ApneaDetector: Detecting Sleep Apnea With Smartwatches

XIANDA CHEN, The Pennsylvania State University
YIFEI XIAO, The Pennsylvania State University
YEMING TANG, The Pennsylvania State University
JULIO FERNANDEZ-MENDOZA, The Pennsylvania State University
GUOHONG CAO, The Pennsylvania State University

Sleep apnea is a sleep disorder in which breathing is briefly and repeatedly interrupted. Polysomnography (PSG) is the standard clinical test for diagnosing sleep apnea. However, it is expensive and time-consuming which requires hospital visits, specialized wearable sensors, professional installations, and long waiting lists. To address this problem, we design a smartwatch-based system called ApneaDetector, which exploits the built-in sensors in smartwatches to detect sleep apnea.

Through a clinical study, we identify features of sleep apnea captured by smartwatch, which can be leveraged by machine learning techniques for sleep apnea detection. However, there are many technical challenges such as how to extract various special patterns from the noisy and multi-axis sensing data. To address these challenges, we propose signal denoising and data calibration techniques to process the noisy data while preserving the peaks and troughs which reflect the possible apnea events. We identify some special characteristics of sleep apnea such as signal spikes which can be captured by smartwatch, and propose methods to extract proper features to train machine learning models for apnea detection. Through extensive experimental evaluations, we demonstrate that our system can detect apnea events with high precision (0.9674), recall (0.9625), and F1-score (0.9649).

CCS Concepts: • Human-centered computing → Ubiquitous and mobile computing systems and tools; Mobile devices.

Additional Key Words and Phrases: Mobile health, apnea detection, smartwatch, signal denoising, data calibration

ACM Reference Format:
the subject, which increases the cost and the waiting time. The high cost and the inefficiency of the existing sleep apnea diagnose system have prevented many people from receiving affordable and accessible services.

With the rapid development of wearable devices and wireless technologies, various approaches for sleep monitoring and sleep apnea detection have been proposed. Radio frequency and sonar based techniques [15, 16, 30, 52, 54] have been proposed to detect chest movement due to breathing by monitoring the wireless or sound signal, i.e., the channel state information or the shift in carrier frequency. Although these approaches can monitor respiratory rate or even sleep apnea, the performance is sensitive to environmental changes, and most studies are performed in controlled environments. Recently, commercial wrist-worn devices such as Fitbit have been released, which are able to track users’ sleep. However, they can only record some coarse-grained sleep data such as sleep duration and body movement. Other researchers [6, 27, 46] have shown that using merely a single smartwatch, it is possible to capture a rich amount of information about sleep context and sleep events, including respiratory rate, body position, and body movement. However, none of them focuses on detecting sleep apnea with smartwatch.

Different from the aforementioned work, in this paper, we propose a smartwatch-based system called ApneaD- etector, which exploits the built-in sensors in smartwatch to detect sleep apnea. During sleep, respiration leads to the periodic movement of the chest, arms, and wrists. The sensors in smartwatch can capture such movement, and then detect sleep apnea. Through a clinical study, we identify some special characteristics of sleep apnea captured by smartwatch, which can be leveraged by machine learning techniques to detect sleep apnea. However, since the wrist movement is very subtle and noisy, there are many technical challenges such as how to extract various special patterns from the noisy and multi-axis sensing data. To address these challenges, we propose signal denoising and data calibration techniques to process the raw accelerometer data, smoothing away noise while preserving the peaks and troughs which reflect the respiratory cycles and possible apnea events. Based on our clinical study, we found that the subject suffering from sleep apnea will more likely make one or several intense breaths at the end of a sleep apnea event due to lack of oxygen. Such intense breaths will cause a signal spike in the accelerometer data. We propose an efficient and effective method to identify and quantify such signal spike, which can be used as an important feature for detecting sleep apnea.

The main contributions of the paper are as follows.

- Through a clinical study of twenty subjects, we demonstrate the feasibility of detecting sleep apnea using merely a single smartwatch.
- We identify some special characteristics of sleep apnea such as signal spikes captured by smartwatch, which can be leveraged by machine learning techniques to detect sleep apnea.
- We propose signal denoising and data calibration techniques to process the noisy data, and propose efficient and effective methods to extract proper features for apnea detection.
- Through extensive experimental evaluations, we demonstrate that apnea events detected by our system is highly correlated with the ground truth.

The rest of the paper is organized as follows. We introduce the background and motivation in Section 2. We present the design of ApneaDetector in Section 3. Section 4 presents the evaluation results, and Section 5 presents related work. More discussions are provided in Section 6, and Section 7 concludes the paper.

2 BACKGROUND AND MOTIVATION

Traditionally, polysomnography (PSG) is the standard clinical test to diagnose sleep apnea. In PSG, the technician attaches the subject with many sensors, including a nasal pressure transducer to measure the airflow, a pulse oximeter to measure oxygen saturation, EEG sensors to measure brain activity, and so on. These sensors collect the data when the subject is in sleep, based on which the technician monitors the subject’s sleeping patterns and diagnose sleep apnea events.
There are three kinds of sleep apneas [42]. An Obstructive Sleep Apnea (OSA) event occurs when there is a complete or partial blockage of the upper airway during sleep. The subject tries to pull air into the lungs; however, the air does not reach the lungs because of blockage. A Central Sleep Apnea (CSA) event occurs when the subject holds his/her breath for a non-negligible duration, usually more than ten seconds. During central apnea, the chest movements are flat indicating the absence of breathing effort. A hypopnea event occurs when the subject’s breathing becomes shallow. More specifically, the amplitude of the chest movement will drop by more than 30%. The apnea-hypopnea index (AHI) is the key metric used for sleep apnea diagnosis, which is defined as the average number of apnea and hypopnea events per hour of sleep. The AHI values are categorized as normal (<5), mild sleep apnea (5 - 15), moderate sleep apnea (15 - 30), and severe sleep apnea (>30). Figure 1 shows the airflow measured by the nasal pressure sensor for different kinds of sleep apneas.

**Motivation.** Although technicians can easily identify sleep apnea events by examining the signals collected from various sensors in PSG test, it is challenging to detect sleep apnea using only accelerometer sensor in a smartwatch. To demonstrate the feasibility, with approval by the Penn State Institutional Review Board (IRB), we have conducted a clinical study with twenty subjects at Penn State Hershey Sleep Research & Treatment.
Case. In the clinical study, each subject was prescribed to undergo a regular PSG test. The subject also wore a smartwatch to collect the accelerometer data of the wrist movement.

