data we collect should be sinusoidal due to the repeated movement of muscles on wrist. However, we find that sometimes during sleep apnea events, the acceleration data has some irregular perfor-
Figure 4.3: Airflow and watch data during Hypopnea.

(a) Airflow during Hypopnea. (b) Acceleration data along three axes during Hypopnea.

Figure 4.4: Irregular acceleration data during CSA.

(a) (b) (c)

Figure 4.4: Irregular acceleration data during CSA.

In order to know whether we need to calibrate the data or not, we can run an Augmented Dickey Fuller Test (ADF Test) [10]. In statistics, an ADF Test is an unit root test that tests the stationarity of a time series. The unit root is a characteristic of a time series that makes it non-stationary. The test model can be written as follows:
\[ y_t = c + \beta t + \alpha y_{t-1} + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} + \cdots + \phi_p \Delta Y_{t-p} \]

where \( \phi_p \Delta Y_{t-p} \) is the first difference at time \( t - p \). The null hypothesis is \( \alpha = 1 \). In other words, if \( \alpha = 1 \), an unit root exists, indicating that the data is not stationary.

We can run the ADF Test on our data set and by getting the \( p \) value, we can know if the data reject the null hypothesis or not. If \( p \) value is large, meaning that the data is not stationary and we need to calibrate it. On the other hand, if \( p \) value is small, meaning that we do not need to calibrate the data. We set the threshold for \( p \) is 0.05 in ApneaDetector. For example, the \( p \) value of Fig 4.4(a) is 0.423, so we need to calibrate the data. As shown in Fig 4.5, we take the lag 1 difference of the data to make it sinusoidal again. It can be validated that the calibrated data does not have time-dependent trend, as the the \( p \)-value for Fig 4.5 is less than 0.01.

4.2.2 Sleep Duration

One critical point is to estimate the total sleep time. Both OSA and CSA are marked only when a patient is asleep. However, during sleep all night, the patient may be affected by other sleep disorders, such as insomnia. Insomnia can cause patients to wake up for a long time and be unable to fall asleep. The acceleration data collected at these moments should be discarded. In PSG, several EEG sensors are used to measure the brain activity to determine
whether the patient is asleep or awake. However, the smartwatch is not able to monitor brain activity.

To detect whether the acceleration data is recorded during sleep or not, we calculation the total acceleration \( \mathbf{a}_t = \sqrt{a_x^2 + a_y^2 + a_z^2} \) and compare it with a threshold range \( \gamma \). The accelerometer equipped on the smartphone measures all accelerations that affect the device, and during motionless sleep, the majority of acceleration is dominated by gravity. Thus, we can compare the total acceleration \( \mathbf{a}_t \) with gravity \( 9.8 \text{ m/s}^2 \) to check if the acceleration is collected during sleep or not. The threshold range \( \gamma \) is set to \([9.8 \pm 5\%]\), which means that if \( \mathbf{a}_t \) changes over 5%, the patient is not in motionless sleep. A non-motionless sleep does not mean the patient wakes up; it may also be caused by vigorous breathing (e.g., OSA) or wrist movement. While vigorous breathing and wrist movement can cause the total acceleration data out of \( \gamma \), those data should back to normal immediately. However, if the patient suffers insomnia, the wake-up time will last a long time; thus, the total acceleration data will be out of \( \gamma \) in a non-negligible time. Based on this assumption, we apply Cole-Kripke sleep detection algorithm [12] on our data. More specifically, we consider now only activities in the current window, but also activities in previous windows and future windows. Because the predicted output is binary, a logistic regression model is fitted to predict the sleep status of the current window by gathering all activities together. The logistic regression model can be written as follows:

\[
p(i) = 1 - \frac{1}{1 + Z} \tag{4.1}
\]

\[
Z = \exp(\beta + \beta_{-4}A_{-4} + \beta_{-3}A_{-3} + \beta_{-2}A_{-2} + \beta_{-1}A_{-1} + \beta_0A_0 + \beta_1A_1 + \beta_2A_2)
\]

where \( A_{-i} \) means the activity score of the previous \( i \)th window and \( A_i \) means the activity score of the next \( i \)th window. The activity score is, in a window, the number of samples that their amplitude is over threshold \( \gamma \), where \( \gamma \) is \([9.8 \pm 5\%]\). \( p(i) \) means the probability of \( i \)th window be wake or sleep. The
predicted probability $p$ is set as 0.2 for the best result, meaning that $p > 0.2$ as wake, otherwise as sleep. Figure 4.6 shows the sleep status collected from a patient’s sleep for one night. It can be shown that our algorithm can detect the wake-up phase correctly and ignore the sudden wrist movement caused by OSA.

