An Active Sleep Monitoring Framework Using Wearables

H. M. SAJJAD HOSSAIN, SREENIVASAN R. RAMAMURTHY,
MD ABDULLAH AL HAFIZ KHAN, and NIRMALYA ROY,
University of Maryland Baltimore County

Sleep is the most important aspect of healthy and active living. The right amount of sleep at the right time helps an individual to protect his or her physical, mental, and cognitive health and maintain his or her quality of life. The most durative of the Activities of Daily Living (ADL), sleep has a major synergic influence on a person's fuctional, behavioral, and cognitive health. A deep understanding of sleep behavior and its relationship with its physiological signals, and contexts (such as eye or body movements), is necessary to design and develop a robust intelligent sleep monitoring system. In this article, we propose an intelligent algorithm to detect the microscopic states of sleep that fundamentally constitute the components of good and bad sleeping behaviors and thus help shape the formative assessment of sleep quality. Our initial analysis includes the investigation of several classification techniques to identify and correlate the relationship of microscopic sleep states with overall sleep behavior. Subsequently, we also propose an online algorithm based on change point detection to process and classify the microscopic sleep states. We also develop a lightweight version of the proposed algorithm for real-time sleep monitoring, recognition, and assessment at scale. For a larger deployment of our proposed model across a community of individuals, we propose an active-learning-based methodology to reduce the effort of ground-truth data collection and labeling. Finally, we evaluate the performance of our proposed algorithms on real data traces and demonstrate the efficacy of our models for detecting and assessing the fine-grained sleep states beyond an individual.

CCS Concepts: • **Information systems** → **Information systems applications**; **Mobile information processing systems**; • **Computing methodologies** → **Machine learning**; **Learning settings**; **Active learning settings**; • **Computer systems organization** → **Embedded and cyber-physical systems**; • **Hardware** → *Communication hardware, interfaces and storage*; *Sensor applications and deployments*;

Additional Key Words and Phrases: Sleep monitoring, active learning, crowdsourcing, gradient classifier, wearable technology

ACM Reference format:
H. M. Sajjad Hossain, Sreenivasan R. Ramamurthy, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. 2018. An Active Sleep Monitoring Framework Using Wearables. *ACM Trans. Interact. Intell. Syst.* 8, 3, Article 22 (July 2018), 30 pages.
https://doi.org/10.1145/3185516

The reviewing of this article was managed by associate editor H. M. Sajjad Hossain.
This research was supported in part by the NSF under Grants CNS-1344990, CNS-1544687, and IIP-1559752; the ONR under Grant N00014-15-1-2229; Constellation: Energy to Educate; the UMB-UMBC Research and Innovation Partnership Grant; and the Alzheimer's Association Research Grant AARG-17-533039.
Author's addresses: H. M. S. Hossain, S. R. Ramamurthy, and MD A. Al Hafiz Khan; emails: {riaj.sajjad, sreeni1, mdkhan1}@umbc.edu; N. Roy (corresponding author), Department of Information Systems, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD, 21250, USA; email: nroy@umbc.edu.

https://doi.org/10.1145/3185516

1 INTRODUCTION

Sleep monitoring has been in the limelight of research due to the growing need for good-quality sleep in a person's day-to-day life. Moreover, sleep has a bi-directional relationship with the well-being of a person (Saeb et al. 2017). Sleep disorders such as sleep apnea, chronic obstructive pulmonary disease (COPD), chronic renal diseases, and various other medical conditions can manifest as disruptions in sleep patterns and thus affect sleep quality (Parish 2009). Kline (2013) has defined sleep quality as one's satisfaction of the sleep experience, integrating aspects of sleep initiation, sleep maintenance, sleep quantity, and refreshment on awakening. Sleep quality is highly correlated to exhaustion, discomfort, depression, and lack of concentration during the day. The quality of sleep reveals intuitive symptoms caused by the underlying diseases. Clinical studies have suggested that if the sleep quality is improved, the underlying symptoms of a patient might improve, too. Monitoring sleep could help a physician diagnose the underlying condition. Clinically, polysomnography (PSG) is used for sleep monitoring, which is also the "gold standard" for sleep monitoring and diagnosis of sleep-related disorders. PSG captures multiple physiological parameters such as electroencephalogram (EEG), electrocardiogram (ECG), electromyogram (EMG), and electrooculogram (EOG) simultaneously, which makes it the best solution to diagnose sleep disorders. PSG provides general sleep measures, such as total sleep time (TST) and sleep efficiency (SE), and also detects specific sleep stages. Conducting PSG requires a specialized environment like a sleep center or clinic, which makes it un-suitable for day-to-day sleep monitoring. Recent advances in the field of sensor technology and wearables have led to the advent of various sleep monitoring wearable devices such as Fitbit (2007), Actiwatch (Actigraph 2004), the BASIS watch (Basis Band B1 2014), Misfit Shine, Withings Pulse O2, and so on. These devices are commercially available in the market except for the BASIS watch (Basis Band B1 2014). These devices record accelerometer data and heart rate to monitor sleep quality. Research indicates that Actiwatch and Fitbit captures the TST and the SE well. In other words, it focuses on calculating the duration of sleep, which gives a very good insight on someone's sleep hygiene. However, the BASIS B1 band (Basis Band B1 2014) classifies sleep stages (REM, light sleep, deep sleep) and provides the means to identify patterns and triggers that are causing sleep disturbances (Mantua et al. 2016). The use of wearable devices has also been proved effective for in-home sleep measurements and evaluations (Kuo et al. 2017).

To evaluate a person's sleep quality, it is important to detect the stages and micro stages of sleep. Researchers view this problem as a classification problem and have been using machine-learning algorithms on the data extracted from the wearable devices. Existing sleep monitoring studies have been using supervised learning algorithms where they collect and label a set of training data with pre-defined classes that the system aims to detect. Larger labeled training dataset with consistent label information can help to build a more generalized classifier (Liao and Zhu 2014; Sordo and Zeng 2005). One of the major challenges faced by researchers while addressing this problem is the collection of the ground-truth information. Collecting ground truth without violating privacy (using cameras) of the individuals is an extremely difficult task. It is also a herculean task for the test subjects and other human annotators to annotate the data manually. Leveraging unsupervised algorithm can help to eliminate the requirement for labeled data. However, drawing boundaries between similar instances (with respect to properties) but belonging to different classes are difficult using unsupervised learning. The microscopic sleep states are very hard to differentiate based on the accelerometer data as the signal patterns are quite similar for different states. For example, subtle movements during sleeping and not sleeping while lying in the bed exhibits similar signature pattern. Feedbacks from the users can help us acquire proper labels of data instances for similar circumstances and help us to differentiate between them. By employing active learning (AL) (Settles 2012), we can acquire feedbacks from the users for important data instances. AL can

not only mitigate the manual effort needed for collecting ground-truth information but also reduce the training time. AL retrieves the most informative data instances from a pool of unlabeled data instances and poses them as queries to the annotators. As a consequence, we only have to label a handful number of instances. Several researches showed the effectiveness of AL in activity recognition domain (Alemdar et al. 2011; Bagaveyev and Cook 2014; Hossain et al. 2017). Using AL, we can improve our model incrementally based on the feedback provided by the annotators. AL works in an online manner, where we receive a stream of instances and then form a pool from which we select the single most informative instance. To get bulk label information, crowdsourcing (Chittilappilly et al. 2016) has been exploited in different problem domains. It has been used in activity recognition domain as well (Chang et al. 2017; Hwang and Lee 2012). We can collect ground-truth information of unlabeled samples in a bulk that will help to improve our model. In addition, identifying and classifying only pre-defined sleep stages is not sufficient for medical diagnosis. Even though the current state-of-the-art wearable technology cannot replace the PSG in clinical evaluations, obtaining accurate sleep stages has been the aim of every researcher who works on this problem domain. Further, we have identified certain weaknesses in the literature, such as patients suffering from nightmare disorder, muscle contractions have not been considered in any study. In our previous work on Sleep Well (Hossain et al. 2015), we proposed a sleep monitoring model using an accelerometer-mounted wearable device to classify previously unseen various sleep states and that further improves patients' sleep hygiene by being able to pinpoint the causes of sleep disturbances. We trained our model using supervised and unsupervised learning algorithms and identify the basic sleep states: Rapid Eye Movement (REM), non-REM (NREM) sleep, awake, movement, getting up from bed, getting up, and sitting. In this article, we extend our work and introduce a crowdsourcing model to collect large volume of labels and also introduce a sleep scoring module.

Wearable devices have become common and accessible to people these days and researchers are making them evolve by incorporating various sensors as attested by the new release of smart watches such as Google Android Wear (Android Wear 2014), and so on. The world is moving more towards wearable technology that has expedited a plethora of application domains ranging from health care applications (Jonas et al. 2014) to sports (Daiber and Kosmalla 2017). In this study, we have used two different wearable devices EZ430-Chronos (2013) and wActiSleep BT (Actigraph 2004) worn on the waist. After collecting the data, we apply a variant of gradient descent algorithm to build a classification model. Further, we apply importance-weighted active learning to label the uncertain data points and also incorporate previously unseen sleep states. Active learning improves the annotation effort greatly and improves the performance of classification model. We also exploit crowdsourcing to collect bulk ground-truth information offline. To discover abrupt changes on the data streams, increase the classification accuracy, remove noises, and provide greater support for informativeness in active learning, we propose an online change point detection algorithm. Finally, we show the results of our proposed algorithm using a publicly available benchmark dataset (Borazio et al. 2014) that provides the sleep phases determined by clinical polysomnography where the data were collected using a wrist-worn device. The main contributions of the article are summarized below:

- We investigate several classification approaches and propose a gradient descent classification model for recognizing the underlying microscopic context states associated with the sleep disorders.
- We introduce an online change point detection–based classification approach to detect any abrupt changes on streaming dataset for better microscopic sleep state classification, data noise, and uncertainty reduction.

- We develop an active-learning-based sleep monitoring model that extensively reduces the data annotation effort and ground-truth data collection from the user personal space and helps scale the model among a community of individuals.
- We demonstrate a crowdsourcing model using visual illustration and rank the workers based on their feedback.
- We evaluate our model based on two real-world datasets, one with polysomnography result along with the wrist-worn accelerometer sensor values from 42 subjects (Borazio et al. 2014), other with general labeled data collected from 17 test subjects.
- We evaluate the quality of sleep using the Pittsburgh Sleep Quality Index (PSQI) (Buysse et al. 1989) and the Webster scale (Webster et al. 1982).

2 RELATED WORKS

In medical studies, PSG is the major sleep study tool to diagnose a patient's sleep quality (Chesson et al. 1997). Polysomnography records biophysiological changes that occur during the sleep. Apart from PSG, some other sleep study tools are the Multiple Sleep Latency Test (MSLT) and Maintenance of Wakefulness Test (MWT) (Johns 2000). These diagnoses are cumbersome and need a lot of prior setup; for example, PSG requires 12 channels requiring almost 22 wire attachments to a patient. Obviously, this imposes a great level of discomfort to patients and researchers. Early works (Oakley 1997) involving wearable devices to replicate the polysomnography results validated the applicability of actigraphy in sleep monitoring. Sadeh (2011) provided further justification for the use of actigraphy in sleep research. Van et al. proposed a model to support the feasibility of continuous home monitoring of sleeping trends using wearable devices (Van Laerhoven et al. 2008). Recently, the authors of Kuo et al. (2017) proposed a wearable actigraphy device with a low sampling rate for in-home sleep assessment. Several other works (Matsui et al. 2017; Purta et al. 2016; Sathyanarayana et al. 2016, 2017; Sun et al. 2017) also ascertain the strength and simplicity of wearable devices in sleep monitoring. The authors of Rofouei et al. (2011) developed a wearable neck cuff system for monitoring physiological signals in real time. The authors of Nguyen et al. (2016) have developed a lightweight and inexpensive in-ear wearable sensing system that can capture electrical activities of the brain and eye and facial muscles. Nguyen et al. (2016) have used a supervised non negative matrix factorization algorithm to adaptively analyze the signals. A sleep monitoring model using image analysis has been proposed in Nakajima et al. (2000), but it has proved inefficient in case of low light conditions at night. Liao and Yang (2008) used near-infrared cameras to overcome this challenge, but the images still created non-uniformity. A novel sleep monitoring framework (LullaBy) to capture and monitor the sleeping environment using a microphone, light sensor, and motion sensor has been proposed in Kay et al. (2012). Yanzhi et al. (Ren et al. 2015) put emphasis on the importance of breathing pattern while sleeping and the proposed model captures the breathing sound using signal envelop detection on the acoustic data. The proposed model can detect snore, cough, turn over, and get up using the acoustic features. The authors of Zhang et al. (2013) proposed a real-time system to monitor the sleep conditions where pulse oximeter is exploited to monitor user's pulse oxygen saturation (SPO2) during the sleep process.