Figure 2 shows the accelerometer data of the wrist movement under different sleep apnea. In general, respiration leads to the periodic movement of the chest, abdomen, arms, and wrists, which can be captured by the accelerometer sensor in smartwatch. Figure 2(a) shows the accelerometer data during an OSA event (corresponding to Figure 1(a)), between the 12th and 31st seconds as labeled by the technician. In OSA, the airflow is blocked, and the subject will not have enough oxygen. In many cases, the subject is likely to make one or several intense breaths before returning to normal breathing. As shown in Figure 2(a), such intense breaths will cause a large vibration (i.e., a signal spike) in the accelerometer data. A similar spike can also be seen in hypopnea as shown in Figure 2(c). As shown in Figure 2(b), there is no spike because the airway remains open in CSA. Since the subject holds breath during the CSA event, the accelerometer data are flat.

Based on our clinical study, 92% of OSA events generate such signal spikes, and 70% of hypopnea events generate spikes. As a result, by detecting these spikes, most of the OSA and hypopnea events can be detected. Since the wrist movement is very subtle, the collected sensor data is very noisy. To identify the useful signal, we have to address technical challenges on how to extract such special patterns like spikes from the noisy, temporal, and multi-axis sensing data. We also need to identify other features to detect sleep apnea events such as CSA that do not have spikes. We will present the technical details for detecting sleep apnea in the next section.

3 APNEADETECTOR

In this section, we present the design of ApneaDetector. We first present techniques for signal denoising, data calibration, and then present techniques to extract proper features for apnea detection.

3.1 Signal Denoising

During sleep, the respiration related periodic movement caught by the accelerometer data is weak and easy to be dominated by noise. To extract respiratory signal from the weak and noisy accelerometer data, we design a filter to process the raw accelerometer data, smoothing away noise while preserving the peaks and troughs which reflect the respiratory cycles and possible apnea events. The commonly used filter in digital signal processing is the moving average filter, mainly because it is the easiest digital filter to understand [45]. However, this method may also smooth out the spikes, as shown in the blue line in Figure 3.

To preserve the spike, we use the Total Variation filter (TV filter) [41]. Let \(\mathbf{d} \in \mathbb{R}^{n \times 1}\) denote a series of raw accelerometer data in a sampling window, \(\tilde{\mathbf{d}} \in \mathbb{R}^{n \times 1}\) denote the filtered data, and \(d_i\) and \(\tilde{d}_i\) denote the \(i^{th}\) sample in \(\mathbf{d}\) and \(\tilde{\mathbf{d}}\) respectively. Then, the TV filter is to find an appropriate \(\tilde{\mathbf{d}}\) which minimizes the total variation, as
expressed in Equation 1,
\[
\hat{d} = \arg \min_{\tilde{d}} \left[ E(d, \tilde{d}) + \lambda V(\tilde{d}) \right]
\]
(1)

where \( E(d, \tilde{d}) \) measures the closeness between the filtered data \( \tilde{d} \) and the raw data \( d \), i.e., \( E(d, \tilde{d}) = \frac{1}{n} \sum_{i=1}^{n} (d_i - \tilde{d}_i)^2 \); the total variation \( V(\tilde{d}) \) describes the fluctuation in the filtered data \( \tilde{d} \), i.e., \( V(\tilde{d}) = \sum_{i=2}^{n} |\tilde{d}_i - \tilde{d}_{i-1}| \); and the regularization parameter \( \lambda > 0 \) controls how smoothing \( \tilde{d} \) is. If \( \lambda \) is too small, the filtered data will be close to the original data but with very little noise being removed, as shown as the red line in Figure 3(a); if \( \lambda \) is too large, the filtered data will be smooth but less like the original data (Figure 3(b)). As shown in Figure 3(c), when \( \lambda = 0.2 \), the TV filter can reduce noise and preserve the signal spike. We set \( \lambda = 0.2 \) in ApneaDetector.

The optimization problem in Equation 1 can be solved by the following iterations:
\[
\begin{align*}
\tilde{d}_{i+1} & = d - A^T z_i \\
\end{align*}
\]
\[
\begin{align*}
\tilde{z}_{i+1} & = \text{clip} \left( z_i + \frac{1}{\lambda} A \hat{d}_{i+1}, \frac{\lambda}{\lambda} \right)
\end{align*}
\]
(2)

where \( i \) is the iteration index, the matrix \( A \) is defined as
\[
A_{(n-1) \times n} = \begin{bmatrix}
-1 & 1 & & \\
-1 & 1 & & \\
& & & \\
& & & \\
& & & \\
-1 & 1 & & \\
\end{bmatrix},
\]
and the iterative clipping function is defined as
\[
\text{clip}(b, \beta) = \begin{cases} 
\beta \times \text{sign}(b) & \text{if } |b| \geq \beta \\
\beta \times \text{sign}(b) & \text{if } |b| \leq \beta
\end{cases}
\]

The optimization problem in Eq. 1 is convex and then the algorithm converges with any initialization. Thus, we can simply set \( z_0 = 0 \) for the initialization, and solve the problem by using Equation 2 iteratively to filter out noise in the raw accelerometer data.

### 3.2 Data Calibration

In general, the collected accelerometer data should be sinusoidal due to the repeated chest and then wrist movement. However, for some CSA events, the data captured by the smartwatch does not always look like that shown in Figure 2(b). For example, Figure 4(a) shows collected accelerometer data of a CSA event after signal denoising. At first glance, such data does not look like a sleep apnea event; it does not have a regular shape and should be discarded. After looking into the details of this data, we found that some features are embedded in the data. Although such features are hard to be recognized and connected to any sleep apnea event, they are actually related to CSA events after some data calibrations, as shown in Figure 4(b). Next, we explain how to identify this kind of irregular data and transform it to a recognizable sleep event pattern.

As shown in Figure 4(a), the data has some trend effects. Such trend effects might occur when the subject experiences subtle changes in wrist movement. Since the sleep event pattern is hidden because of the trend effects, we should first identify and remove such trend effects to identify the sleep event pattern. One simple solution to identify such trend effects is to calculate the moving average of the data, and the time series data is identified to have trend effects if the moving average monotonically rises/declines. However, this method is not flexible and has many limitations. Its performance depends on the size of the moving average window, which is hard to determine, and may result in oscillatory fluctuations. Moreover, if the average of the time series data is
not constant over time, this time series data has trend effects. Unfortunately, it is hard to quantify if the moving average is constant or not, and thus this method may result in errors.