Figure 4.6: The total acceleration collected from a patient’s sleep for one night. The bottom shows actual sleep status collected by PSG and estimated sleep status by ApneaDetector.

### 4.3 Feature extraction

For machine learning models, extracting the proper features is essential for detecting sleep apnea. Since the magnitude of the change in acceleration is strongly related to the occurrence of sleep apnea, we are interested in extracting the features that can represent the magnitude of amplitude changes. As shown in Table 4.1, we extract seven parameters as features of the machine learning algorithms. In addition to mean, standard deviation, and median, we also extract other parameters to distinguish different sleep apnea.
<table>
<thead>
<tr>
<th>Feature*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_sr</td>
<td>the maximum standardized residual of the acceleration.</td>
</tr>
<tr>
<td>std_acc</td>
<td>the standard deviation of the acceleration.</td>
</tr>
<tr>
<td>dis_peaks</td>
<td>standard deviation of distances among peaks.</td>
</tr>
<tr>
<td>num_peaks</td>
<td>number of peaks in the window.</td>
</tr>
<tr>
<td>amp_peaks</td>
<td>amplitude of peaks in the window.</td>
</tr>
</tbody>
</table>

Table 4.1: Features extracted in ApneaDetector. All features are applied to each axis.

4.3.1 Spikes

As described earlier, when OSA or hypopnea finishes, the patient is likely to produce several intense breaths, and the wrist may move violently, so the acceleration data collected by the smartwatch will create sudden spikes. We use these sudden spikes as an indication of the occurrence of OSA or hypopnea. For example, in Figure 4.1(b), spikes arise near the 46th seconds of all three axes, and an OSA event is annotated between the 20th and the 41st seconds. Although these spikes are easily recognized by human beings, for computers, we need to design an algorithm to achieve this. One straightforward way is to set a threshold. We can calculate the mean $\mu$ and the standard deviation $\sigma$ of the window and label any sample out of range $[\mu \pm 2\sigma]$ as spikes. However, since different patients have different sleep patterns, some of them breathe harder while sleeping, while others breathe lighter. It is hard to set a constant threshold to detect spikes. In ApneaDetector, we do not give a fixed threshold. Instead, we calculate the residual of the data, the formula can be written as

$$\text{residual} = y_i - \hat{y}_i$$

where $y_i$ means the $i^{th}$ observed sample in a window and $\hat{y}_i$ means the predicted value. If there exists a spike in the window, it should cause the largest residual than other samples. So, we only need to use the largest residual.
However, simply use residual as the criterion could be problematic. For example, one patient has shallower breathing, resulting in lower wrist muscle vibrations, while another patient has deeper breathing, and wrist muscle vibrations are larger. Even both of them are in normal sleep, the residual are different. To eliminate the error caused by this situation, we can use *maximum standardized residual* (max_{sr}) instead. The standardized residual can be calculated by

\[
max_{sr} = \max \left( \frac{y_i - \hat{y}_i}{\text{std}(m)} \right)
\]

where \(\text{std}(m)\) means the standard deviation of a window \(m\). The usage of this method can standardize the residual, so that even the original data is in large scale, the residual can still be restricted in a range. The question left is how to get the predicted value \(\hat{y}_i\). Here we propose two ways to get the predicted value.

**Use mean directly**

One way is to use mean value as the predicted value. Since most time the acceleration data is sinusoidal and does not have trend (or we have already removed trend in data calibration step), the mean value can be represented as the baseline of the whole data. So the maximum standardized residual can be written as

\[
max_{sr} = \frac{\max(y_i) - \text{mean}(m)}{\text{std}(m)}
\]

One potential problem of this method is that the existence of spike may affect the mean. However, since the duration of spike is very small, we can ignore its effect on the whole window.
Use ARIMA model

Another way to get the predicted value is to use autoregressive integrated moving average (ARIMA) model. In time series analysis, an ARIMA model is a class of model that captures a suite of different standard temporal structures in time series data. ARIMA models are quite flexible in that they can fit several different types of time series, i.e., pure autoregressive (AR), pure moving average (MA), and combined AR and MA (ARMA) series [41]. ARIMA models can be applied in some cases where data is not stationary, whereby applying differencing step the non-stationarity can be eliminated. In ApneaDetector, after acceleration data is segmented and filtered by the total variation filter, an ARIMA model is fitted to the filtered acceleration data. The difference between the filtered acceleration data and the data predicted by the ARIMA model (hereafter referred to as residuals) is expected to have the constant mean and variance. By setting a proper critical value, spikes from filtered acceleration data can be detected, which indicates the occurrence of sleep apnea.