Pressure bed sensors have been used to supervise the postures and movements of the users in sleep (Foubert et al. 2012; Nam et al. 2016a). Though these methods are unobtrusive and do not create discomfort to the users, but still it has not been streamlined due to its cost and deployment issues. Hoque and Stankovic (2010) used fine-grained body positions from accelerometer data using WISP tags attached to the sides of a bed. A novel framework for pressure image analysis to monitor sleep postures including a set of geometrical features for sleep posture characterization and three sparse classifiers for posture recognition has been proposed in Liu et al. (2013).

The authors of Nam et al. (2016b) have proposed a sleep monitoring framework comprising of an accelerometer and a pressure sensor. Features pertaining to body motion, respiration, body activity, and heart rate were extracted and the proposed framework fuses information from various features and detects the stages of sleep. Odunmbaku et al. (2016) have proposed a combined framework for fall and sleep monitoring of elderly people by hypothesizing that the acceleration calculated from the accelerometer data will be in the range 0–1.5 m/s^2. The authors of Velicu et al. (2016) have used a custom-made accelerometer chip that streams data to an Arduino board. In addition to the accelerometer, a ECG sensor is also used. Features such as Heart Rate and RR interval were extracted, and Kushida's algorithm-derived equation was used to differentiate the sleep stages.

Sleep-related research are gaining attention due to the recent proliferation of low-cost easy-to-deploy technologies based on mobile and ambient sensors and its large penetration in the market. Commercial wearbale devices, such as Fitbit (2007), Zeo (2003), Actigraph (2004), Jawbone, Sleep Tracker, and so on, have been used extensively these days for monitoring sleep and activities of daily living (ADLs). iSleep (Hao et al. 2013) uses the built in microphone sensor of smartphone to detect the events that are closely related to sleep quality like body movements, coughing, snoring, and so on. The authors of Fahim et al. (2013) used the accelerometer sensor of the smartphone to track the sleep duration and user movement patterns. Chen et al. (2013) proposed a passive approach to track some stationary features, such as user silence, ambient light, phone usage and charging, and so on, for monitoring sleep habits, and developed a mobile application BeWell (Lane et al. 2011) for unified health monitoring. Bai et al. (2012) used the daily context information of a user to define sleep quality. Sleep Hunter (Gu et al. 2014) used the accelerometer and microphone sensors of the smartphone, a fine-grained detection of sleep stage transition for sleep quality monitoring, and an intelligent wake-up call. Mimo Baby Monitor is a bodysuit for infants aged 0–12 and incorporates a respiratory sensor, an accelerometer sensor, and temperature sensor to measure the physiological signals, body movements, and temperature, respectively. These signals are transmitted via bluetooth to an online data cloud and to the caretaker's mobile (Mimo 2016). Using several android apps like *Sleep As Android*, *Sleep Time Smart Alarm Clock* (Sleep Time Smart Alarm Clock 2015), *Sleep cycle*, *SleepBot*, and so on, it is possible to monitor the quality of sleep. Other commercial unobtrusive technologies like Beddit (2015) and Hello Sense (2014) can also monitor sleep. Hello Sense also tracks the quality of the sleeping environment. Kaplan A (2001) proposed to use change-point segmentation on PSG data to differentiate the macrostructural organization of sleep. A point process–based novel model for the assessment of heart rate variability and respiratory sinus arrhythmia based on PSG data has been proposed in Citi L (2011).

Active learning has been investigated in the activity recognition domain in several works (Alemdar et al. 2011; Bagaveyev and Cook 2014; Hossain et al. 2017). Enamul et al. (Hoque and Stankovic 2012) proposed an activity recognition model *AALO* in single inhabitant smart home context using active learning. The authors of Hossain et al. (2017) applied active learning by expected error reduction analysis in a smart home environment for classifying activities that include sleep. In Sahami Shirazi et al. (2013), the authors propose a model for crowdsourcing sleep data. Yung-Ju et al. proposed a mobile crowdsourcing model to annotate travel activities in real-world settings (Chang et al. 2017). Hwang and Lee (2012) proposed a crowdsourcing framework that models the combination of scene, event, and phone context to map the unseen audio data with activities.

In this article, we take a different approach and look into the fundamental problem of scaling the sleep monitoring models beyond a specific individual. To realize this, first we analyze the microscopic physiological contexts and psychological clauses behind a sound or bad sleep. We investigate traditional classification algorithms to successfully detect those events and propose a

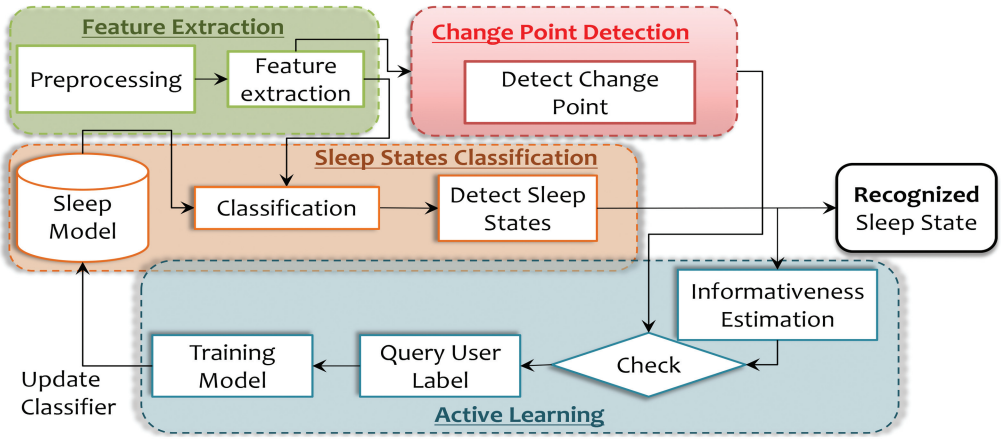

Fig. 1. An architectural overview of the Sleep Well (Hossain et al. 2015) framework.

novel online change point detection–based method for enhancing the classification accuracy and eventually help guide the design of a community scaling model using active learning.

3 OVERVIEW OF THE SLEEP WELL FRAMEWORK

Sleep is not just a dormant part of our lives; we remain very active and pass through several important stages of sleep. Interference or disturbance in these states can cause impatience, drowsiness, and lack of concentration during the regular activities of daily living. Therefore, to maintain good sleep we have to sleep a certain amount of time in each of those sleep states. There are two main types of sleep states:

- Non-Rapid Eye Movement (NREM) (also known as *quiet sleep*). NREM consists of three states (stage-1, stage-2, stage-3).
- Rapid Eye Movement (REM) (also known as *active sleep*).

A complete and healthy cycle of sleep consists of a progression from states 1 to 3 before reaching REM state, and then the cycle starts over again. If REM sleep is disrupted and the person wakes up, then the person's circadian cycle is disrupted. To complete the cycle the person will move to REM state directly next time. Thus, it is very important to sleep a good amount of time each day and maintain a good sleep cycle. REM sleep is considered active sleep, because in this state people dream. If a person is having a nightmare disorder, then too often it is possible that he/she is having problems completing the sleep cycle. In this article, we first focus on properly classifying the sleep cycle into these finer states. We also propose to integrate some other broader intermediate sleep states such as *movement*, *getting up from bed*, and *getting up and sitting*. These other states would help to identify the casual and formal causes of sleep disturbance and sleep latency and provide meaningful insights on designing scalable sleep monitoring technologies and automated assessment methodologies.

3.1 Sleep Well Architecture

In Figure 1, we demonstrate our proposed framework. Our proposed framework consists of the following logical components.

Table 1. Features Used for Sleep Micro-States Classification

	Name	Definition
Time Domain Features	Mean	AVG $(\sum x_i)$, AVG $(\sum y_i)$, AVG $(\sum z_i)$
	Mean-magnitude	AVG $\sqrt{x_i^2 + y_i^2 + z_i^2}$
	Magnitude-Mean	$\sqrt{\bar{x}^2 + \bar{y}^2 + \bar{z}^2}$
	Variance	VAR $(\sum x_i)$, VAR $(\sum y_i)$, VAR $(\sum z_i)$
	Co-Variance (Two-axis correlation)	$cov(xy)$; $cov(yz)$; $cov(xz)$
	Standard Deviation	$\sigma_x = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$; $\sigma_x = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$; $\sigma_x = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$
Frequency Domain Features	FFT-Magnitude	$m_j^{(x)} = \lvert a_j + b_j i \rvert$; $m_j^{(y)} = \lvert a_j + b_j i \rvert$; $m_j^{(z)} = \lvert a_j + b_j i \rvert$
	FFT-Energy	$\frac{\sum_{j=1}^{N}(m_j^2)}{N}$ for x, y, z respectively

- **Feature Extraction:** After collecting raw sensor data, this component preprocesses and extracts the low level signal features as shown in Table 1 from the processed raw sensor data (details in Section 4.1).
- **Change Point Detection:** After extracting features and analyzing the sleep data, we noticed that a change point occurs in sleep transitions (transition from one stage to another; for example, unconscious movement during sleeping, waking up, being restless in bed, etc.). The importance of these change points has proven to be very effective, as it helps in removing noise in the data and to detect the exact point of the sleep transition. For example, when a person gets up from the bed and starts walking, the accelerometer readings other than sleep classes become noisy. Therefore, by identifying change points, we can partition the data and have more fine-grained information for easing the training effort (details in Section 4.2).
- **Classification:** At this stage we train our model using the features from processed raw sensor data and build up our classification model to recognize the several intermediate sleep states. We investigated an online gradient descent (Karampatziakis and Langford 2011) as our classification algorithm. This is different from traditional gradient descent by dealing with importance weights to collaborate with active learning and learning reductions. To calculate the average loss during the classification process we propose to use squared loss function (details in Section 4.3).
- **Active Learning:** After feeding the test data into our classification model and getting the prediction, active learning helps to calculate the informativeness of each data point. If any data point falls within an uncertain space and while predicting it is found to be the most informative, then if the actual label of the point is provided it would have a more significant impact on the classification model. This component then initiates a prompt for "*query user label*" and gets the ground truth from the user. Subsequently, the labels are then used for re-training and updating the model. This component helps to ensure better classification accuracy with minimal user feedback. This also helps us to scale the sleep monitoring model across multiple individuals. The input from the change point detection method strengthens the active-learning query selection by asking the user to label the appropriate sleep state transitioning step (details in Section 4.4).
- **Crowdsourcing:** For large-scale deployment, we apply crowdsourcing to reduce the ground-truth collection effort. The problem with the existing crowdsourcing platforms is

that there is no standard to evaluate the quality of the workers. In our model, we calculate two parameters for each worker—*reliability* and *awareness*. Using these two parameters, we rank the user and get the most out of our crowdsourcing platform (details in Section 4.5).

4 SLEEP WELL FRAMEWORK DESIGN

In this section, we describe in detail the design of our Sleep Well framework. We first discuss several micro-states of sleep and feature extraction process. Next we discuss an online change point detection algorithm to have a better handle on the microscopic sleep state classification problem.

4.1 Sleep Event Detection and Feature Extraction

We extract low-level features using each of the three components of the triaxial accelerometer signal to capture the aspects of movements while sleeping. We use both time and frequency domain features in our framework. As the user is not physically active while sleeping, very few movements are involved, so we choose a lower sampling frequency. We extract features from data using windows of 60 samples, corresponding to 1s of accelerometer data. From each window, we calculate the features mentioned in Table 1. Time domain features help to differentiate between dynamic and static movements. The frequency domain features help to identify patterns within acceleration data, which aids in discriminating discrete movements and their intensities.

4.1.1 Feature Selection. We scale and make our model computationally effective by discarding unnecessary features. We select the subset of features, best fit for our model by applying the Restricted Forward feature Selection (RFS) algorithm (John et al. 1994). It was performed in two steps. First, we applied the Forward feature Selection (FS) algorithm, which ranks the features in decreasing order of their accuracy. The FS algorithm iterates through the feature space and measures the Leave-One-Out-Cross-Validation (LOOCV) error for each component in the feature space $\{f_1, f_2, f_3, \ldots, f_N\}$. In case of traditional FS, after the first iteration, FS calculates the best individual feature f_i. In the next iterations, FS finds the best subset consisting of two components, f_i and one other from remaining $N - 1$ features. In the following iterations, FS ranks more features and evaluates the subset accordingly, so that after N iterations, the winner is the overall best feature set in these N iterations. In the second step, we invoke the RFS to restrict the number of features to rank at each iteration. After the first iteration, we consider only the first $N/2$ ranked features for the following iteration. After adding another feature to the winner of the first iteration at the second iteration, we consider the first $N/3$ components of the remaining ranks. RFS repeats this process until it finds the best m feature sets. The difference between conventional FS and RFS is that RFS considers only a part of the remaining ranked features, whereas FS considers all the features. Of eight features, the feature selection algorithm chose four features (FFT-Magnitude, FFT-Energy, Mean-Magnitude, and Co-Variance) that help to attain classification accuracy closer to using all eight features.