To address aforementioned problems, we propose to identify such trend effects using the augmented Dickey–Fuller (ADF) test [50]. The ADF test is a common statistical method to test whether a given time series data is non-stationary or not, i.e., whether the accelerometer data has time-dependent trend or not. In ADF test, the accelerometer data can be represented by an autoregressive model:

$$y_t = c + \beta t + \alpha y_{t-1} + \sum_{i=1}^{l} \phi_i \Delta y_{t-i} + \epsilon_t$$

where $c$ is a constant, $\beta$ is the coefficient on a time trend, $\alpha$ is the coefficient on the sample at time $t - 1$, $l$ is the lag order of the autoregressive process, $\Delta$ is the first difference operator (i.e., $\Delta y_t = y_t - y_{t-1}$), and $\epsilon_t$ is the error term. The null hypothesis of the test is that the model is non-stationary ($\alpha = 1$), i.e., the accelerometer data is time dependent. The ADF test returns a $p$-value, which is the evidence against the null hypothesis. The smaller the $p$-value, the stronger the evidence to reject the null hypothesis. In ApneaDetector, for each window, the ADF test is applied to determine whether the accelerometer data has time-dependent trend. When performing the ADF test, the Akaike information criterion (AIC) is the criteria used in ADF to estimate relative quality of statistical models (i.e., different parameter settings for the autoregressive model) for the given accelerometer data. The ADF test returns a $p$-value of the best fitted model. In ApneaDetector, if the $p$-value is lower than the critical value (i.e., 0.05), the accelerometer data does not have time-dependent trend. Otherwise, the accelerometer data has time-dependent trend, and need to be calibrated.

After examining the apnea events that have time-dependent trend, we found that the accelerometer data for these events all presents a linear trend (i.e., upward or downward). To remove this linear trend in the accelerometer data, we apply the first-order differencing technique, which is commonly used to eliminate a linear trend in a time series data [43]. Taking Fig 4(a) as an example, the ADF test returns the $p$-value of 0.326 that is much higher than the critical value (0.05), and thus the accelerometer data has time-dependent trend and should be calibrated. Figure 4(b) shows the calibrated data after applying the first-order differencing to remove the trend effect. It can be validated that the calibrated data does not have time-dependent trend, as the $p$-value for Figure 4(b) is less than 0.01.

Fig. 4. Data calibration: (a) collected accelerometer data, (b) data after removing the trend effects (i.e., subtracting the current value from the previous value).
3.3 Feature Extraction and Classification

In this subsection, we present the technical details on how to extract the proper features for training the classifiers to detect sleep apneas.

3.3.1 Spike Detection. As shown in Figure 3(a) and Figure 3(c), the subject suffering from OSA and hypopnea will more likely make one or several intense breaths due to lack of oxygen. Such intense breaths will cause a signal spike in the accelerometer data, which can be used as an important feature for identifying OSA and hypopnea events. In what follows, we first show that the conventional threshold-based method is inapplicable to detect spike for apnea detection, and then propose a statistical analysis based method to quantify the existence of spikes without using a fixed threshold.

Traditionally, identifying spikes in time series data can be considered as a binary problem, i.e., is there any sample in the data greater than a threshold (e.g., two standard deviation over the mean [13])? However, using a threshold-based solution to detect the spike may not be the best option, since it is hard to set a constant threshold considering subjects have different sleep patterns. In addition, identifying the accelerometer sample of extremely high amplitude as spike does not meet the objective of apnea detection, i.e., using a threshold-based solution to detect spikes may mis-identify a normal sleep as a sleep apnea. Figure 5 shows the accelerometer data (after being denoised) of a normal sleep event. As shown in the figure, since the accelerometer data during normal sleep only has minor changes (± 0.02) along the mean value, its standard deviation is very small, i.e., the threshold for identifying signal spike is very small. Then, using the threshold-based solution may misclassify normal sleep as a spike, as shown in Figure 5.

To address this problem, we propose the following residual based technique to quantify the existence of spikes without using a fixed threshold. The residual of the accelerometer sample at time \( t \) is defined as the difference between the observed value and predicted value:

\[
r_t = y_t - \hat{y}_t
\]

where \( y_t \) denotes the \( t^{th} \) sample and \( \hat{y}_t \) denotes the predicted value. Similar to the aforementioned threshold-based solution, simply using residuals to detect spikes is difficult, since their magnitude depends on the scale of the measured data. We can eliminate the units of measurement by standardizing the residual as follows:

\[
sr_t = \frac{r_t}{std(r)}
\]

(3)

where \( r_t \) is the residual for the \( t^{th} \) sample and \( std(r) \) is the standard deviation of all residuals in a window.

To obtain the predicted value \( \hat{y}_t \), we propose an efficient and effective method to calculate the standardized residual. Since the acceleration data is sinusoidal and does not have trend (or being removed in data calibration), the mean value (i.e., the baseline of the data) can be used to represent the predicted value. Similarly, the standard deviation of the accelerometer data can be used to represent the standard deviation of the residuals in a window.
Thus, the maximum standardized residual can be expressed as follows.

$$\text{max}_sr = \max_{t \in w} \frac{|y_t - \text{mean}(w)|}{\text{std}(w)}$$  \hspace{1cm} (4)

For a window of accelerometer data, we use the maximum standardized residual ($\text{max}_sr$) to represent the existence of a sleep apnea (OSA and hypopnea) event. To better differentiate between normal sleep and sleep apnea, we should maximize the difference between $\text{max}_sr$ of normal sleep events and that of sleep apnea events. However, the $\text{max}_sr$ value calculated by Equation 4 may not well differentiate sleep apnea events from normal sleep events, since the whole window is used to calculate the standard deviation in the denominator. This window includes the signal spikes and normal breathing pattern besides the sleep apnea part, which can significantly increase the standard deviation (i.e., denominator), and then decrease the difference between the $\text{max}_sr$ of normal sleep events and sleep apnea events.