An ARIMA model is characterized by three terms: $p$, $d$ and $q$, where $p$ is the order of autoregressive (AR) term; $q$ is the order of the moving average (MA) term, and $d$ is the number of differencing required to make the time series stationary. Let $Y_t$ denote the current state, a pure autoregressive (AR) model is one that $Y_t$ depends only on its lags, which can be written as

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \epsilon_t$$  \hspace{1cm} (4.2)

where $Y_{t-1}$ means the $i^{th}$ lag of the series, $\beta_i$ is the coefficient, $\alpha$ is the intercept term and $\epsilon_t$ is the error term. Likewise, a pure moving average (MA) model is one that $Y_t$ depends only on the lagged forecast errors, which can be written as

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \cdots + \phi_q \epsilon_{t-q}$$  \hspace{1cm} (4.3)
where $\phi_i$ is the coefficient and $\epsilon_i$ is the error term. An ARIMA model is one where the data is differenced at least once to make it stationary so that the AR and MA terms can be combined. So the equation becomes:

$$ Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \cdots + \phi_q \epsilon_{t-q} \quad (4.4) $$

In order to fit an ARIMA model, the differencing order $d$ should be determined first. The right order of differencing is the minimum differencing required to get a near-stationary series, which roams around a defined mean and the auto correlation function (ACF) plot reaches zero fairly quickly. For example, Figure 4.7 shows the differencing data once and twice of original data. The autocorrelation of first-order differencing and second-order differencing is similar, and both reach to zero after short lags, indicates that differencing the original data once is enough to find AR and MA term.

The next step is to identify the AR term $p$ and the MA term $q$. A suitable ARIMA model can be evaluated by two criteria: the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). Both criteria can be used to measure the goodness of fit of the model. [8] shows that AIC can be derived in the same Bayesian framework as BIC by using different prior probabilities. In ApneaDetector, we choose AIC as the criterion of our ARIMA model. The equation of AIC can be written as

$$ AIC = 2k - 2 \ln(\hat{L}) \quad (4.5) $$

where $\hat{L}$ is the maximum value of the likelihood function for the model and

$$ k = p + q + 1 \quad (4.6) $$

Since $\hat{L}$ is the maximum value, that means for AIC, the smaller the value, the more preferable the model is. We can test several lags and lagged forecast errors to find the smallest value of AIC. As shown in Figure 4.8, we choose
Figure 4.7: Data differencing and its auto correlation.
the best fit model, that is \( p = 5 \) and \( q = 7 \). The ARIMA model for the data is

\[
Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_5 Y_{t-5} + \epsilon_t + \phi_2 \epsilon_{t-2} + \phi_3 \epsilon_{t-3} + \phi_7 \epsilon_{t-7} \tag{4.7}
\]

Figure 4.8: AIC of 1st order differencing of original data

Fitting ARIMA model can be done in many tools such as tsoutliers package in R [13]. For example, Figure 4.9 shows the result of applying ARIMA model to acceleration data from Figure 4.1. The blue line is predicted data by the ARIMA model, the gray line in the background is the filtered acceleration data, and red dots indicate potential spikes. The ARIMA model detects the location of spikes as expected. However, this method also has its own limitations. One problem is that fitting such model is very time consuming and needs heavy computation. In the extreme case, if the spike occurs at the beginning of the window, the whole predicted data would be wrong. To eliminate such error, we might need to do several fitting to get the best model, for example, forward fitting and backward fitting. This would take huge amount of time and waste computation resources.
4.3.2 Distance between peaks

When CSA occurs, the patient holds his/her breath for more than 10 seconds and then returns to normal. As shown in Figure 4.2(b), CSA occurs between the 15th and 32nd seconds, during this time, the changing rate of acceleration data reduces to almost zero, and as a result, the amplitude of acceleration data reduces to zero. Thus, measuring the amplitude in the acceleration data is critical to detecting CSA. As discussed in the previous section, the acceleration data during sleep can be approximated as a periodic sinusoidal wave. Therefore the peaks of these sinusoidal waves represent the amplitude, and the locations of these peaks produce the periodicity. When CSA occurs, the distance between two consecutive peaks increases. If the distance is larger than ten seconds, we mark it as a CSA event. Thus, we design a peak detection algorithm to track the distance between two adjacent peaks.