4.2 Change Point Detection

Change point detection helps find the abrupt variations in the sleep data stream. While some change points provide meaningful insights and some not, our motivation in this work is to find the sleep transitions by calculating the change points (abrupt signal changes) and distinguish between the important and unimportant changes. This is not only helpful to detect the sleep-related events appropriately but also help remove noisy data points from the dataset. We develop a Bayesian online change point detection– (Adams and MacKay 2007) based algorithm for finer sleep-related event identification and online data noise reduction.

We first partition the entire sleep dataset in different regions based on a run length (Adams and MacKay 2007). Let, $x_{1:N} = \{x_1, x_2, x_3, \ldots, x_N\}^T$ denote the N data points observed over time T, which is divided into non-overlapping partitions. Consider if we find K change points and then let the dataset of partitioned data be $\{\rho_1, \rho_2, \rho_3, \ldots, \rho_k\}$ at time indices $\{t_1, t_2, t_3, \ldots, t_k\}$, where by definition $t_0 = 0$ and $t_{k+1} = N$. The discrete probability distribution over a time interval t_i to t_j is denoted by $g\ (t_i - t_j)$. Each partition ρ_t denotes a segment of the data at time t. The length of the each partition or time since the last change point occurred is defined as "run length," r. The run length goes back to 0 if change point occurs; otherwise, it increases by 1 as follows:

$$r_n = \begin{cases} 0, & \text{if changepoint occurs at } (n-1) \\ r_{n-1} + 1, & \text{otherwise} \end{cases}.$$

The conditional probability that a change point occurs on time t_k after the last change point at time t_{k-1} is

$$P(t_k|t_{k-1}) = g(t_k - t_{k-1}), \text{ where } 0 < k - 1 < n. \tag{1}$$

We assume that the predictive distribution of a change point at any time instant t only depends on the recent data. So the change points are assumed to follow Markov process. Thus the prior probability of a change point at a time instant t_k is dependent on the probability distribution of the observed data over the time interval and the preceding change point,

$$P(t_k) = \sum_{i=0}^{k-1} g(t_k - t_i) P(t_{k-1}). \tag{2}$$

The change point detection algorithm finds the number of change points and their position by calculating the posterior probability $P(r_n|x_{1:n})$ and integrating it with the predictive distribution $P(x_{n+1}|x_n)$. We do this by calculating the joint distribution of the current run length and observed data $P(r_n, x_{1:n})$,

$$\begin{aligned} P(r_n, x_{1:n}) &= \sum_{r_{n-1}} P(r_n, r_{n-1}, x_{1:n}) \\ &= \sum_{r_{n-1}} P(r_n, x_n|r_{n-1}, x_{1:n-1}) P(r_{n-1}, x_{1:n-1}) \\ &= \sum_{r_{n-1}} P(r_n|r_{n-1}) P(x_n|r_{n-1}, x_{n-r:n}) P(r_{n-1}, x_{1:n-1}), \end{aligned} \tag{3}$$

where $P(r_n|r_{n-1})$ is the transition probability and $P(x_n|r_{n-1}, x_{n-r:n})$ is the data segment likelihood probability. We calculate the transition probability using equation

$$P(r_n|r_{n-1}) = \begin{cases} h(r_{n-1} + 1), & \text{if } r_n = 0 \\ 1 - h(r_{n-1} + 1), & \text{if } r_n = r_{n-1} + 1 \end{cases},$$

where $h(x) = g(x)/\sum_{i=x}^{\infty} g(i)$. We calculate the posterior probability using Bayes's rule:

$$P(r_n|x_{1:n}) = \frac{P(r_n, x_{1:n})}{\sum_{i=0}^{n-1} P(r_i, x_{1:i})}. \tag{4}$$

We calculate the posterior probability of the run length at the time index that corresponds to a new data sample. The pseudo code of this procedure is summarized in Algorithm 1.

ALGORITHM 1: Change Point Detection

Initialize: $P(r_0) = 1$
for Each new data point x_n **do**
  Calculate the data segment likelihood probability, $P(x_n|r_{n-1}, x_{n-r:n})$
  Calculate the transition probability, $P(r_n|r_{n-1})$
  Calculate the joint distribution, $P(r_n, x_{1:n})$
  Find the posterior distribution on current run length, $P(r_n|x_{1:n})$
  Calculate the predictive distribution of x_n based on previous observation. $P(x_n|x_{n-1})$
end for

4.3 Classification

We classify the sleep states using an online gradient descent method that leverages the importance weight on streaming data samples. To build up our classification model accurately, we consider other sleep contexts such as body movements, but most of the sample data points resemble stationary states during sleep. Online gradient descent with importance weight aware updates (Karampatziakis and Langford 2011) helps to overcome this limitation of data by assigning weight to classes with lesser data points. The key principle here is as follows: *The assignment of importance weight h to a sample that make it appears like a regular example of h times in the dataset.* We assume C is our classification model and use a squared loss function to examine the consistency of C. The goal of our classification model is to minimize the loss function that reflects better accuracy. After each iteration of gradient descent, C is not altered; rather, it is improved by adding an estimator h to optimize the loss function. We assume y is the true label and p is prediction of our model, where $l(p, y)$ is the loss function as shown in Equation (5). At each step C is updated using Equation (6),

$$l(p, y) = \frac{1}{2}(y - p)^2, \tag{5}$$

$$C_{m+1} = C_m + h(x). \tag{6}$$

Let w be the vector of weights, and the training set is a set of (x_i, y_i, h_i), $i = 1, \ldots, T$, where x_t is a vector of d features. For linearity, we assume $p = w^T x$. Our goal is to assign w in such a way that the model C converges to the optimized solution. Assigning weight to a data point (x, y), h times in a row have a cumulative effect with scaling factor $k(h)$ as shown in Equation (7). This scaling factor is defined by Equation (8), where η is the learning rate (Karampatziakis and Langford 2011). At each iteration, this weight is updated accordingly to the loss function l. Our base classifier C is a multinomial logistic regression model. Our proposed classification algorithm for finer non-stationary sleep states detection is shown in Algorithm 2. We first initialize the importance weights for each of the data instances and then train our classifier C. The weights of the data instances get adapted based on the prediction made by our trained model,

$$w_{i+1} = w_i - k(h)x, \tag{7}$$

$$k(h) = \frac{p - y}{x^T x}(1 - e^{-h\eta x^T x}). \tag{8}$$

4.4 Active Learning–Based Community Scaling

Our goal in this article is to scale the sleep monitoring model to a community of individuals. While significant research has been done on sleep monitoring and assessment and intervention strategies, lack of novel scaling algorithms prohibits the deployment, large-scale validation, and acceptance

ALGORITHM 2: Importance Weighted Sleep Classification

Input: Extracted feature vectors from raw data with their respective labels.
Output: Trained model and updated importance weight of the data points.
Initialize: $\forall_i w_i \leftarrow 0$
Get the feature vector for data point x_i
while true **do**
  Train classifier C
  Calculate the scaling factor k(h)
  for $i = 0$ to N **do**
    Calculate the weight w_i for each x_i
    Update: $w_i \leftarrow w_i - \frac{p_i - y_i}{x_i^T x_i}(1 - e^{-h\eta x_i^T x_i})x_i$
  end for
  if $l(p, y)$ converges **then**
    break
  end if
end while

of these technologies for healthy lifestyle, smart health, and independent living applications. In this section, we investigate how active-learning-based machine-learning algorithms help build an informative model in presence of minimally labeled datasets. We also depict how change point detection–based time-series data analytics methodology helps reduce the data uncertainty and guides the selection of the most informative query.

Active learning has been proved to be very effective when combined with supervised learning when a large pool of unlabeled data is available. Though traditional passive learning takes the initiative to label the unlabeled data randomly, most of the data points that are selected randomly do not ensure better classification. It is difficult to collect all of the sleep-related ground-truth information from the user though by using the accelerometer sensor it is possible to broadly monitor the user sleep behavior and the specific sleep duration. To collect more fine-grained details about sleep, we train our proposed gradient-based classifier with the causes of sleep disruption (such as waking up from nightmares, muscle cramp, etc). By applying active learning, we propose to collect the labels of these informative data points so that our model can better classify the sleep stages and conditions and help scale this model in the presence of a minimal amount of ground truth. While applying active learning, one constraint is that we have to assure that the whole labeling process does not become too intrusive. Crowdsourcing can help us overcome this constraint by collecting a large amount of labeled data via arbitrary participants and providing aid in community scaling.

4.4.1 Query Selection. In the following, we briefly discuss the query selection approaches for active learning:

- **Query Synthesis:** The active learner asks the human annotator for "label membership" by using membership queries. In this approach, the learner generates instances rather than samples from an existing unlabeled set. But the problem with this approach is that the human annotator may have difficulty interpreting and labeling arbitrary instances.
- **Stream-based selective sampling:** Each unlabeled instance is drawn at a time from the input source, and the learner may decide instantly whether to query the instance or not. As we are using online classification algorithm and the data are processed in stream, we use this sampling strategy for our active learner.

- **Pool-based sampling:** Evaluates and ranks the entire collection of unlabeled data before selecting the best query from a pool of instances.

4.4.2 Sampling Metrics. Different sampling metrics such as least confident, margin sampling, or maximum entropy–based sampling are common in active-learning algorithms. We propose to use the importance-weighted active-learning approach to build our community-scaled sleep monitoring model (Beygelzimer et al. 2009). To decide which points are most informative, we first calculate the utility measurements of unlabeled data points. Whether a data point x_t will be queried or not depends on the history of labels seen so far based on our change point detection, gradient-based classification, and the identity of the point. If a change point is detected at data point x_t at time index t_n, and the label of x_t is inconsistent with the label of current run r_n, then we invoke active learning. A probability measure p_t is maintained for each data point x_t. A coin flip, $Q_t \epsilon \{0, 1\}$ with $E[Q_t] = p_t$, determines whether the data point will be queried or not. If the data point is queried based on the past history, then we update importance weight by $\frac{1}{p_t}$.

The active-learning algorithm maintains an effective hypothesis space H_t throughout the process. Initially, H_t contains all of the hypotheses from global space H. The expected loss of a hypothesis, $h \epsilon H$ at time T, is defined by Equation (9),

$$L_T(h) = \frac{1}{T} \sum_{t=1}^{T} \frac{Q_t}{p_t} l(h(x_t), y_t). \tag{9}$$

As it progresses, H_t becomes narrower by taking a subset, and ensuring that the factual loss of H_{t+1} is not much worse than the smallest loss, L_t^* in H_t,

$$H_{t+1} = \{h \epsilon H_t : L_{t+1}(h) \leq L_t^*(h)\}. \tag{10}$$

For each data point x_t, the active-learning algorithm looks at the range of predictions and their losses by hypotheses in H_t and sets the sampling probability to the size of this range,

$$p_t = \max_{f,g \epsilon H_t} \max_{y} l(f(x_t), y) - l(g(x_t), y). \tag{11}$$

If the range is too high above the rejection threshold, then the hypotheses disagree greatly with each other. This certifies that the current prediction of x_t lies in the uncertain region. The active-learning algorithm then queries for the label to settle the uncertainty. Our proposed active-learning algorithm for largely reducing the micro-sleep states annotation effort is shown in Algorithm 3.

Apart from using only predefined class labels, the user can introduce a new unseen class along with indicative attributes with the help of active learning. While prompting for label of data point x_t, we also collect the reason for their choice of label in restricted number of words. We find specific attributes from the provided reason and associate that attribute with the data point x_t. For example, if a user labels a data point as "getting up & sitting" and specifies the reason as "woke up from nightmare," then the Sleep Well framework extracts the attribute "Nightmare" from the provided reason. Subsequently, we re-evaluate our classification model and apply a recursive classification to associate the provided attributes to similar data points. This will help our model achieve microscopic sleep state classification and finer evaluation for more elaborative and accurate diagnosis of patients and eventually scale the model beyond an individual premises.