To solve this problem, we refine the definition of the maximum standardized residual, by considering the special characteristics of sleep apnea, i.e., there is a subwindow of flat signal samples prior to the signal spike, as shown in Figure 6. Figure 6(a) shows the annotated sleep apnea event starting from the 12th second to 32nd second. Figure 6(b) shows the accelerometer data of the sleep apnea event in Figure 6(a), where there is a subwindow of relatively flat signal samples, and the signal spike occurs after the sleep apnea event. There is a lagging time between the end of the sleep apnea event and the signal spike, because the subject is likely to make intense breath due to lack of oxygen after the sleep apnea event. More formally, Equation 4 is refined as follows by considering this subwindow.

$$\text{max}_sr = \frac{|y^* - \text{mean}(\tilde{w})|}{\text{std}(\tilde{w})}$$  \hspace{1cm} (5)

where $y^*$ is the signal spike and $\tilde{w}$ is the subwindow used for calculating the standard deviation. For a window of accelerometer data, we first identify the signal spike that has the highest magnitude, i.e., $y^* = \max_{t \in w} \sqrt{a_{x,t}^2 + a_{y,t}^2 + a_{z,t}^2}$, and then determine the subwindow for calculating the maximum standardized residual.

**Determining Subwindow Size and Lagging Time.** To determine the proper subwindow, in ApneaDetector, we need to setup the lagging time (i.e., where the subwindow ends) and the subwindow size, which are important...
Table 1. Mean, median and 97% percentile of the duration of different sleep apnea events.

<table>
<thead>
<tr>
<th></th>
<th>Mean (sec)</th>
<th>Median (sec)</th>
<th>97% Percentile (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSA</td>
<td>17.1</td>
<td>15.9</td>
<td>27.5</td>
</tr>
<tr>
<td>CSA</td>
<td>16.9</td>
<td>16.6</td>
<td>26.4</td>
</tr>
<tr>
<td>Hypo</td>
<td>19.6</td>
<td>18.3</td>
<td>33.7</td>
</tr>
</tbody>
</table>

Fig. 7. (a) CDF of the lagging time in sleep apnea. (b) Determining the proper subwindow and lagging time in ApneaDetector.

Factors in calculating the maximum standardized residual. Figure 7(a) shows the CDF of the lagging time in sleep apnea events based on the data collected from our clinical study. As can be seen, the lagging time ranges from 0 to 15 seconds. Table 1 shows the mean, median and 97% percentile of the duration of different sleep apnea events, where most sleep apnea events last less than 35 seconds. Since a sleep apnea event lasts for at least 10 seconds, the subwindow size ranges from 10 to 35 seconds.

Based on the data collected from our clinical study, Figure 7(b) shows the relationship among subwindow, lagging time, and the difference of the maximum standardized residual (i.e., Equation 5) between sleep apnea events and normal sleep events. Generally speaking, the difference of max_sr increases as the subwindow size decreases. The difference of max_sr reaches maximum when the subwindow is ten seconds, which is the minimum sleep apnea duration. When the subwindow is 10 seconds, the difference of max_sr reaches maximum when the lagging time is seven seconds. Thus, in ApneaDetector, the subwindow size is set to be ten seconds and the lagging time is set to be seven seconds.

We use the maximum standardized residual (max_sr) to represent the existence of a sleep apnea (OSA and hypopnea) event. Figure 8(a) shows the max_sr value by using the subwindow based method for different types of sleep activities in the data collected in our clinical study. As can be seen, the max_sr value for sleep apnea, especially OSA events, is much larger than that of normal sleep. This is because the residual of the signal spike is very high and the standard deviation of the signal samples in the subwindow (i.e., during the apnea event) is very small. For hypopnea events, the max_sr value is smaller than that of OSA because the signal during the event is not flat but is a sinusoidal wave with small amplitude. Figure 8(b) shows the max_sr value using the mean based method (i.e., Equation 4). Although max_sr of different types of sleep events still follow similar pattern, the difference among them is not as obvious as that in Figure 8(a). Thus, in ApneaDetector, we choose the subwindow based method for calculating the max_sr value.
3.3.2 Peak Detection. The maximum standardized residual can be used to detect spikes in OSA and hypopnea events. However, CSA does not have spikes and other features should be identified for detecting such events. The accelerometer data during normal sleep follows a periodic sinusoidal wave, and its amplitude reduces to almost zero during a CSA event. When a CSA event occurs, the distance between two consecutive peaks of the sinusoidal wave right before and after the CSA event becomes larger than ten seconds. To capture this pattern, we design a peak detection algorithm to track and measure the distance between two consecutive peaks.

Traditional algorithms [40] detect peaks by identifying the transition point at which a signal changes from an uptrend to a downtrend. It finds all local maxima by simply comparing with neighboring signals. However, this algorithm may result in errors during CSA events. Figure 9(a) shows that the traditional peak detection algorithm keeps a number of unintended peaks in the accelerometer data of the CSA event in Figure 2(b). Note that the accelerometer data has been denoised using TV filter.

To address this problem, the peak detection algorithm can be enhanced by considering the following two aspects. First, setting a threshold for the minimum amplitude of the detected peak. In ApneaDetector, the amplitude threshold is set 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. Second, setting a threshold on the minimum distance between two consecutive peaks, which is related to the respiratory rate [46]. In ApneaDetector, we set the distance threshold to be three seconds, because the normal respiratory rate for a healthy adult typically varies from 12 to 20 breaths per minute, i.e., three seconds at the maximum frequency [7, 11]. As shown in Fig 9(b), the enhanced peak detection algorithm identifies the correct peaks.

Based on peaks, we extract three features for Lost detection: distance between peaks, number of peaks, and amplitude of peaks. Figure 10 shows how these features represent different types of sleep activities based on the data collected in our clinical study.

Distance between peaks. The maximum distance between consecutive peaks (named peak_dis) in a window of accelerometer data is extracted as a feature for apnea detection. Figure 10(a) shows the peak_dis value for different types of sleep activities, and the peak_dis value for sleep apnea is much larger than that of normal sleep.