Traditional peak detection algorithm identifies the transition point of a signal from an uptrend to a downtrend. It finds all local maxima by a simple comparison of neighboring values. In other words, if the middle point the larger than its neighbors, it is marked as a peak. This algorithm, however, would produce a number of faulty peaks with acceleration data. Figure 4.10 shows applying traditional peak detection algorithm on acceleration data of z-axis when CSA occurs. As shown in the plot, several faulty peaks are detected by the traditional peak detection algorithm. To remove such misdetection, we apply two new measurements to improve the algorithm.

The first improvement is to set a threshold on the minimum distance between two consecutive peaks. The normal respiratory rate for healthy adults is between 12 and 20 breaths per minute [14], which means three seconds at
the maximum frequency. The frequency of acceleration data collected by the smartwatch is 8Hz. We set the threshold to be 24 samples in our algorithm. In other words, we will not consider the next 24 sample points when a peak has been detected.

The second improvement is to set a threshold of minimum amplitude of detected peaks. In ApneaDetector, we set the amplitude threshold to be the mean value of the accelerometer data in the sampling window. A detected peak is kept only if its amplitude is larger than the threshold; otherwise it is discarded.

Figure 4.11 shows the detected peak after applying aforementioned improvements to the same acceleration data. It can be shown that all erroneous peaks are removed and our algorithm identifies the correct peaks. The distance between the third and the forth peak is larger than 10 seconds, indicates the occurrence of CSA event.

### 4.3.3 Number and amplitude of peaks

As discussed in the previous section, the normal respiratory rate for healthy adults is between 12 and 20 breaths per minute. The number of breathing in the acceleration data is reflected as the number of peaks, which indicates that we can count the number of peaks within one window as another feature. In a 60-second window, the number of peaks during normal sleep is between 12
to 20. This value will decrease during sleep apnea because the patient is not breathing, so there will be no peaks. On the other hand, during hypopnea, since the patient still has breathing, the number of peaks should be similar to normal sleep.

We also extract the standard deviation of the amplitude of peaks in a sampling window as a feature for apnea detection. The amplitude of peaks during normal sleep is expected to be small, since the accelerometer data on the wrist have regular pattern due to the repeated cycles of inhalations and exhalations. By contrast, the accelerometer data varies drastically during sleep apnea, resulting in a significant standard deviation value. In hypopnea, although the subject’s breathing becomes shallow, it does not change that much as sleep apnea and thus does not produce a substantial standard deviation like sleep apnea.
Chapter Five

Performance Evaluation

In this section, we evaluate the performance of ApneaDetector based on data collected in real experiments. We first show the dataset collection. Then we show the performance of respiratory rate estimation, followed by the evaluation of sleep apnea detection.

To measure the performance, we use Apnea-Hypopnea Index (AHI) as the criteria. The AHI is the number of apneas or hypopneas recorded during the study per hour of sleep [1]. It can be calculated as follows:

\[
AHI = \frac{\text{# of Apnea} + \text{# of Hypopnea}}{\text{Total sleep hours}}
\]  

The severity of sleep apnea is classified in Table

<table>
<thead>
<tr>
<th>AHI</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;5</td>
<td>Normal</td>
</tr>
<tr>
<td>5 - 15</td>
<td>Mild Sleep Apnea</td>
</tr>
<tr>
<td>15 - 30</td>
<td>Moderate Sleep Apnea</td>
</tr>
<tr>
<td>&gt; 30</td>
<td>Severe Sleep Apnea</td>
</tr>
</tbody>
</table>

Table 5.1: AHI Rating.
5.1 Experimental dataset collection

As shown in Table 5.2, we collect data from 20 patients over 150 hours of sleep from a sleep clinic. Some patients have few sleep apnea events occurrence, while others have a lot. Each patient spent one night in the sleep clinic with PSG equipment and a smartwatch on his/her wrist during sleep. The smartwatch was synchronized with the PSG equipment. After each night, we read the acceleration data from the smartwatch’s SD card. The data from PSG equipment is considered as ground truth, and the collected acceleration data is processed separately to obtain experimental results.