4.5 Crowdsourcing

Crowdsourcing has been proven to be an effective component for collecting labels in many machine-learning applications. Large-scale data processing and annotating the data with multiple annotators or experts alleviate the traditional process for gathering ground-truth data that are lengthy, costly, and time-consuming. However, by using crowdsourcing, we accumulate a large

ALGORITHM 3: Active Learning with Importance Weighted Sampling

Input: L = set of labeled instances $\{(x, y)^l\}_{l=1}^{L}$
U = set of unlabeled instances $\{(x)^u\}_{u=1}^{U}$
A classifier model, C_θ
Output: Updated classifier model, C_θ.
Updated importance weight of queried data points.
for every instance in U **do**
  set p_t of instance x_t using equation (11)
  $y_t \longleftarrow$ Prediction of C_θ for x_t
  $queried \longleftarrow False$
  * *Check if x_t is a change point or not* *\
  if x_t falls in between successive run r_{n-1} and r_n using the posterior probability $P(r_n|x_{1:n})$ **then**
    if y_t is not same as the label of current run r_n **then**
      query label y_t.
      $L_t \longleftarrow L_{t-1} \cup \{x_t, y_t, \frac{1}{p_t}\}$
      $queried \longleftarrow True$
    end if
  end if

  if p_t is greater than rejection threshold and $queried = False$ **then**
    query label y_t.
    $L_t \longleftarrow L_{t-1} \cup \{x_t, y_t, \frac{1}{p_t}\}$
  else
    $L_t \longleftarrow L_{t-1}$
  end if
  Update the hypothesis space H_t
end for
C_θ = Best hypothesis from H_t
return C_θ

volume of labeled data, but we also increase the risk of introducing a lot of noisy and ambiguous labels into our classifier. So it is necessary to identify potential reliable annotators and limit the effect of introducing noisy labels. However, a major challenge in crowdsourcing is to verify the provided labels. To tackle this, we propose to calculate the inter-annotator agreement using Fleiss' Kappa statistics and identify the proper label for the data point of concern. In our model, we rank the annotators based on their reliability and awareness of the feedback. Here reliability refers to the correctness of the feedback, and awareness indicates the willingness of the annotators. For each annotator, we maintain a probability measurement, δ_{tk}^{i}:

$$\delta_{tk}^{j} = P(y^j = k | y^j = t). \tag{12}$$

In Equation (12), δ_{tk}^{j} denotes the probability that the annotator j provides the class label k to an instance given that the true class label is t. Our goal is to learn the parameter δ_{tk}^{j} for each annotator. Suppose there are R annotators. If the number of data instances is N, then we initialize a $N \times R$ matrix M. M_{ij} denotes the label of instance i provided by annotator j. Given that y_i^t is the true label, for instance, x_i, we assume that $y_i^1, y_i^2, \ldots, y_i^R$ are independent. Let $y = \{y_1^t, y_2^t, y_3^t, \ldots, y_N^t\}$ be the set of true labels for the data instances. In our model δ is the reliability parameter. Our goal is to learn an optimal estimator $\hat{y}$ of y to minimize the imposed error by provided noisy data

points. By taking Beta prior on the reliabilities, we can formulate a maximum likelihood estimator $\hat{\delta}$ of δ using Equation (13). By taking y as a hidden variable, we can estimate $\hat{\delta}$ using expectation maximization (Raykar et al. 2010),

$$\hat{\delta} = \arg\max_{\delta} \log P(\delta|M) = \arg\max_{\delta} \log \sum_{y} P(\delta, y|M). \tag{13}$$

We calculate the awareness of a certain class c, a_j^c by taking the percentage of data of class c labeled by the jth annotator. We assign weight w_j^c to each annotator by taking the product of their respective reliability of a certain data instance i and awareness. Based on the weight for each class, we rank the annotators with respect to each class,

$$w_j^c = \delta_{tk}^j a_j^c. \tag{14}$$

To verify our estimator, we calculate the interuser agreement by using Fleiss' Kappa statistics. The kappa k is defined in Equation (15). In Equation (15), $1 - \bar{P}_e$ gives the degree of agreement that is achievable, and $\bar{P} - \bar{P}_e$ gives the degree of agreement actually achieved. If the annotators agree with each other, then $k = 1$, and if not, then $k \leq 0$,

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}. \tag{15}$$

Let n be the number of annotation per annotator, and let k be the number of classes. Then n_{ij} represent the number of annotators who assigned instance i to the j class. First calculate P_j, the proportion of all assignments that were to the class j:

$$P_j = \frac{1}{Rn} \sum_{i=1}^{R} n_{ij}, \qquad 1 = \frac{1}{n} \sum_{j=1}^{k} n_{ij}. \tag{16}$$

Now calculate P_i, which denotes how many annotator pairs are in agreement, relative to the number of all possible annotator pairs.

$$\begin{aligned} P_i &= \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1) \\ &= \frac{1}{n(n-1)} \sum_{j=1}^{k} (n_{ij}^2 - n_{ij}) \\ &= \frac{1}{n(n-1)} \left[\left(\sum_{j=1}^{k} n_{ij}^2 \right) - (n) \right]. \end{aligned} \tag{17}$$

Now we can measure $\bar{P}$, which is the mean of all P_i and $\bar{P}_e$ using P_j:

$$\begin{aligned} \bar{P} &= \frac{1}{N} \sum_{i=1}^{N} P_i \\ &= \frac{1}{Rn(n-1)} \left(\sum_{i=1}^{R} \sum_{j=1}^{k} n_{ij}^2 - Rn \right). \end{aligned} \tag{18}$$

$$\bar{P}_e = \sum_{j=1}^{k} p_j^2. \tag{19}$$

In Figure 2, the architecture of our crowdsourcing platform is shown.

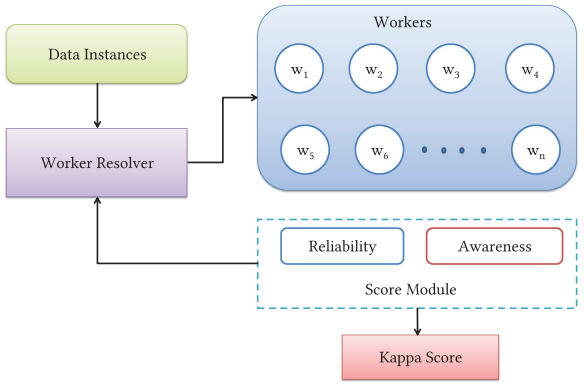

Fig. 2. An architectural overview of crowdsourcing.

5 SLEEP WELL FRAMEWORK EVALUATION

To evaluate our framework, we focus on the following specificities: (i) the performance of different classification algorithms in comparison to our classification approach, (ii) cross-user performance by building model with a user's sleep model and testing with someone else's model, (iii) performance of our framework using different wrist-band devices with accelerometer sensor, (iv) impact of active learning on our model, and (v) precision of classifier when sleep attributes are introduced in the model by active learning.

5.1 System Implementation

We have implemented and tested our model by using two separate devices, wActiSleep-BT (Actigraph 2004) and EZ430-Chronos (2013), and both of these devices contain a 3-axis accelerometer sensor. The EZ430-Chronos device also has heart monitor, pressure, and temperature sensors. We collected raw accelerometer data from both of these devices using API provided by the manufacturers. We implemented our own software to extract raw data using the C# programming language and then extracted the features using the python numpy library. We sampled the data at 60Hz frequency. For importance-weighted classification and active learning, we used the machine-learning tool Vowpal Wabbit (2016).

5.2 Ground-Truth Collection

We asked the users to log their sleep habits using a sleep diary to correctly label the data points. We asked the participants to note down their sleep routines (preferred sleeping postures, regular hours of sleep, light intensity and sleep latency) each day of the experiment. There were many challenges involved while collecting the ground truth from the sleep diary. For example, consider two different scenarios, (1) the user is *awake & lying* and (2) *awake & not lying*. In case of stationary states (when the user is not moving but he or she is either lying or just sitting in the bed), the accelerometer readings are almost identical. Also when a user gets up in the middle of the night and performs some activities (checking his or her phone, going to bathroom, etc.), there are movements involved. It was challenging to identify which movements were during sleep and which were due to some activities. The user was unable to correctly state the reason of movements in some cases. In Figures 3 and 4, we can see two different movements (awake and standing, awake and lying). The user went to bathroom at 2:03 AM and came back to bed at 2:12 AM. However, at 3:03 AM, the user was moving while lying. Therefore, to assist the ground-truth collection, we investigate a posture

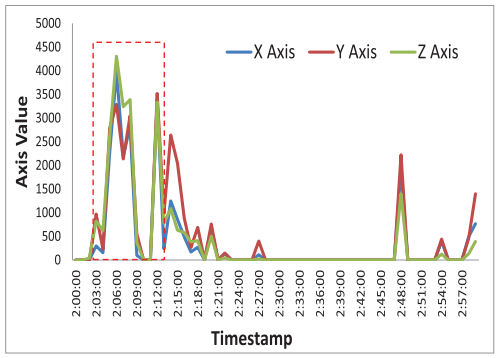

Fig. 3. Accelerometer reading when standing.

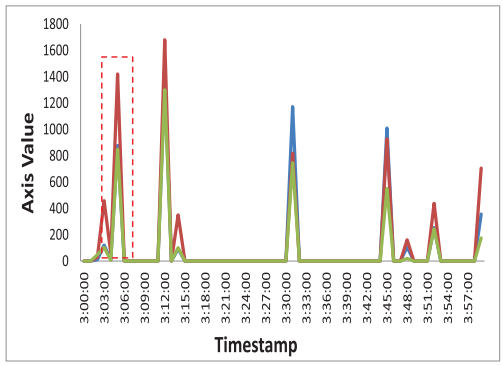

Fig. 4. Accelerometer reading when lying.

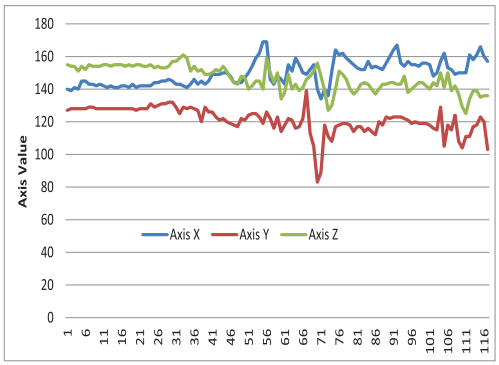

Fig. 5. Raw accelerometer data from dataset 5.3.1.

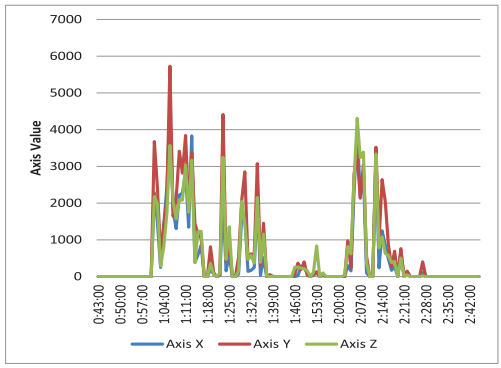

Fig. 6. Raw accelerometer data from dataset 5.3.2.

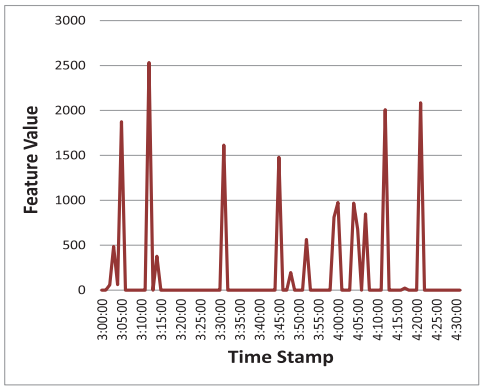

Fig. 7. Timestamped feature value.

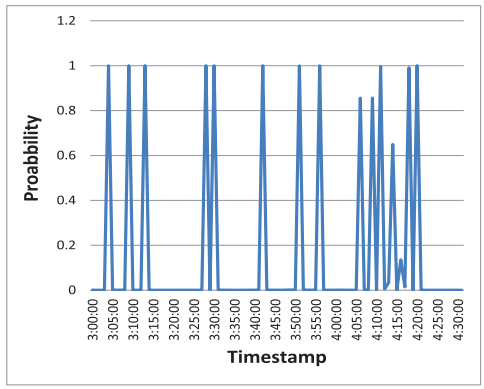

Fig. 8. Detected change points associated to Figure 7.

analysis using the inclination of the accelerometer. We observe that when perpendicular to gravity, accelerometers are more sensitive to small changes in inclination, but as the inclination increases, the accelerometer becomes less sensitive to it. To resolve this issue, we propose to use two axes. As we are using wrist-worn bands, inclination of axes y and z are used to define the posture of

$$\theta_y = \tan^{-1}\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \quad (20)$$

$$\theta_z = \tan^{-1}\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \quad (21)$$

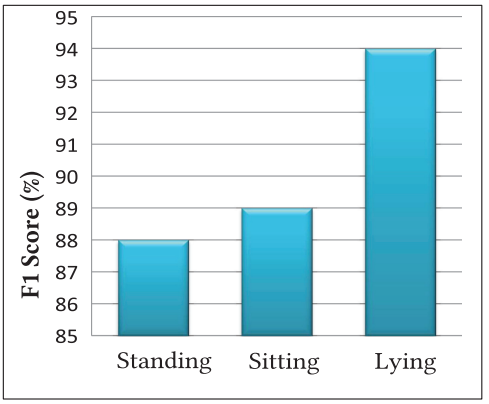

Fig. 9. Inclination measurement accuracy.

the user. The z axis measures the direction of the gravity in the horizontal position, so coupling with the inclination of x axis help infer the posture of the user. We calculate the inclination of the device by using Equations (20) and (21).