Number of peaks. Since the number of breaths can be reflected by the number of peaks in the accelerometer data, we extract the number of peaks (named peak_num) within a window as a feature. Figure 10(b) shows the number of peaks for different types of sleep activities. For normal sleep, peak_num is between 12 and 20, as the normal respiratory rate for a healthy adult typically varies from 12 to 20 breaths per minute [7, 11]. While the subject still breathes during a hypopnea event, the number of peaks is smaller than that during normal sleep. During OSA and CSA events, the subject makes no breathing and thus the number of peaks in the accelerometer data is much less.
Amplitude of peaks. We extract the standard deviation of the amplitude of peaks (named peak_amp) in a window as a feature for apnea detection. Figure 10(c) shows the peak_amp value for different sleep activities in the data collected in the clinic study. The peak_amp value during normal sleep is small, since the accelerometer data on the wrist has regular pattern due to the repeated cycles of inhalations and exhalations. By contrast, the accelerometer data varies drastically during sleep apnea, resulting in a large standard deviation value. In hypopnea, the subject’s breathing becomes shallow, and thus has a smaller standard deviation than OSA.

3.3.3 Multi-axis Correlation. The accelerometer data can be collected from X, Y, and Z axis, and more information can be extracted from the multi-dimensional data for apnea detection. For example, Figure 2(a) shows the collected accelerometer data corresponding to one OSA event, where signal spikes can be seen along three axes. However, although Y axis and Z axis have similar data pattern, X axis looks like the inverse of the other two. To make use of this information, we need to consider the data reliability from each axis and the data correlation among multiple axes.

Traditionally, co-variance can be used to measure the joint variability of the accelerometer data along two different axes. However, it is hard to interpret the meaning of this co-variance (between $-\infty$ and $+\infty$), because it is not normalized.

To address this problem, we use the normalized co-variance, i.e., the co-variance normalized by the product of their standard deviations. The normalized co-variance measures the correlation between the accelerometer data of two axes. It has a value between -1 to +1. The closer it is to +1 or -1, the more closely the two variables are related. The positive sign represents the direction of the correlation, i.e., if the accelerometer data of one axis

Fig. 9. Peak detection algorithms: (a) Traditional, (b) Enhanced.

Fig. 10. Peak features extracted in ApneaDetector. (a) Distance of peaks. (b) Number of peaks. (c) Peak amplitude.
Table 2. Features extracted in ApneaDetector.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_sr</td>
<td>the maximum standardized residual of the data along each axis</td>
</tr>
<tr>
<td>peak_dis</td>
<td>the maximum distance between two consecutive peaks along each axis</td>
</tr>
<tr>
<td>peak_num</td>
<td>the number of peaks of the data along each axis</td>
</tr>
<tr>
<td>peak_amp</td>
<td>the standard deviation of the amplitude of the peaks along each axis</td>
</tr>
<tr>
<td>cov_acc</td>
<td>the normalized co-variance between the data along each pair of axes</td>
</tr>
</tbody>
</table>

increases, the other one also increases. In ApneaDetector, the normalized co-variance between the accelerometer data along every two different axes is extracted as a feature.

3.3.4 Extracted Features and Classifiers. Table 2 shows the extracted features for apnea detection, and we have explained the reasons for choosing these feature in the previous subsections. To illustrate the feasibility of detecting sleep apnea events using these extracted features, we use features such as maximum standardized residual and number of peaks to visualize how they can be applied to differentiate these sleep events.

Figure 11 shows the feasibility of using extracted features to differentiate sleep apnea from normal sleep, with a two-dimensional scatter plot, where the horizontal dimension represents the number of peaks and the vertical dimension represents the maximum standardized residual of the accelerometer data along x-axis. As shown in the figure, normal sleep has more peaks than apnea and hypopnea. Specifically, the mean number of peaks for normal sleep, apnea, and hypopnea are 13.82, 7.9, and 8.7, respectively. From the other dimension, normal sleep has much lower maximum standardized residual than apnea and hypopnea. Specifically, the mean value of maximum standardized residual for normal sleep, apnea, and hypopnea are 4.32, 21.99, and 13.36, respectively.

Since AHI is the key metric used for sleep apnea diagnosis, ApneaDetector focuses on improving the accuracy of estimating AHI. When calculating AHI, it is not necessary to differentiate between different kinds of sleep apnea events, and thus OSA, CSA, and hypopnea events are combined as one category, in addition to the category of normal sleep events.

After features are extracted, different classification algorithms such as Decision Tree (DT) [38], Naive Bayes (NB) [31], Support Vector Machine (SVM) [48], Random Forest (RF) [32], and Adaptive Boost (ABT) [39], can be used to train classifiers for apnea detection. DT, NB, and SVM are simplex machine learning methods for classification, while RF and ABT are based on ensemble learning techniques, i.e., using multiple weak classifiers (i.e., decision trees) to make better predictions. RF leverages parallel ensembling technique, where each decision tree makes its own decision independent of each other, and the order of the decisions is not important. Thus, RF can run decision trees in parallel using multiple processors. By contrast, ABT uses a sequential ensembling technique, i.e., ABT runs weak classifiers iteratively, and the classifier outputs are weighed based on previous results. In ABT, the sequence of each weak learner can affect the prediction accuracy, and the sequential operations are time consuming. In Section 4, we will evaluate the performance of different classification algorithms, and select the best.
3.4 Identifying Sleep Duration

When using AHI for diagnosing sleep apnea, only sleeping time is counted. In PSG, whether the subject is sleeping or awake can be easily determined by measuring the brain activity based on the EEG signal. However, smartwatch does not have the EEG signal. Recent work [5] has shown that body movements can be used for identifying the sleep duration with acceptable accuracy for apnea detection. Therefore, we identify and remove the non-breathing body movement duration (i.e., being awake) when calculating the total sleep time.

The non-breathing body movement will result in large fluctuations in the accelerometer data, which can be detected. However, we cannot simply discard the accelerometer samples with amplitudes larger than a threshold, since sleep apnea can also generate high amplitude samples. Based on our observations from the clinical study, the acceleration data will be back to normal immediately after an apnea event, but the data anomalies will last much longer if the subject suffers insomnia or is awake. Thus, instead of considering samples one by one, we consider samples in segments, and leverage the fact that sleep status is temporarily correlated when identifying sleep duration. Each segment has 30 seconds of data, because in PSG the collected records are segmented into 30-second slots, each of which is labeled as sleep or wake by the technician.