5.2 Respiratory rate

5.2.1 Effect of window size and filter

One of the most important measurement factors of estimation is the window size. Different sizes of the window will affect the estimation accuracy. We test three different window sizes: 15 seconds, 30 seconds, and 60 seconds. Figure 5.1 shows the mean absolute estimation error and the absolute error when using moving average filter and Kalman-Mode Filter under different window sizes. Kalman-Mode Filter performs much better than the moving average filter. For both two filters, the mean absolute estimation error decreases, and the absolute error range is reduced when the window size is increased. This is expected because FFT is applied to the acceleration data for respiratory rate estimation, and the frequency resolution of the FFT depends on the sampling rate and the number of samples used in the FFT. We fixed the sampling rate to 16Hz in our experiment, so the frequency resolution is increased with more samples used in FFT (i.e., larger window size), which makes the estimation is more accurate. Thus in ApneaDetector, the window size is set to 60 seconds with Kalman-Mode Filter.
Table 5.2: Descriptive characteristics of the clinic study.

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Age</th>
<th>Record Time (hour)</th>
<th>AHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F</td>
<td>48</td>
<td>7.7</td>
<td>6.69</td>
</tr>
<tr>
<td>2</td>
<td>M</td>
<td>67</td>
<td>7.6</td>
<td>24.77</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>70</td>
<td>8.0</td>
<td>15.00</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>69</td>
<td>8.9</td>
<td>3.10</td>
</tr>
<tr>
<td>5</td>
<td>M</td>
<td>41</td>
<td>7.8</td>
<td>15.85</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>50</td>
<td>8.2</td>
<td>1.14</td>
</tr>
<tr>
<td>7</td>
<td>M</td>
<td>37</td>
<td>8.2</td>
<td>2.95</td>
</tr>
<tr>
<td>8</td>
<td>F</td>
<td>55</td>
<td>7.3</td>
<td>51.52</td>
</tr>
<tr>
<td>9</td>
<td>F</td>
<td>71</td>
<td>8.2</td>
<td>0.33</td>
</tr>
<tr>
<td>10</td>
<td>F</td>
<td>71</td>
<td>8.3</td>
<td>22.59</td>
</tr>
<tr>
<td>11</td>
<td>F</td>
<td>61</td>
<td>8.2</td>
<td>0.98</td>
</tr>
<tr>
<td>12</td>
<td>F</td>
<td>66</td>
<td>7.9</td>
<td>4.47</td>
</tr>
<tr>
<td>13</td>
<td>M</td>
<td>63</td>
<td>7.3</td>
<td>15.42</td>
</tr>
<tr>
<td>14</td>
<td>M</td>
<td>59</td>
<td>7.4</td>
<td>47.48</td>
</tr>
<tr>
<td>15</td>
<td>M</td>
<td>65</td>
<td>8.1</td>
<td>9.01</td>
</tr>
<tr>
<td>16</td>
<td>M</td>
<td>36</td>
<td>7.5</td>
<td>15.23</td>
</tr>
<tr>
<td>17</td>
<td>F</td>
<td>72</td>
<td>8.7</td>
<td>12.39</td>
</tr>
<tr>
<td>18</td>
<td>F</td>
<td>42</td>
<td>8.0</td>
<td>1.43</td>
</tr>
<tr>
<td>19</td>
<td>M</td>
<td>71</td>
<td>8.3</td>
<td>90.47</td>
</tr>
<tr>
<td>20</td>
<td>F</td>
<td>69</td>
<td>8.3</td>
<td>12.10</td>
</tr>
</tbody>
</table>

M: 8  avg: 59.3  std: 12.4
F: 12  avg: 8.0  std: 0.4
      avg: 17.6  std: 22.2
5.2.2 Effect of regularization parameter

When denoise the raw acceleration data, the regularization parameter $\lambda$ will have an impact on the filtered data, which will affect the subsequent estimation accuracy. We choose three different $\lambda$: 0.2, 1.0 and 5.0. Figure 5.2 shows the mean absolute estimation error and the absolute error when using Kalman-Mode Filter and 60-second window size under different regularization parameter $\lambda$. The mean absolute estimation error and the absolute error increase when $\lambda$ is larger. This is expected since when $\lambda$ is small, even some noise cannot be filtered, the respiration signal in the raw data is retained and can be used to estimate respiratory rate, while large $\lambda$ may remove respiration signal. Furthermore, large $\lambda$ may smooth outlier from the raw acceleration data, which will affect the future construction of ARIMA models and the detection of sleep apnea. Thus in ApneaDetector, the regularization parameter $\lambda$ is set to 0.2.
5.3 Sleep apnea detection

5.3.1 Total sleep time estimation accuracy

As discussed in Section 4.2, acceleration data collected during the wake-up time should be discarded. To achieve this, we apply Cole-Kripke sleep detection algorithm and fit a logistic regression model (4.1) to predict the sleep status. We choose data from 9 patients as training set and the rest as validation set. Table 5.3 shows the detailed coefficients of the model. The P Value of each $A_i$ is less than 0.001, meaning that each predictors are significant and should be included in the model. The intercept of the model is -2.1, meaning that if all activities $A_i$ are 0, the probability of this window is wake is only

$$P(wake) = 1 - \frac{1}{1 + e^{-2.1}} = 0.109$$

indicates that this window is very likely to be sleep, which is in line with expectations.