We faced a challenge to define the threshold values for these inclinations as different users have different sleeping postures. We experimented with different sleep positions (On side, Face down, On your back) and calculated the inclination of the device in those positions. We ran the J48 decision tree classifier on the postural data. Based on the results of the classifier, we defined the inclination threshold for different states, such as if $\theta_y < 16°$, then the user is standing, and if $16° < \theta_y < 61°$, then the user is considered to be sitting, and for $\theta_x > 61°$ the user is considered to be lying. Figure 9 shows the results of our posture calculation using the inclination method. We also installed couple of motion sensors in the environment to strengthen our ground-truth collection. We put two motion sensors (Aeotec Multisensor (2006)) near both sides of the bed and another one mounted near the user's body when he/she is sleeping. The sensor mounted near the body captures the motion when the user is in the bed while the other two on the sides of the bed monitor when the user is out of the bed. While extracting information from the sensor, we assumed that consecutive two data points from the sensors mounted on the sides correspond to getting in and then getting out of the bed. These multisensors also have built in light sensor, so we can detect the light condition in the sleeping environment using these sensors. Now we are able to validate the movements of the users and calculate the overall time he/she remained out of the bed efficiently by consolidating the inclination measurements, motion sensor data, and data from the sleep diary. We were able to label most of the data points correctly and remove noisy data points.

5.3 Datasets

We use real data traces collected from ≈60 users to validate the performance of our Sleep Well framework. We also compare our results for data from different body position.

5.3.1 Dataset with Clinical Ground Truth. We evaluate our model using a publicly available benchmark dataset from Technische Universität Darmstadt (Borazio et al. 2014) which provides sleep phases determined by clinical polysomnography. The dataset consists of timestamped raw acceleration data collected using wrist-worn data logger at a sampling rate of 100Hz and includes the sleep stages (movements, awake, NoREM 1-3, REM, unknown) from 42 lab patients. The trend of raw accelerometer reading in this dataset is shown in Figures 3–8. The sampling frequency is

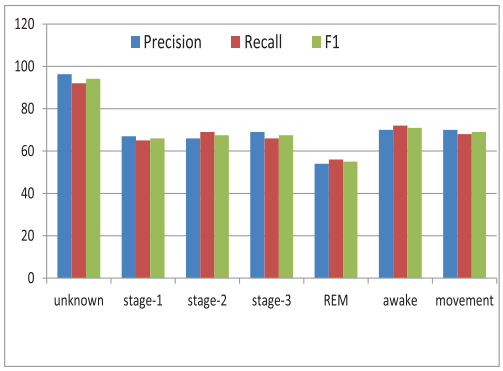

Fig. 10. Precision, recall and F1-measurement for inter user classification (dataset 5.3.1).

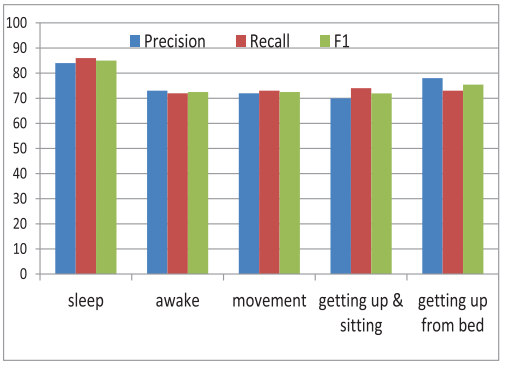

Fig. 11. Precision, recall and F1-measurement with inter user classification (dataset 5.3.2).

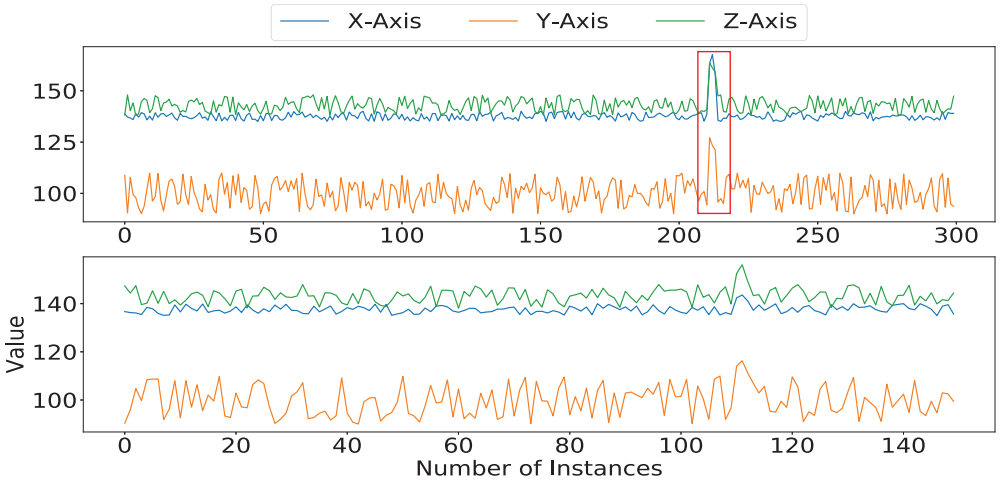

Fig. 12. The figure illustrates the effect of various sampling frequencies of the raw accelerometer data. The upper figure depicts raw accelerometer data at 100Hz frequency and the bottom figure at 25Hz frequency. The highlighted portion in the upper figure indicates a very subtle movement while sleeping.)

set to high due to the nature of the activities. Different sleep stages (NoREM 1-3 and REM) actually refer to macro activity, *sleep*. To understand the difference between movements pertaining to different sleep stages, it is necessary to capture data at higher sampling frequency. We demonstrate the effect of high and low sampling frequency in Figure 12. The spike in the highlighted box in the upper figure indicates a very subtle movement while the user is sleeping. However, when down-sampling the data to 25Hz (bottom figure) we experience no such spike and we lost the movement information. As a result it is necessary to choose a sampling frequency according to the definition and sensitivity of the classes. There are seven different classes in this dataset among which majority of the data points are labeled as unknown (51%) and awake (24%) with only a few important data points that affect the classification model. After inspecting the dataset, we note that the value of different data points of different classes were very close that imposes bias in our classification model. We handle this bias by assigning less weight to abundant data points (unknown and awake) and improve the classification process and accuracy.

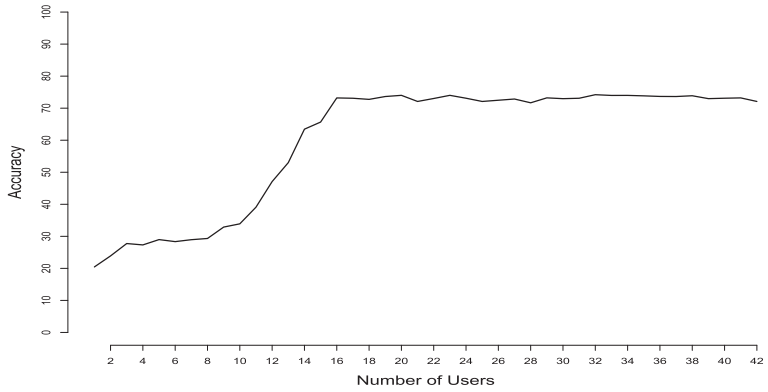

Fig. 13. The trend in intra user classification accuracy with varying population size.

5.3.2 Actigraph and Chronos Dataset. We collected sleep data using wActiSleep-BT and EZ430-Chronos at a sampling rate of 60Hz from 17 participants for two weeks. Of 17 participants (11 males and 6 females), 13 were graduate students, 3 were working professionals, and 1 was unemployed. We conducted a survey beforehand to know about their sleep routine. Using the survey, we gathered information regarding sleep time, average sleep hour, movement frequency on a scale of 1–5 and existing sleep disorders. We then selected a set of participants with diverse sleep routine and disorders. We asked the participants to put on the sensor when they go to the bed. The participants were also instructed to maintain a log the timing of getting up and getting in bed. The participants put the sensors on their waist using a belt. We noticed that wActiSleep-BT device has better sensitivity due to slight movements rather than EZ430-Chronos that help differentiate between actual movements and sleep patterns from a user. Almost 65% data points of this dataset belong to *Sleep* class and 22% to *Awake* class. As a consequence, our dataset is also imbalanced. Figure 6 shows the raw readings from ActiSleep device.

5.4 Evaluation Methodology

5.4.1 Supervised Learning. We carried out our experiments with 17 participants (11 males and 6 females) over two weeks where each participant has provided data for 8–10 days. We validated that 17 is a statistically significant population size using *t-test*. The trend in training accuracy (intra user) with respect to size of the population is shown in Figure 13. We see that after increasing the population size more than 17, the accuracy is not changing significantly. We proved our hypothesis that it did not happen by a chance by conducting a *t-test* using dataset 5.3.1. We conducted *t-test* with varying population size and received p-value of 2.021 with 95% confidence interval. Although we get a bit more accuracy if we increase the population size (74%), due to training time and resource consumption, we chose 17 as the optimum population size. Of these 17 participants, 7 wore the EZ430-Chronos device and other 10 put on wActiSleep-BT. We split each dataset into two parts, one for training and other for testing. To overcome the class imbalance problem, the importance weight play a significant role. We look at the confusion matrices of classifying two classes (Sleep, Awake) in Tables 2 and 3. It is evident that due to class imbalance, a lot of the Awake class instances are inferred as Sleep if importance weighting is not used. We applied our classification models on the datasets mentioned in Sections 5.3.1 and 5.3.2.

Intra User Classification. We tested different classification models with our proposed Online Stochastic Gradient Descent (OSGD) method: Support Vector Machine (SVM), Multilayer Perceptron

Table 2. Confusion Matrix of Sleep–Awake classifier with Importance Weighting

	Sleep	Awake
Sleep	91%	9%
Awake	14%	86%

Table 3. Confusion Matrix of Sleep–Awake Classifier without Importance Weighting

	Sleep	Awake
Sleep	80%	20%
Awake	51%	49%

Table 4. Accuracy (Dataset: 5.3.1)(%) (Intra User)

	OSGD	SVM	MP	DT	LR	LB	RF
Unknown	98.76	96.50	85.80	99.01	98.93	98.37	97.95
Stage-1	69.45	44.44	60.36	58.57	47.63	61.82	70.40
Stage-2	70.29	41.02	58.54	59.09	48.18	68.46	71.87
Stage-3	68.36	39.15	63.36	48.24	49.33	60.77	63.94
REM	58.22	37.28	49.11	41.55	38.31	41.03	59.10
Awake	74.78	68.12	72.10	70.01	64.96	66.90	72.73
Movement	72.59	69.31	62.88	70.66	65.60	63.27	71.25
Average	73.20	56.54	66.13	63.87	58.99	65.80	72.46

Table 5. Accuracy (Dataset: 5.3.2)(%) (Intra User)

	OSGD	SVM	MP	DT	LR	LB	RF
Sleep	87.79	87.98	80.8	84.01	73.32	76.63	88.52
Awake	77.9	71.35	75.66	67.91	73.56	70.21	75.69
Movement	76.25	74.87	68.32	68.41	70.27	72.11	75.14
Getting up & sitting	72.11	64.58	67.39	68.11	64.85	69.15	70.02
Getting up from bed	78.21	69.89	70.36	70.1	71.19	62.39	73.39
Average	78.45	73.73	72.50	71.70	70.63	70.09	76.54

(MP), LogitBoost (LB), Random Forest (RF), Logistic Regression (LR), and Decision Tree (DT), using different user's dataset. The accuracy of different classification model using one of the subject's dataset from each dataset is shown in Tables 4 and 5. The average accuracy of OSGD is 73.20% for a patient from dataset 5.3.1 and 78.45% for dataset 5.3.2. This attests that consideration of inclination and sensor data and using it to correct labels in dataset 5.3.2 help yield better classification results. Also the results indicate that putting the device on the waist endows better accuracy. We investigated this disparity and found that hand movements are more abrupt and arbitrary, which results in more confusing data points. Also very subtle body movements are difficult to distinguish when using a wrist-worn accelerometer.