To determine the sleep status, we propose a logistic regression model by considering not only the activity score in the current segment, but also the activity scores in previous four segments and the next two segments. The activity score for a segment is defined as the number of accelerometer samples whose total accelerometer value (i.e., \( \text{acc} = \sqrt{a_x^2 + a_y^2 + a_z^2} \)) is outside of the predefined threshold, which is set to \([9.8 \pm 3\%] \text{ m/s}^2\). This is because during motionless sleep, most of the accelerometer data is dominated by gravity, while for non-breathing body movements (i.e., being awake), the total accelerometer becomes much larger. The logistic regression model can be written as follows:

\[
\rho = 1 - \frac{1}{1 + Z} \\
Z = \exp(\beta_0 + \beta_{-4}A_{-4} + \beta_{-3}A_{-3} + \beta_{-2}A_{-2} + \beta_{-1}A_{-1} + \beta_0A_0 + \beta_1A_1 + \beta_2A_2)
\]  

(6)

where $A_0$ is the activity score of the current segment, $A_{-i}$ is the activity score of the $i^{th}$ segment prior to the current segment, and $A_i$ is the activity score of the $i^{th}$ segment after the current segment. The coefficients $\beta$’s are the parameters of the regression model, which will be determined in the next section.

4 PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of ApneaDetector based on the data collected in our clinical study.

4.1 Clinical Study

With approval by our Institutional Review Board (IRB), we conducted a clinical sleep study with twenty subjects at Penn State Milton S. Hershey Medical Center, where the subjects were prescribed to undergo the regular PSG test. That is, while undergoing the regular PSG test, each subject also worn the smartwatch (Huawei Watch 2 \cite{47, 51}) running ApneaDetector to collect the sensor data. The PSG test was performed according to the criteria of American Academy of Sleep Medicine \cite{35}, i.e., the subject is attached with a number of sensors and electrodes to measure multiple biophysiological signals, which are then used for sleep disorder diagnosis by the sleep physician. In the meanwhile, the technician will help the subjects wear and setup the smartwatch on their wrists. The smartwatch is fully charged before being given to the subjects, to make sure it can record the sensor data for whole night. The smartwatch and PSG equipment are synchronized with the Internet time, so that we can extract the corresponding period of the sensor data when apnea events occur. The PSG study reports (i.e., sleep apnea events labeled by the sleep physician) are used as the ground truth to evaluate the performance of ApneaDetector.

Twenty subjects (eight males and twelve females) participated in the clinical study. The subjects’ ages range from 36 to more than 72, with an average of 59.3. The subjects present certain diversity in terms of the severity of sleep apnea, i.e., some subjects have few sleep apnea events, while others have a lot. All subjects record the sensor data for around eight hours which is the average total time in bed.

Figure 12 shows the number of sleep apnea events collected for each subject. In total, there are 1018 OSA events, 125 CSA events, and 818 hypopnea events. Some subjects suffer from severe OSA but mild hypopnea, such as subject 18. Others have a large amount of hypopnea but less OSA events, such as subject 2. In general, the number of CSA is much smaller than that of OSA and hypopnea.

4.2 Apnea Detection

Before presenting the evaluation results of ApneaDetector, we first select the classifier that has the best performance. We evaluate different classification algorithms such as Decision Tree (DT) \cite{38}, Naive Bayes (NB) \cite{31},
ApneaDetector: Detecting Sleep Apnea With Smartwatches

15

DT NB SVM RF ABT 0.90 0.92 0.94 0.96 0.98 1.00 F1-score

(a)

Fig. 13. Performance comparison of different classifiers for apnea detection: (a) F1-score. (b) Time.

Table 3. The performance of ApneaDetector.

<table>
<thead>
<tr>
<th>Sleep Events</th>
<th>Random Forest</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apnea and Hypopnea</td>
<td></td>
<td>0.9674</td>
<td>0.9625</td>
<td>0.9649</td>
</tr>
<tr>
<td>Normal</td>
<td></td>
<td>0.9753</td>
<td>0.9778</td>
<td>0.9765</td>
</tr>
</tbody>
</table>

Random Forest (RF) [32], Support Vector Machine (SVM) [48], and Adaptive Boost (ABT) [39]. From clinical study, we set the window size to be 60 seconds and collected 1018 OSA events, 125 CSA events, 818 hypopnea events, and three thousand normal sleep events randomly sampled from all twenty subjects. The events are divided into two parts: one part (i.e., the training set) includes the events from ten subjects, and the other part (i.e., the testing set) includes the events from the other ten subjects. Note that a group based cross-validation scheme is used, where the sample data from subjects is isolated to the training set or the testing set. That is, the sample data from a subject will not appear in both the training set and the testing set. We apply 10-fold cross-validation on the training set to determine the best classifier for ApneaDetector.

Figure 13 shows the results of different classifiers in terms of F1-score and execution time. To avoid the effect of imbalanced class distribution in the sample data, F1-score is used as the performance indicator to determine the classifier for ApneaDetector. In Figure 13(a), the green triangle represents the mean and the yellow line represents the median. As shown in the figure, ABT and RF have the highest F1-score. As shown in Figure 13(b), RF is much faster than ABT. Thus, the RF approach is used in ApneaDetector.

Table 3 shows the performance of ApneaDetector in terms of precision, recall, and F1-score. For a certain type of event, its recall is defined as the ratio of the number of true positives (i.e., the events which are correctly identified as such type) to the number of actual positives (i.e., such type of events which are actually in the test set), and its precision is defined as the ratio of the number of true positives to the sum of the number of true positives and false positives (i.e., the events which are identified as such type but actually not). The F1-score is defined as the harmonic mean of precision and recall. Here, OSA, CSA, and hypopnea events are combined as one category, since the AHI index is calculated as the number of apnea (OSA and CSA) and hypopnea events per hour of sleep. As shown in the table, ApneaDetector can achieve high precision and recall, both are above 96%.

Since there is no existing work on using commercial smartwatches to detect sleep apnea, we compare ApneaDetector with other designs using different features. In sleep monitoring (e.g., respiration rate, sleep time, body
amplitude-based features have been used for training the classifiers. For example, in [6, 18, 46], the amplitude-based features are extracted for monitoring sleep quality, including mean, variance, and standard deviation of the accelerometer data along each axis, and the mean value of the total accelerometer of three axes. 

Figure 14 compares the performance of apnea detection using aforementioned amplitude-based features and the features extracted by ApneaDetector. As can be seen, ApneaDetector performs much better than the amplitude-based method. Specifically, the precision, recall, and F1-score for detecting sleep apnea events in ApneaDetector are above 96%, and the precision, recall, and F1-score in the amplitude-based method are around 76%.