Figure 5.3 and Table 5.4 shows the Bland-Altman plot and the statistical parameters of the total sleep time of 19 patients estimated by ApneaDetector compared with total sleep time recorded by PSG. The overall mean absolute
error is 27.3 minutes, and the median absolute error is 28.9 minutes. The average overlap percentage is 85.1%, indicates that most sleep status is estimated correctly by ApneaDetector. In general, the error of estimation could be the following reasons. One reason is that the patient may wake up for a very short period of time due to the effects of sleep disorder. This period of time is marked as awake in PSG but may not be detected by ApneaDetector. Another reason is that some patients woke up in the middle of the night and lie on the bed without frequent movements. The brain activity recorded by PSG is high, but the acceleration data collected by the smartwatch remains normal. This is a fundamental limitation of estimating total sleep time without monitoring brain activity. However, later we will show that total sleep time determined by ApneaDetector is acceptable for detecting sleep apnea.

## 5.3.2 Features of sleep apnea

As discussed in Section 4.3, we extract different features from acceleration data to distinguish sleep apnea and normal sleep. We extract a total of 1018 OSA events, 125 CSA events, and 818 hypopnea events. Figure 5.4 shows the number of sleep apnea events collected by each patient. Some patients suffer from severe OSA but mild hypopnea, such as patient 18. Others have high

| Estimate     | Std. Error | z-value | Pr(>|z|) |
|--------------|------------|---------|----------|
| Intercept    | -2.109743  | 0.039778| -53.038  | < 2e-16  |
| $\beta_{-4}$ | 0.027100   | 0.003114| 8.703    | < 2e-16  |
| $\beta_{-3}$ | 0.018043   | 0.003297| 5.472    | 4.45e-08 |
| $\beta_{-2}$ | 0.018606   | 0.003580| 5.197    | 2.03e-07 |
| $\beta_{-1}$ | 0.043335   | 0.004721| 9.180    | < 2e-16  |
| $\beta_0$    | 0.094392   | 0.007272| 12.980   | < 2e-16  |
| $\beta_1$    | 0.013071   | 0.003606| 3.625    | 0.000289 |
| $\beta_2$    | 0.021515   | 0.003181| 6.763    | 1.35e-11 |

Table 5.3: Coefficients of the fitted logistic regression model.
amounts of hypopnea but less OSA events, such as patient 2. In general, the number of occurrences of CSA is much smaller than that of OSA and hypopnea. To eliminate the larger errors that may be caused by the smaller CSA sample amounts, we combine OSA and CSA and mark them as sleep apnea.

Figure 5.4: CSA, Hypopnea, and OSA by each patient

Figure 5.5 shows the number of peaks in a 60-second window of different types of sleep. The number of peaks can reflect the patient’s breathing frequency in one minute. A patient has an average of 12 times of inhalation and exhalation during normal sleep, which is as excepted. During hypopnea, the frequency of sleep depends on the severity of the hypopnea event. If the hypopnea is severe enough to be similar to apnea, which means the
<table>
<thead>
<tr>
<th></th>
<th>PSG $^a$</th>
<th>ApneaDetector $^a$</th>
<th>$MAE^{a,b}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Validation</td>
<td></td>
</tr>
<tr>
<td>Total Sleep time (min)</td>
<td>334.5 ± 58.3</td>
<td>346.6 ± 59.2</td>
<td>24.9</td>
</tr>
<tr>
<td>Sleep Efficiency (%) $^c$</td>
<td>81.4 ± 7</td>
<td>84.2 ± 5.3</td>
<td>5.1</td>
</tr>
<tr>
<td>Accuracy (%) $^a$</td>
<td></td>
<td>85.3 ± 5.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Sleep time (min)</td>
<td>341.5 ± 37.3</td>
<td>353.8 ± 41.2</td>
<td>27.2</td>
</tr>
<tr>
<td>Sleep Efficiency (%)</td>
<td>81.8 ± 8.6</td>
<td>84.2 ± 7</td>
<td>6.9</td>
</tr>
<tr>
<td>Accuracy (%) $^a$</td>
<td></td>
<td>85.1 ± 7.4</td>
<td></td>
</tr>
</tbody>
</table>

$^a$: Mean ± SD.
$^b$: Mean Absolute Error
$^c$: Sleep efficiency = Total sleep time / Total time in bed

Table 5.4: Sleep parameters scored by PSG vs. ApneaDetector

Patient’s breathing is so shallow that peaks are not produced, then the patient’s breathing rate will decrease. If the hypopnea is not so severe, and the patient still has breathing, then the respiratory frequency does not change very much. On the other hand, the average respiratory rate during sleep apnea is low as 6. The complete disappearance of breathing will significantly reduce the breathing frequency, thereby reducing the number of peaks on acceleration data.