The major accuracy improvement was noticed for inferring the micro sleep state. Although individual accuracy for classes *Stage 1*, *Stage 2*, and *REM* for the Decision Tree (DT) classifier was better in dataset 5.3.1, the average accuracy of inferring sleep states (sleep stages 1–3, REM) is 66.58% which is better than the average of DT classifier (66.32%), while for our dataset we achieved 87.79% accuracy.

Cross User Classification. It is important that a classification process will not only recognize the sleep states of an already seen user but also help generalize the classification for new users. We cross validated our approach with the inter user classification model. We trained our model using 20 patient's data from dataset 5.3.1 and tested the trained model with remaining 22 patients'

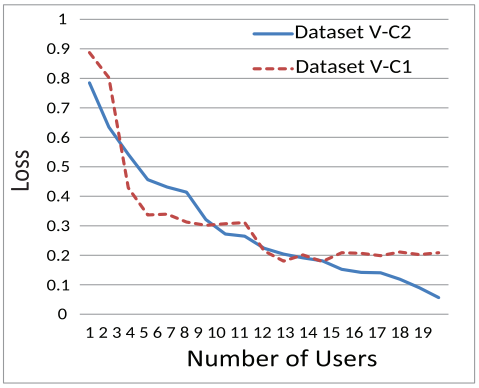

Fig. 14. Trend of loss for inter user classification in different datasets.

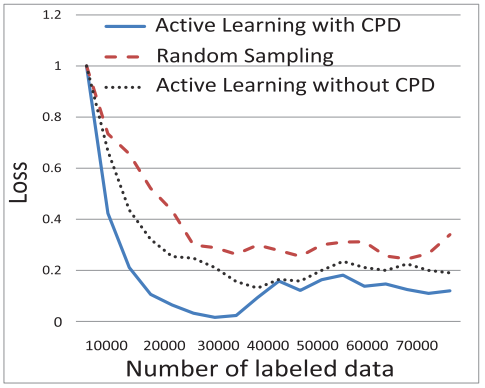

Fig. 15. Different active learning techniques for dataset 5.3.1.

data. The average accuracy was 69.79%. With data from dataset 5.3.2, we achieved 75.46% overall accuracy. Figure 10 and 11 shows the results in Precision (percentage of times that a recognition result made by the model is correct), Recall (percentage of times that a sleep state is detected), and F1-measurement (combination of both recall and precision) for both the datasets. Figure 14 also shows the trend of loss for different datasets.

5.4.2 Active Learning Experiments. In addition to supervised learning, we evaluate how we can improve the classification result using active learning with minimal user feedback. We have discussed our active-learning algorithm in Section 4.4. We sampled both the datasets with a window of 60s on accelerometer data. Each sample is a feature vector with 16 dimensions. The initial labeled dataset L_1 consisting of 135,089 samples (from dataset 5.3.1) and L_2 consisting of 42,000 samples (from dataset 5.3.2) are provided to the individual classifier C_1 and C_2 for training. Then unlabeled dataset U_1 of 510,113 (dataset 5.3.1) and U_2 of 121,147 samples (dataset 5.3.2) are used to test the classifier C_1 and C_2. The samples are provided sequentially with respect to timestamp.

The uncertain data points, meaning the points that the classifier was unable to classify, are queried in accordance with our active-learning algorithm (3). We calculated the loss at each phase after a data point is queried and the model is re-trained. We compared our result with randomly selected samples for labeling. To further assist the active-learning process, we validated the results with our change point detection (CPD algorithm discussed in Section 4.2). When a change point is detected in the dataset, we cross validated the change points with the classification result with respect to the timestamp. Figures 7 and 8 plot the association of change points with timestamped accelerometer data points. If the label of the sample is not consistent between each of the models, then we imposed active learning and queried the data point. Initially, with L_1 and L_2, we note that the average classification accuracy is 63.8% and 70%. We applied importance-weighted active learning and see that the model converged faster with change point detection; 86,719 samples (17% of total samples) from U_1 and 8,843 samples (7.3% of total samples) from U_2 were queried for the model to converge in presence of CPD that helped achieve 72% (dataset 5.3.1) and 76.89% (dataset 5.3.2) accuracy, while with randomly selected data points 68% and 73% accuracy was observed. Figures 15 and 16 shows the change in loss with random sampling, active learning with and without CPD techniques with different datasets. We see that active learning with CPD outperforms the other strategies. In dataset 5.3.1, we notice from Figure 15 that the change in loss is irregular. After analyzing dataset 5.3.1, we found out that due to the presence of noisy data points the loss increased.

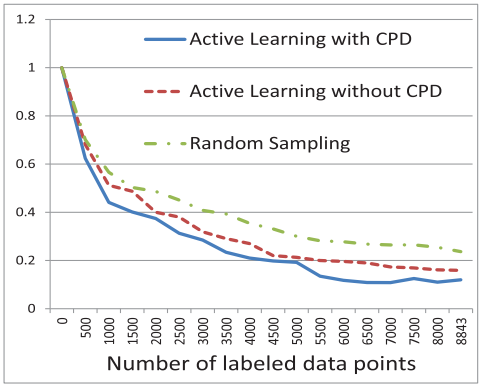

Fig. 16. Different active-learning techniques for dataset 5.3.2.

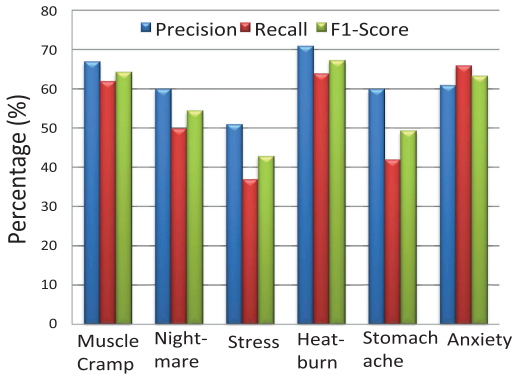

Fig. 17. Different attributes classification result in getting up & sitting partition.

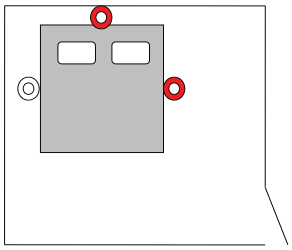

Fig. 18. Visual illustration of sensor activation.

Table 6. Fleiss Kappa Score (Inter user agreement)

Class	Kappa	Z-Score	P-Value
Sleep	0.468	2.8	0.167
Awake	0.423	5.97	0.013
Movement	0.447	4.52	0.00678
Getting up & Sitting	0.537	2.298	0.051
Getting up from the bed	0.835	1.256	0.0006

5.4.3 Crowdsourcing Experiments. During this process, we faced a challenge regarding what kinds of data to show that can reflect the sleep classes. As audio, video, or image data violate the privacy of the user so we had to come up with a different methodology rather than traditional image-based crowdsourcing. We presented some semantic information from the users sleeping habits (regular hours of sleep, sleep latency, posture, average number of times the user gets up at night, how much the user moves on average in percentage and light condition) and a visual illustration of sensor activation (discussed in Section 5.2) to the annotators. In Figure 18, we show an example of visual illustration. The double circled objects represent sensors, and the activation is marked by red color. In this example, the sensor mounted near the head and the sensor mounted near the right side of the bed are activated as the user was getting up from the bed. Ten annotators participated in our crowdsourcing experiment with a dataset containing 10,000 data instances from dataset 5.3.2. In Table 6, the kappa coefficient, z-score, and p-value for individual classes are shown. The kappa coefficients of sleep, awake, and movement classes are considerably low than other two classes. This was due to the nature of these activities and sensitivity of the motion sensor. During our experiment, we have seen that movements with smaller intensities while in sleep or awake is sometimes not captured by the motion sensor. As a result, most of the annotators defined those data instances as sleep. However, movements with higher intensities while sleeping are annotated as awake, movement, and getting up & sitting. Also, the first three activities are related to the state lying. As a result, the posture information using inclination measurements did not help much. For getting up from bed, the visual illustrations were less noisy and easier to depict, and as a result the inter user agreement score is much better.

5.4.4 Introduction of New Unseen Class and Attributes. A user is able to personalize the model by introducing new unseen classes and attributes with the help of active learning. We simulated our active-learning algorithm by introducing new class labels in the classification model. While collecting the query label we also asked for the reason behind choosing the label from the annotator, so that we can look for important indicators for the clinicians. We restricted the length of the reason in five words. For example, if a sample is queried and the annotator labels the sample as *getting up and sitting*, then he or she can also state the reason for labeling the data such as *muscle cramp, stress or anxiety, nightmare*, and so on, which are microscopic events for sleep disruption. We applied a nested classification by considering these microscopic events as class labels. After classifying using our defined general class labels, we partitioned each class label data and applied our classification algorithm in separate partitions again by considering the provided attributes as labels. For example, let us assume an user states reason "A" as the cause of sleep disruption or any kinds of changes in the pattern. Our framework then partitions the data, and the number of partition is equal to the number of class labels (in our model it is 5), and, as a result, in each partition the data points are of same class. The Sleep Well framework then performs a classification on separate partitions with class label "A." This nested classification process ascertains the microscopic sleep events. The precision, recall, and F1 score of recorded attributes (muscle cramp, heatburn, stomach ache, stress, anxiety, and nightmare) for the parent class *"getting up and sitting*" are presented in Figures 16 and 17.

6 SLEEP SCORING

Various sleep scoring models have been proposed over the years, like PSQI (Buysse et al. 1989), in Webster et al. (1982), the Sleep Quality Scale (Cole et al. 1992; Sadeh et al. 1994; YI et al. 2006), "ZQ" of Zeo (2003), which takes in consideration total sleep, deep sleep, REM sleep, wake time, and the number of times woken up. The Webster et al. (1982) and Cole-Kripke (Cole et al. 1992; Jean-Louis et al. 1997; Sadeh et al. 1994) models use the knowledge of actigraphy. In actigraphy, the presence of movements indicates wakefulness and the absence of movements indicates sleep. Sleep efficiency is then calculated by taking the ratio of time slept versus total time spent in bed. PSQI (Buysse et al. 1989) is designed for assessing long-term sleep quality. PSQI contains 19 questions regarding the habit of sleep and the events that cause sleep disruption. The habit of sleep includes sleeping time, sleep latency, sleep duration, and wake up time. From our classification results, we can easily find out the trend of user's sleep (sleeping time, wake up time, sleep duration). In our framework, we are collecting the reasons for sleep disruption using active learning, which also help in answering the questions related to trouble in sleep. In our work, we calculate the sleep–wake cycle of the user using the Cole-Kripke (Cole et al. 1992) algorithm to verify the sleep duration and try to find answers to as many questions as possible from PSQI (Buysse et al. 1989).

In the Cole-Kripke algorithm, the sleep–wake state at any epoch is calculated by considering the previous 4 minutes' and next 2 minutes' actigraphy information. The model is defined by the following equation:

$$D = P(W_{-4}A_{-4} + W_{-3}A_{-3} + W_{-2}A_{-2} + W_{-1}A_{-1} + W_0A_0 + W_{+1}A_{+1} + W_{+2}A_{+2}). \quad (22)$$

Here P denotes the scaling factor, and W_{-i}, W_0, and W_{+i} are the weighting factors for the previous minute, the present minute, and the following minute. The activity scores for the previous, present, and following minute are expressed by A_{-i}, A_0, and A_{+i}. If $D < 1$, then the state is considered to be sleep and for $D \geq 1$ the state is wake. We will adapt the solution for these parameters proposed in Cole et al. (1992) developed by Webster et al. (1982). The rule of assigning scores are as follows: (a) after 4 minutes scored as wake, next 1 minute is also scored as wake, (b) after 10 minutes scored as wake, the next 3 minutes are also scored as wake, (c) after 15 minutes scored as wake, the next

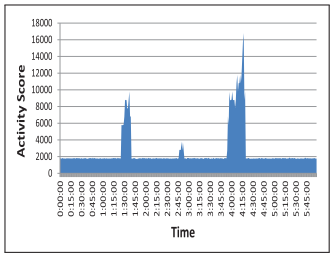

Fig. 19. Activity score for timestamps between 12:00 AM and 6:00 AM.

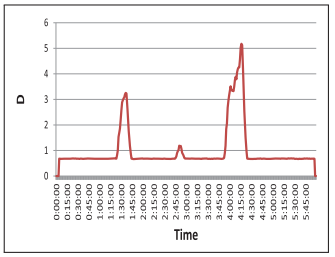

Fig. 20. Calculated D from corresponding activity score using Equation (23).

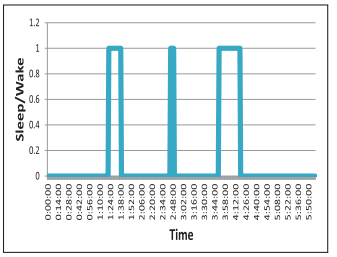

Fig. 21. Sleep–wake estimation from calculated D between 12:00 AM and 6:00 AM.