### 4.3 AHI Estimation

The apnea-hypopnea index (AHI) is the key metric used for sleep apnea diagnosis, which is defined as the average number of apnea and hypopnea events per hour of sleep. To estimate the total sleep time, we fit the logistic regression model to identify the sleep status by considering the previous four segments and the next two segments. We select the trace data from ten subjects for training the regression model (i.e., Equation 6), and the rest ten subjects are used for validation, and they are also used for evaluating how well ApneaDetector estimates the AHI value.

Table 4 shows the coefficients of the logistic regression model. As shown in the table, the $p$-value of all the coefficients are very small (i.e., close to zero), meaning that all coefficients values are significant for the model. Thus, we choose these values as coefficients for our regression model. With the coefficients determined for the

| Coefficient | Value   | Std. Error | z-value | $p(|z|)$  |
|-------------|---------|------------|---------|----------|
| $\beta$     | -2.109743 | 0.039778   | -53.038 | < 2e-16  |
| $\beta_{-1}$| 0.027100  | 0.003114   | 8.703   | < 2e-16  |
| $\beta_{-2}$| 0.018043  | 0.003297   | 5.472   | 4.45e-08 |
| $\beta_{-3}$| 0.018606  | 0.003580   | 5.197   | 2.03e-07 |
| $\beta_{-4}$| 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 |

4.3 AHI Estimation

The apnea-hypopnea index (AHI) is the key metric used for sleep apnea diagnosis, which is defined as the average number of apnea and hypopnea events per hour of sleep. To estimate the total sleep time, we fit the logistic regression model to identify the sleep status by considering the previous four segments and the next two segments. We select the trace data from ten subjects for training the regression model (i.e., Equation 6), and the rest ten subjects are used for validation, and they are also used for evaluating how well ApneaDetector estimates the AHI value.

Table 4 shows the coefficients of the logistic regression model. As shown in the table, the $p$-value of all the coefficients are very small (i.e., close to zero), meaning that all coefficients values are significant for the model. Thus, we choose these values as coefficients for our regression model. With the coefficients determined for the
regression model (Equation 6), for a window of accelerometer data, ApneaDetector determines whether this window belongs to sleep or awake status. After that, ApneaDetector computes the sleep time by subtracting the duration of being awake from the total record time.

Figure 15 shows the Bland-Altman plot of the total sleep time of twenty subjects estimated by ApneaDetector, compared with the PSG ground truth. The x-axis represents the average value of the PSG ground truth and the ApneaDetector estimation, and the y-axis represents the difference between the PSG ground truth and the ApneaDetector estimation. The overall mean absolute error is 26.4 minutes, and the median is 18.5 minutes. The estimation error can be due to the following reasons. The subject may wake up for a very short period of time due to the effects of sleep disorders, the duration of which can be identified as awake using the fine-grained sensors in PSG, but may not be detected by ApneaDetector. The sleep time may be slightly overestimated for some subjects who woke up in the middle of the night and lay on the bed without frequent body movements. The brain activity during this period is high and can be identified by PSG, but the accelerometer data collected by the smartwatch remains normal. These are fundamental limitations of estimating total sleep time without monitoring brain activity with PSG. However, in what follows, we will show that the sleep time estimated by ApneaDetector is acceptable for calculating the AHI value, which is the key metric used for sleep apnea diagnosis.
We use the intra-class correlation coefficient (ICC) [25] to evaluate the performance of AHI estimation in ApneaDetector. In sleep research community, ICC is typically used to evaluate the performance of AHI estimation, i.e., to examine the correlation between the PSG ground truth and ApneaDetector. We use the sliding window technique to evaluate the whole night sleep trace (i.e., excluding the duration of being awake). The window is set to be 60 seconds and move at a step of 30 seconds; if sleep apnea events are detected in two consecutive windows, they are considered as one event. Figure 16 shows the scatter plot of the AHI values estimated by ApneaDetector and that of PSG. The PSG AHI is calculated by the actual number of apnea events divided by the actual sleep time, and the AHI of ApneaDetector is obtained by the number of identified apnea events divided by the estimated sleep time. As shown in the figure, the AHI values estimated by ApneaDetector and the PSG ground truth are highly correlated, with ICC value of 0.882. Thus, ApneaDetector is demonstrated to be able to accurately estimate the AHI index.

5 RELATED WORK

5.1 Sleep Monitoring

Various technologies including RFID, geophone, sonar, and WiFi (radio frequency) have been proposed for sleep monitoring. In [29], RFID tags deployed near the subject are used to collect respiration information for detecting sleep sound-activities (i.e., snore, cough, somniloquy). Researchers used a geophone mounted under the mattress to monitor heart rate [21] and respiratory rate [22]. DoppleSleep [36] tracks the body movement and estimates the breathing rate using a short-range Doppler radar. Lin et al. [28] utilize the Doppler radar sensor to differentiate various sleep stages and estimate the breathing rate. Hsu et al. [19] use the FMCW (frequency modulated continuous wave) radio to capture subject location and breathing, and then monitor the subject’s sleep and awake status. Yue et al. [53] utilize the FMCW radio to disentangle breathing signals from mixtures of breathing signals, and monitor sleep stage for multiple individuals. UbiBreathe [1] leverages the changes in the WiFi RSS (received signal strength) to monitor respiratory rate. FullBreathe [54] employs the fine-grained CSI (Channel State Information) information of WiFi signals for monitoring respiration rate. Zeng et al. [55] propose to monitor the respiration of multiple persons by leveraging the WiFi hardware equipped with multiple antennas. However, these approaches require either dedicated or customized hardware platforms.

There has been considerable research on sleep monitoring using smartphones. iSleep [15] uses the microphone on smartphone to record sleep-related acoustic events (e.g., body movement, cough and snore), and then measures the sleep quality. Other studies applying the microphone on smartphone for sleep monitoring include [14, 37], to name a few. In [3], a smartphone is bound on the subject’s chest and the accelerometer and gyroscope are utilized to extract the chest movement pattern for estimating respiratory rate.