Figure 5.6 shows the mean of the peak amplitude of different types of sleep. The amplitude of peaks during normal sleep is expected to be small, since the accelerometer data on the wrist have regular pattern due to the repeated cycles of inhalations and exhalations. By contrast, the accelerometer data varies drastically during sleep apnea, resulting in a significant standard deviation value. In hypopnea, although the subject’s breathing becomes shallow, it does not change that much as sleep apnea and thus does not produce a substantial standard deviation like sleep apnea.

Figure 5.7 shows the maximum standardized residual of different types of sleep. As can be seen, apnea events have the most significant maximum standardized residual on average. This is because the spikes generated by apnea.
events greatly affect the maximum standardized residual. The acceleration data of normal sleep is relatively average; thus, the maximum standardized residual is relatively small. On the other hand, some hypopnea events generate spikes while others not, so the average maximum standardized residual is larger than normal sleep.

5.3.3 Overall performance

We evaluate the performance of different classification methods by using precision (PPV) and recall (TPR). The precision and recall can be calculated as

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

After extracting features discussed in Section 4.3, we implement four commonly used classification algorithms, Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM). Since the AHI index is calculated by the summation of the number of sleep apnea and the
number of hypopneas, we combine apnea events and hypopnea events and label them as Apnea. For other sleep data without apnea events, we label them as Normal. We spilt the whole dataset with 70% of training set and 30% of testing set. The window size is 60 seconds. We also run a k-Fold Cross-Validation to eliminate the resulting bias.

<table>
<thead>
<tr>
<th>Sleep Status</th>
<th>NB PPV</th>
<th>NB TPR</th>
<th>DT PPV</th>
<th>DT TPR</th>
<th>RF PPV</th>
<th>RF TPR</th>
<th>SVM PPV</th>
<th>SVM TPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0.94</td>
<td>0.97</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.96</td>
</tr>
<tr>
<td>Apnea</td>
<td>0.95</td>
<td>0.90</td>
<td>0.97</td>
<td>0.96</td>
<td>0.97</td>
<td>0.96</td>
<td>0.94</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table 5.5: The performance of different classification approaches.

As shown in Table 5.5, Random Forest outperforms all the other schemes, and 98% of Normal precision and 96% of Apnea recall, meaning that the false positive rate of Apnea is only 4%. Decision Tree can also recognize different kinds of sleep status, with 97% of Normal precision and 96% of Apnea recall. In ApneaDetector, we choose Random Forest as the classifier.
Figure 5.7: Maximum standardized residual of different types of sleep.

Table 5.6: Performance of different spikes detection methods

<table>
<thead>
<tr>
<th></th>
<th>Use ARIMA model</th>
<th>Use mean directly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.9868</td>
<td>0.9860</td>
</tr>
<tr>
<td>Running Time (sec)</td>
<td>26543</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Effect of spikes detection method

We mentioned two ways to get the predicted value to help us detect spikes, use mean value and use ARIMA model. To decide which one we use, we choose one patient and apply two different methods on the whole night data. We use `tsoutlier` package in R to help us fit the ARIMA model. A total number of 456 ARIMA models are fitted. As shown in Table 5.6, both methods have the same accuracy, but the average time for fitting one ARIMA model takes about one minute while using mean value directly takes up negligible time (i.e. about 1 ms per window). The aim of ApneaDetector is to be a lightweight application that does not need too much CPU computation, so we choose use mean value as the predicted value to detect spikes.
Effect of window size

Different window sizes may affect the prediction result. Table 5.7 shows the mean, median and 97% percentile of OSA, CSA, and hypopnea events duration. It can be seen that almost all OSA, CSA and hypopnea events are around 30 seconds.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>97% Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSA</td>
<td>17.10</td>
<td>15.94</td>
<td>27.5</td>
</tr>
<tr>
<td>CSA</td>
<td>16.94</td>
<td>16.56</td>
<td>26.4</td>
</tr>
<tr>
<td>Hypopnea</td>
<td>19.25</td>
<td>18.13</td>
<td>33.7</td>
</tr>
</tbody>
</table>

Table 5.7: Mean, median and 97% percentile of event duration

Since the critical feature of detecting sleep apnea events are spikes after the end of the events, we need to make sure the spikes are within the window. Figure 5.8 shows the CDF of the distance between the end of apnea events and their corresponding spikes. We can see that about 90% of spikes occur within ten seconds after the end of the events. So we need at least an additional ten seconds after the events.