4 minutes are scored as wake, (d) 6 minutes or less surrounded by at least 10 minutes scored as wake (before and after) are also scored as wake, and (e) 10 minutes or less surrounded by at least 20 minutes scored as wake (before and after) are also scored wake. Then regression was applied for solving Equation (20) using the scores calculated from dataset 5.3.1. After fitting the values in Equation (20), we get the following equation:

$$D = 0.00001(404A_{-4} + 598A_{-3} + 326A_{-2} + 441A_{-1} + 1408A_0 + 508A_{+1} + 350A_{+2}), \quad (23)$$

where activity score A denotes the number of epochs with movements in that particular minute. We calculate the state of each minute using 21 and find the number of minutes the user was sleeping. After that we calculate the following components of PSQI:

(1) We identify the usual time to go to bed at night from the user log diary and validate it with the mounted motion sensor firing sequence. Our framework maintains two separate variables for the motion sensors mounted on the sides of the bed, r and l. On firing of the sensors, we increase the variables. If $\frac{|r-l|}{2}$ is even, then the user got in and then got out of the bed, and if odd, then the user is in in the bed.
(2) Usual slack time for the user to fall asleep by subtracting the timestamp of answer (a) from the usual time the user fell asleep. Additionally, we record the slack time for each day.
(3) Usual time the user gets up in the morning.
(4) Hours of sleep at each night.
(5) We calculate the number of days the user could not sleep within 30 minutes using the information in (b).
(6) How many times the user woke up in the middle of the night or early morning using our classification model and the motion sensor data.
(7) How many times the user got up to use the bathroom using active learning and attribute detection (Section 5.4.4).
(8) Cannot breathe comfortably using active learning.
(9) Feel too cold and too hot using active learning.
(10) Have bad dreams or nightmare using active learning.
(11) Have pain using active learning.
(12) Other reasons for the sleep disruption using active learning.

Rest of the components of PSQI are subjective that we could not measure using our framework. But we are able to automate most of the components with proper validation. Queries 8–12 are not posed in real time. We pose these queries later the following day. In Figures 19, 20, and 21,

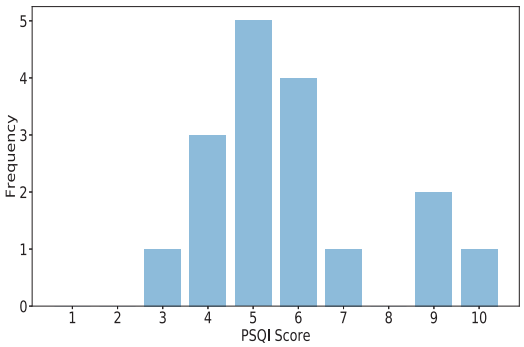

Fig. 22. Histogram of PSQI scores calculated using actigraphy.

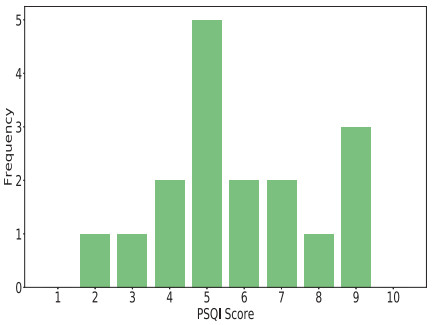

Fig. 23. Histogram of PSQI scores using PSQI questionnaire.

the activity score, D score, and sleep–wake state estimation from the D score is plotted. From the figures, it is visible that the participant was awake thrice. We verified our findings using the participant's log diary, where it was stated that at around 1:30 AM and 3:45 AM he woke up to use the restroom. Around 2:45 AM we see a spike in accelerometer data and the motion sensor near the head also fired, but the participant did not mention anything about staying awake during this time. After calculating the PSQI components, we calculate the final PSQI score. Figures 22 and 23 shows the frequency distribution of PSQI scores using actigraphy and the originial questionnaire for 17 participants, respectively. Although the distributions look quite similar, the population size of 17 is not statistically significant. We want to investigate and prove our hypothesis with a larger population size in the future.

7 DISCUSSIONS AND FUTURE DIRECTIONS

In the current version of the Sleep Well framework, we did not discuss individual sleep scoring based on our sleep state classification. Most of the sleep scoring models like the PSQI (Buysse et al. 1989) or the Webster scale (Webster et al. 1982) do not consider the habit of an individual's sleep. In our experiment, we found that one participant was moving frequently while sleeping and woke up 2 or 3 times at night to use the bathroom. Even if the participant had disrupted sleep according to our data, according to the participant's feedback he had a sound sleep. We look forward to investigating the sleep habits of individuals (like movements and getting up frequently) using change point detection in the future and devising a dynamic sleep scoring module. Different components of PSQI such as PSQIDURAT (duration of sleep), PSQIDISTB (sleep disturbance), PSQILATEN (sleep latency), PSQIDAYDYS (day dysfunction due to sleepiness), PSQIHSE (sleep efficiency), PSQISLPQUAL (overall sleep quality), and PSQIMEDS (need meds to sleep) can be presented in the form of a tree data structure, where each component represents a node in the tree. The outcome of the diagnostic procedures obtained from the set of questions being asked can help calculate the average score of each of the components and can be assigned as a conditional statement to each of the branch.

In the current version of our framework, we invoke the change point detection only in the case of active learning. As a future research direction, it is possible to combine change point detection with a classification model to perform a more thorough time series analysis. For example, abrupt sleep disturbances (muscle cramp, nocturnal panic attacks, etc.) cause sudden changes in the data points. It would be beneficial to capture these subtle changes and correlate the run length with the classification process.

In the current implementation, we considered only the inclination of the device to infer the user's current posture. Therefore, another future direction is to integrate locomotive activity (such as sitting, standing, walking, etc.) recognition with our framework to improve the noise reduction methodology. The orientation and the positioning of the devices affects the performance of the model as well. We would like to investigate the transfer-learning algorithm to address this issue in our future work. For active-learning experiments, we assumed that the user will always provide the correct label. In real life, it is possible that the user may provide the wrong label or leave it blank. Also, while collecting the attributes of sleep disruption causes, we are using a hard coded attribute list. If the labeler provides different attributes, then the model fails to assimilate them. In such cases, imperfect annotation handling and optimal querying can be further studied to improve the performance of active learning. This is also applicable for crowdsourcing the sleep data for malicious user identification. The visual illustration provided while applying crowdsourcing was also not able to provide an accurate representation in some cases. Although we provided semantic information about the user's sleep routine, it did not improve the annotation performance, as most of the annotators ignored the information and provided feedback based on the visual illustration only. Without the ground-truth information, this runs the risk of introducing noisy instances into the model. We want to investigate whether adding other modalities like a highly sensitive motion sensor and smart home data (appliance usages) can improve the performance of the annotators in the future. In our current state, we collect the reason of the label from the user and extract the important microscopic attributes from the provided reason. For the future, theis feedback can be further leveraged to inspect the nature of various sleep disturbances at the microscopic level, which will greatly help in longitudinal sleep assessment and diagnoses.

7.1 Reliability of Sleep Technologies

The growing pervasiveness of off-the-shelf sensor-rich wearable and mobile devices in our daily lives presents the convenience to capture the underlying contextual information of activities of daily living inconspicuously. However, the variations of these commercial devices with respect to device manufacturers, CPU powers, and OS types pose serious challenges in the applicability of these technologies in the same settings across different domains. The major type of heterogeneity that is evident is the different sampling frequencies of these devices (Stisen et al. 2015). Also, most of the off-the-shelf devices are designed to be appropriate for clinical studies. In our experiments, the two devices that we used (Actigraph and Chronos) are research devices that provide much higher sampling frequency than the commercially wearable devices. Although the use of these devices in clinical studies is not reliable, these modalities can compliment clinical study by providing extra evidence. For example, we have shown in Section 6 that we can provide further evidence for some of the questions that the participants are supposed to answer for the PSQI test (Buysse et al. 1989). However, we cannot always answer the subjective aspect of clinical studies using actigraphy, as discussed in Section 6. Lockley et al. (1999) also showed that by comparing the actigraphic and subjective measurements of sleep. They found that actigraphic sleep monitoring is inferior in calculating the sleep–wake cycle, sleep latency, duration of night awakenings, and sleep onset. The actigraphic approach misclassifies inactivity as sleep; for example, just lying down and watching television and not moving much will be regarded as sleep and creates ambiguity in the overall calculation of sleep performance. From this perspective, actigraphic approaches might seem unreliable; however, we can overcome this problem by monitoring the daily sleep routine of a person and augment that information in our classification model. Based on this evidence, we think that wearable devices are still not enough to replace the clinical sleep monitoring method. However, they can certainly endorse other important evidence that can help to aid the clinical trials.

7.2 Reliability of Sleep Apps

Apart from wearable devices, there are couple of mobile applications that provide insights about sleep routines of the users. Sleep As Android (2010), Sleep Time Smart Alarm Clock (2015), and the Sleep Bot (2017) app provide information regarding the sleep–wake cycle and sometimes about the sleep environment as well. These statistics only provide rough estimates and ultimately experience a lack of reliability in measuring the sleep performances. These apps are also now capable of extracting more evidence from paired wearables. Although this improves the inference capability, it does not ensure an improvement in reliability.

8 CONCLUSIONS

In this article, we described the design, implementation, and evaluation of Sleep Well, a sleep monitoring framework that helps classify the microscopic sleep states using wearable devices. We postulated a gradient descent–based approach that incorporates with importance weights aware updates in the microscopic sleep state detection process. We also consolidated our framework by blending change point detection and active learning in the inference pipeline. Our classification achieved 78% accuracy with the aforementioned experimental setup. The empirical results demonstrate the effectiveness of our framework in determining different sleep states. The result increased by 7% when active learning was employed. Our approach helps accelerate the faster convergence to optimal sleep states detection accuracy using minimal user feedback in presence of active learning. In addition, with the help of change point detection, we were able to validate and interpret the transitions between these sleep states. In the future, we plan to investigate the combination of change point detection and classification to further improve the accuracy. Also, conforming the attributes from user-provided feedback into our architecture will help provide meaningful insights for better understanding of sleeping behavior.

REFERENCES

Actigraph. 2004. ActiGraph. Retrieved from http://www.actigraphcorp.com/.

Ryan Prescott Adams and David J. C. MacKay. 2007. *Bayesian Online Changepoint Detection.* Cambridge, UK.

Aeotec Multisensor. 2006. MultiSensor 6: Z-Wave motion, light, temperature sensor - Aeotec. Retrieved from http://aeotec.com/z-wave-sensor.

Hande Özgür Alemdar, Tim van Kasteren, and Cem Ersoy. 2011. Using active learning to allow activity recognition on a large scale. In *Proceedings of the 2nd International Joint Conference on Ambient Intelligence (AmI'11).* 105–114.

Android Wear. 2014. Wear OS by Google Smartwatches. Retrieved from http://www.android.com/intl/en_us/wear/.

Salikh Bagaveyev and Diane J. Cook. 2014. Designing and evaluating active learning methods for activity recognition. In *Proceedings of the 2014 ACM Conference on Ubiquitous Computing (UbiComp'14).* 469–478.

Yin Bai, Bin Xu, Yuanchao Ma, Guodong Sun, and Yu Zhao. 2012. Will you have a good sleep tonight?: Sleep quality prediction with mobile phone. In *Proceedings of the 7th International Conference on Body Area Networks.*

Basis Band B1. 2014. Intel Basis. Retrieved from http://www.mybasis.com/.

Beddit. 2015. Beddit Sleep Monitor. Retrieved from http://www.beddit.com/features/.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. 2009. Importance weighted active learning. In *Proceedings of the International Conference on Machine Learning (ICML'09).* 49–56.

Marko Borazio, Eugen Berlin, Nagihan Kücükyildiz, Philipp M. Scholl, and Kristof Van Laerhoven. 2014. Towards benchmarked sleep detection with inertial wrist-worn sensing units. In *Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI'14).*

D. Buysse, C. Reynolds, T. Monk, S. Berman, and D. Kupfer. 1989. The pittsburgh sleep quality index: A new instrument for psychiatric practice and research. In *Psychiatry Research*, Vol. 28. 193–213.

Yung-Ju Chang, Gaurav Paruthi, Hsin-Ying Wu, Hsin-Yu Lin, and Mark W. Newman. 2017. An investigation of using mobile and situated crowdsourcing to collect annotated travel activity data in real-world settings. *Int. J. Hum.-Comput. Stud.* 102 (2017), 81–102.

Zhenyu Chen, Mu Lin, Fanglin Chen, N. D. Lane, et al. 2013. Unobtrusive sleep monitoring using smartphones. In *PervasiveHealth.* 145–152.