In addition to smartphone-based solutions, there has also been extensive work on monitoring sleep activities using wearable devices such as smartwatch. SleepMonitor [46] monitors the respiratory rate and detect the body position during sleep using the built-in accelerometer on smartwatch. SleepGuard [6] tracks various sleep-related events using a single smartwatch, and identify factors affecting sleep quality. Pombo et al. [34] jointly consider heart rate, accelerometer, and sound signals collected from the smartwatch for sleep monitoring. Biowatch [17] monitors the subject’s respiratory rate and heart rate in a controlled laboratory study using built-in accelerometer and gyroscope on smartwatch. Kerz et al. [24] leverage both wearable device (i.e., Fitbit) and smartphone to detect early signs of relapse in psychosis. Zhai et al. [56] propose to classify sleep stage using multimodal sensing data from movement (wearable actigraphy) and cardiac sensors (ECG). There are also a number of smartwatch application for monitoring sleep activities, such as Sleeptracker [44] and FitBit [10]. These works focus on monitoring daily sleep activities, such as respiratory rate, sleep stage, body posture and movement, coughing, snoring. In ApneaDetector, our goal is to detect sleep apnea events using a single smartwatch.
5.2 Sleep Apnea Detection

Polysomnography (PSG) is the standard clinical test to diagnose sleep apnea. It is an expensive and time-consuming test that involves hospital visit, specialized wearable sensors, professional installation, and long waiting lists. Portable recording systems [5, 8, 20, 49] have been proposed for use in home settings, however, they still require the attachment of various sensors and extensive wiring, and lack some important data only available in PSG. With various built-in sensors available in smartphone, many researchers proposed smartphone-based systems for sleep apnea diagnosis. Kalkbrenner et al. [23] exploit the microphone on smartphone to monitor sleep quality and estimate sleep apnea. Alqassim et al. [2] explore the information of snoring and body movement to detect symptoms of sleep apnea. UbiBreathe [1] leverages the changes in the WiFi RSS (received signal strength) to monitor respiratory rate as well as apnea detection. Zhang et al. [57] use a pulsoximeter to sense the SpO2 signal and transmits it to the smartphone for diagnosing sleep apnea. Yang et al. [52] applies the mmWave signals to estimate the breathing rate and detect apnea events (only central apnea and hypopnea events) based on the received signal strength of the reflections. ApneaAPP [30] transforms a smartphone into an active sonar system that emits frequency-modulated sound signals and receives the reflections, based on which the information related to sleep apnea is extracted for apnea detection. Ferrer-Lluis et al. [9] propose a smartphone-based system to detect positional OSA, i.e., some patients sleep in the supine position is likely to encounter OSA events than other sleep positions. These smartphone-based systems either require extra hardware or require placing the smartphone at a specific location, e.g., at a particular angle or near the subject. When using radio frequency technique or sonar technique, the performance is sensitive to the environmental changes and most studies are performed in controlled environments. Different from them, ApneaDetector using smartwatches to detect apnea.

6 DISCUSSIONS

Although ApneaDetector performs well on classifying sleep events into normal sleep and sleep apnea based on the proposed features, there are many challenges on classifying sleep apnea into fine-grained categories, including OSA, CSA and hypopnea. This is because, in practice, the wrist movement captured by smartwatch is very subtle and noisy. With classical machine learning approaches, it is difficult to define the representative features to differentiate sleep apnea from hypopnea. As shown in Figure 11, the features (i.e., maximum standardized residual and number of peaks) extracted for apnea events and hypopnea events are mixed up. Then, sleep apnea and hypopnea events may be mis-classified.

In this paper, our main goal is to identify sleep apnea and hypopnea events for calculating AHI, the commonly used metric for sleep apnea diagnosis, and achieving such goal does not require fine-grained classification. However, classifying sleep apnea into fine-grained categories such as OSA, CSA and hypopnea is also important since it can provide doctors with more information on the severity of the sleep apnea and help diagnose other related diseases. Thus, in the future, we will study deep learning models which could extract both explicit and implicit patterns from the sensing data for classifying sleep apnea into fine-grained categories.

In ApneaDetector, the accelerometer data collected from the smartwatch is used to detect sleep apnea. However, when a user wears watch loosely, it is difficult to capture the respiration signal (the wrist movement) through the smartwatch’s accelerometer data, i.e., it is hard to detect respiration patterns since the accelerometer readings along three axes are too weak. To address this problem, we can use mean and variance-based thresholds to detect such situation and display a warning signal (or vibration) to let the user tighten the smartwatch.

In the clinical sleep study, we do not control how the subjects place their wrist during sleep. The subjects were asked to wear the smartwatch on the wrist, but there is no way to control the wrist movement during sleep. Since ApneaDetector detects sleep apnea based on the accelerometer data generated by wrist movement, its performance can be affected by the wrist placement. By analyzing the collected data, we find that the respiratory signal (i.e., the accelerometer data) is stronger if the wrist with smartwatch is on or close to the body, and the
signal will be weaker if the wrist is far away from the body. ApneaDetector relies on the proposed techniques to deal with all cases. Certainly, when the signal is too weak and embedded into the noise, the detection accuracy will be reduced. Also, a slight turning of the wrist might result in a sudden change in the acceleration data, which can be misidentified as respiratory activity [26]. Further research will be done on identifying better features to deal with such misclassifications.

7 CONCLUSIONS

In this paper, we designed and evaluated a smartwatch-based system called ApneaDetector, which exploits the built-in sensors in smartwatch to detect sleep apnea. Through a clinical study, we identified some special characteristics of sleep apnea captured by smartwatch, which can be leveraged by machine learning techniques for sleep apnea detection. We proposed signal denoising and data calibration techniques to process the noisy data while preserving the peaks and troughs which reflect the possible apnea events. We identified some special characteristics of sleep apnea such as signal spikes that can be captured by smartwatch, and proposed efficient and effective methods to extract proper features to train machine learning models for apnea detection. Through extensive experimental evaluations, we demonstrated that apnea events detected by our system is highly correlated with the ground truth. More specifically, ApneaDetector can detect apnea events with high precision (0.9674), recall (0.9625), and F1-score (0.9649). The AHI values estimated by ApneaDetector and the PSG ground truth are highly correlated. Although our system is designed and implemented on commercially available smartwatch, it can also be ported to other wrist-worn devices (e.g., Fitbit), since the only sensor used in ApneaDetector (i.e., the accelerometer) is available on almost all wrist-worn devices.

REFERENCES

22 • Xianda Chen, Yifei Xiao, Yeming Tang, Julio Fernandez-Mendoza, and Guohong Cao