![Figure 5.8: CDF of the distance between end of OSA events and spikes.](image)

Also, the window in PSG study is divided in 30 seconds, meaning that in ApneaDetector we need to choose a multiple of 30 seconds as the window size. By combining all these reasons, we choose 60 seconds as the window size. A 60-second window can make sure it contains the whole apnea event and the corresponding spikes it occurs.
Effect of classification

When we use machine learning algorithm to do classification, we have two choices. We can combine sleep apnea and hypopnea events together as Apnea, we can also distinguish them. Table 5.8 shows the precision, recall and f1-score of classification result when using Random Forest. Compared with Table 5.5, normal sleep has the same classification result but apnea and hypopnea have lower accuracy.

To explain the reason for this, we plot the distribution of apnea events, hypopnea events, and normal sleep when considering the number of peaks and the maximum standardized residual. As can be demonstrated in Figure 5.9, normal sleep has a larger number of peaks than both apnea and hypopnea events. The mean number of peaks of normal sleep is 14.1, while this number is 6.5 in apnea and 7.8 in hypopnea events. On the other hand, apnea events have a larger maximum standardized residual than normal, the mean value of maximum standardized residual is 4.45 in apnea and 3.73 in hypopnea events, but 3.08 in normal sleep. From this plot, we can also see that although it is relatively easy to distinguish between apnea events and normal sleep, it is hard to differentiate between apnea and hypopnea events. Many hypopnea events and apnea events have a similar number of peaks as well as maximum standardized residuals. This explains why we have many misclassifications between apnea and hypopnea events, while normal sleep remains high accuracy.

<table>
<thead>
<tr>
<th>Sleep Status</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPV</td>
</tr>
<tr>
<td>apnea</td>
<td>0.69</td>
</tr>
<tr>
<td>hypopnea</td>
<td>0.72</td>
</tr>
<tr>
<td>normal</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table 5.8: The classification result of three classes.
Figure 5.9: Peak vs. Residual
Chapter Six

Conclusion and Future Work

6.1 Conclusion

In this thesis, we developed a novel system based on smartwatch to monitor user’s sleep quality. The system can estimate the user’s respiratory rate and detect sleep apnea by analyzing acceleration data on the wrist collected by the accelerometer sensor in smartwatch. By leveraging ARIMA models and training different machine learning models, we can assess the sleep quality. We compare the pros and cons of traditional ARIMA models, and supervised classification algorithms. Through extensive experiments, we validate the usability and effectiveness of our smartwatch based system.

6.2 Future work

In the future, we can extend the current work in the following directions:

- **Multi-sensors.** In this thesis, we utilize the accelerometer sensor in smartwatch to detect sleep apnea. However, many other factors such as electroencephalogram (EEG) and oxygen saturation (SpO2) are used to detect sleep apnea. Although such data can be extracted from PSG, there is no relevant sensor on smartwatch yet. Fortunately, some smartwatch manufacturers are developing this hardware [29]. The detection
of sleep apnea can be more accurate if our system can be enhanced by these sensors in the future.

- **Devices choice.** Although wearing a smartwatch is convenient compared to PSG equipment and do not require patients to visit a hospital or sleep center, there still exist some problems. For example, if patients change wrist position during sleep, this may cause disturbances in acceleration data, which may cause errors in the measurement results. Some researchers are seeking for more lightweight and stable smart devices. This could eliminate errors caused by smartwatch and achieve higher accuracy.

- **More sleep disorders classification.** In this thesis, we only show how to detect central sleep apnea, obstructive sleep apnea, and hypopnea. However, there are other sleep disorders like respiratory effort–related arousal (RERA). These are arousals in sleep that do not meet the definition of sleep apnea events. We can use more sensors in the smartwatch to detect these non-sleep apnea conditions.

- **More sophisticated model.** In this thesis, we choose supervised machine learning model to classify and predict sleep apneas. Recently, more sophisticated machine learning models such as deep learning models are more and more popular. In the future, we will study deep learning models to achieve better results.
Bibliography