Anand Inasu Chittilappilly, Lei Chen, and Sihem Amer-Yahia. 2016. A survey of general-purpose crowdsourcing techniques. *IEEE Trans. Knowl. Data Eng.* 28, 9 (2016), 2246–2266.

R. J. Cole, D. F. Kripke, W. Gruen, D. J. Mullaney, and J. C. Gillin. 1992. Automatic sleep/wake identification from wrist activity. *Sleep* 15, 5 (Oct. 1992), 461–469.

Florian Daiber and Felix Kosmalla. 2017. Tutorial on wearable computing in sports. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'17)*. 65:1–65:4.

Andrew L. Chesson et al. 1997. Practice parameters for the indications for polysomnography and related procedures. *Sleep* 20, 6 (1997), 406–422.

B. Darkhovsky, J. Fell, A. Kaplan, and J. Röschke. 2001. Macrostructural EEG characterization based on nonparametric change point segmentation: Application to sleep analysis. *J. Neurosci. Methods* 106, 1 (2001), 81–90.

EZ430-Chronos. 2013. Chronos: Wireless development tool in a watch. Retrieved from http://www.ti.com/tool/ez430-chronos.

Muhammad Fahim, LeBa Vui, Iram Fatima, Sungyoung Lee, and Yongik Yoon. 2013. A sleep monitoring application for u-lifecare using accelerometer sensor of smartphone. In *Ubiquitous Computing and Ambient Intelligence: Context-Awareness and Context-Driven Interaction*. Vol. 8276. 151–158.

Fitbit. 2007. Fitbit Official Site for Activity Trackers and More. Retrieved from http://www.fitbit.com/.

N. Foubert, A. M. McKee, R. A. Goubran, and F. Knoefel. 2012. Lying and sitting posture recognition and transition detection using a pressure sensor array. In *Proceedings of the Conference on Medical Measurements and Applications (MeMeA'12)*. 1–6.

Weixi Gu, Zheng Yang, Longfei Shangguan, Wei Sun, Kun Jin, and Yunhao Liu. 2014. Intelligent sleep stage mining service with smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'14)*. 649–660.

Tian Hao, Guoliang Xing, and Gang Zhou. 2013. iSleep: Unobtrusive sleep quality monitoring using smartphones. In *SenSys*. Article 4, 4:1–4:14 pages.

Hello Sense. 2014. Hello Sense. Retrieved from https://hello.is/.

Enamul Hoque and John A. Stankovic. 2010. Monitoring quantity and quality of sleeping using WISPs. In *Proceedings of the Information Processing in Sensor Networks Conference (IPSN'10)*. 370–371.

Enamul Hoque and John A. Stankovic. 2012. AALO: Activity recognition in smart homes using active learning in the presence of overlapped activities. In *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare, PervasiveHealth 2012*. 139–146.

H. M. Sajjad Hossain, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. 2017. Active learning enabled activity recognition. *Perv. Mobile Comput.* 38 (2017), 312–330.

H. M. Sajjad Hossain, Nirmalya Roy, and Md Abdullah Al Hafiz Khan. 2015. Sleep well: A sound sleep monitoring framework for community scaling. In *Proceedings of the 16th IEEE International Conference on Mobile Data Management (MDM'15)*. 44–53.

Kyuwoong Hwang and Soo-Young Lee. 2012. Environmental audio scene and activity recognition through mobile-based crowdsourcing. *IEEE Trans. Consum. Electron.* 58, 2 (2012), 700–705.

G. Jean-Louis, H. von Gizycki, F. Zizi, A. Spielman, P. Hauri, and H. Taub. 1997. The actigraph data analysis software: I. A novel approach to scoring and interpreting sleep-wake activity. *Percept. Mot. Skills* 85, 1 (Aug. 1997), 207–216.

George H. John, Ron Kohavi, and Karl Pfleger. 1994. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*. 121–129.

Murray W. Johns. 2000. Sensitivity and specificity of the multiple sleep latency test (MSLT), the maintenance of wakefulness test and the Epworth sleepiness scale: Failure of the MSLT as a gold standard. *J. Sleep Res.* 9, 1 (2000), 5–11.

Stephan Jonas, Andreas Hannig, Cord Spreckelsen, and Thomas M. Deserno. 2014. Wearable technology as a booster of clinical care. In *Medical Imaging 2014: PACS and Imaging Informatics: Next Generation and Innovations*, Vol. 9039. International Society for Optics and Photonics.

Nikos Karampatziakis and John Langford. 2011. Online importance weight aware updates. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI'2011)*. 392–399.

Matthew Kay, Eun K. Choe, and et al. 2012. Lullaby: A capture & access system for understanding the sleep environment. In *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing (UbiComp'12)*.

E. B. Klerman, R. Barbieri, L. Citi, and M. T. Bianchi. 2011. Instantaneous monitoring of sleep fragmentation by point process heart rate variability and respiratory dynamics. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 7735–7738.

Christopher Kline. 2013. *Sleep Quality*. Springer, New York, NY, 1811–1813. DOI: http://dx.doi.org/10.1007/978-1-4419-1005-9_849

Chih-En Kuo, Yi-Che Liu, Da-Wei Chang, Chung-Ping Young, Fu-Zen Shaw, and Sheng-Fu Liang. 2017. Development and evaluation of a wearable device for sleep quality assessment. *IEEE Trans. Biomed. Eng.* 64, 7 (2017), 1547–1557.

Nicholas D. Lane, Mashfiqui Mohammod, and et al. 2011. Bewell: A smartphone application to monitor, model and promote wellbeing. In *Pervasive Computing Technologies for Healthcare*.

Wen-Hung Liao and Chien-Ming Yang. 2008. Video-based activity and movement pattern analysis in overnight sleep studies. In *Proceedings of the International Conference on Pattern Recognition (ICPR'08)*. 1–4.

Zhicheng Liao and Yangyong Zhu. 2014. When a classifier meets more data. *Proc. Comput. Sci.* 30 (2014), 50–59.

J. J. Liu, Wenyao Xu, and et al. 2013. A dense pressure sensitive bedsheet design for unobtrusive sleep posture monitoring. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'13)*. 207–215.

Steven W. Lockley, Debra J. Skene, and Josephine Arendt. 1999. Comparison between subjective and actigraphic measurement of sleep and sleep rhythms. *J. Sleep Res.* 8, 3 (1999), 175–183.

Janna Mantua, Nickolas Gravel, and Rebecca Spencer. 2016. Reliability of sleep measures from four personal health monitoring devices compared to research-based actigraphy and polysomnography. *Sensors* 16, 5 (2016), 646.

Shun Matsui, Tsutomu Terada, and Masahiko Tsukamoto. 2017. Smart eye mask: Eye-mask shaped sleep monitoring device. In *Adjunct Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers (UbiComp/ISWC'17)*. 265–268.

Mimo. 2016. Mimo Baby. Retrieved from http://mimobaby.com/.

K. Nakajima, Y. Matsumoto, and T. Tamura. 2000. A monitor for posture changes and respiration in bed using real time image sequence analysis. In *Engineering in Medicine and Biology Society*, Vol. 1.

Yunyoung Nam, Yeesock Kim, and Jinseok Lee. 2016a. Sleep monitoring based on a tri-axial accelerometer and a pressure sensor. *Sensors* 16, 5 (2016), 750.

Yunyoung Nam, Yeesock Kim, and Jinseok Lee. 2016b. Sleep monitoring based on a tri-axial accelerometer and a pressure sensor. *Sensors* 16, 5 (2016), 750. https://doi.org/10.3390/s16050750.

Anh Nguyen, Raghda Alqurashi, Zohreh Raghebi, Farnoush Banaei-kashani, Ann C. Halbower, and Tam Vu. 2016. A lightweight and inexpensive in-ear sensing system for automatic whole-night sleep stage monitoring. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*. ACM, 230–244.

N. R. Oakley. 1997. Validation with polysomnography of the sleep-watch sleep/wake scoring algorithm used by the actiwatch activity monitoring system. In *Technical Report to Mini Mitter Co., Inc.*

Abdulakeem Odunmbaku, Amir-Mohammad Rahmani, Pasi Liljeberg, and Hannu Tenhunen. 2016. *Elderly Monitoring System with Sleep and Fall Detector*. Springer International Publishing, Cham, 473–480. DOI: http://dx.doi.org/10.1007/978-3-319-47063-4_51

James M. Parish. 2009. Sleep-related problems in common medical conditions. *CHEST J.* 135, 2 (2009), 563–572.

Rachael Purta, Stephen Mattingly, Lixing Song, Omar Lizardo, David Hachen, Christian Poellabauer, and Aaron Striegel. 2016. Experiences measuring sleep and physical activity patterns across a large college cohort with fitbits. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC'16)*. 28–35.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *J. Mach. Learn. Res.* 11 (Aug. 2010), 1297–1322.

Yanzhi Ren, Chen Wang, Jie Yang, and Yingying Chen. 2015. Fine-grained sleep monitoring: Hearing your breathing with smartphones. In *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM'15)*. 1194–1202.

Mahsan Rofouei, Mike Sinclair, Ray Bittner, Tom Blank, Nick Saw, Gerald DeJean, and Jeff Heffron. 2011. A non-invasive wearable neck-cuff system for real-time sleep monitoring. In *Proceedings of the IEEE Conference on Body Sensor Networks (BSN'11)*. IEEE, 156–161.

A. Sadeh. 2011. The role and validity of actigraphy in sleep medicine: An update. *Sleep Med. Rev.* 15, 4 (Aug. 2011), 259–267.

A. Sadeh, K. M. Sharkey, and M. A. Carskadon. 1994. Activity-based sleep-wake identification: An empirical test of methodological issues. *Sleep* 17, 3 (Apr. 1994), 201–207.

Sohrab Saeb, Thaddeus R. Cybulski, Konrad P. Kording, and David C. Mohr. 2017. Scalable passive sleep monitoring using mobile phones: Opportunities and obstacles. *J. Med. Internet Res.* 19, 4 (2017), e118.

Alireza Sahami Shirazi, James Clawson, Yashar Hassanpour, Mohammad J. Tourian, Albrecht Schmidt, Ed H. Chi, Marko Borazio, and Kristof Van Laerhoven. 2013. Already up? Using mobile phones to track & share sleep behavior. *Int. J. Hum.-Comput. Stud.* 71, 9 (Sept. 2013), 878–888.

Aarti Sathyanarayana, Ferda Ofli, Luis Fernandes-Luque, Jaideep Srivastava, Ahmed K. Elmagarmid, Teresa Arora, and Shahrad Taheri. 2016. Robust automated human activity recognition and its application to sleep research. In *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016*. Barcelona, Spain, 495–502. https://doi.org/10.1109/ICDMW.2016.0077.

Aarti Sathyanarayana, Jaideep Srivastava, and Luis Fernández-Luque. 2017. The science of sweet dreams: Predicting sleep efficiency from wearable device data. *IEEE Comput.* 50, 3 (2017), 30–38.

Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers.

Sleep As Android. 2010. Sleep As Android. Retrieved from https://play.google.com/store/apps/details?id=com.urbandroid.sleep&hl=en.

Sleep Bot. 2017. SleepBot - Smart Alarm - Movement Tracker - Sound Recorder. Retrieved from http://mysleepbot.com.

Sleep Time Smart Alarm Clock. 2015. Sleep Time : Sleep Cycle Smart Alarm Clock Tracker. Retrieved from https://play.google.com/store/apps/details?id=com.azumio.android.sleeptime.

Margarita Sordo and Qing Zeng. 2005. On sample size and classification accuracy: A performance comparison. In *Proceedings of the 6th International Conference on Biological and Medical Data Analysis (ISBMDA'05)*. 193–201.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15)*. 127–140.

Xiao Sun, Li Qiu, Yibo Wu, Yeming Tang, and Guohong Cao. 2017. SleepMonitor: Monitoring respiratory rate and body position during sleep using smartwatch. *Interact. Mobile Wear. Ubiq. Technol.* 1, 3 (2017), 104:1–104:22.

Kristof Van Laerhoven, Marko Borazio, David Kilian, and Bernt Schiele. 2008. Sustained logging and discrimination of sleep postures with low-level, wrist-worn sensors. In *Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers (ISWC'08)*. 69–76.

Oana Ramona Velicu, Natividad Martínez Madrid, and Ralf Seepold. 2016. Experimental sleep phases monitoring. In *Proceedings of the 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI'16)*. IEEE, 625–628.

Vowpal Wabbit. 2016. Vowpal Wabbit (Fast Learning) - Machine Learning (Theory). Retrieved from http://hunch.net/ vw/.

J. B. Webster, D. F. Kripke, et al. 1982. An activity-based sleep monitor system for ambulatory use. In *Sleep*, Vol. 5. 389–399.

Hyeryeon Yi, Kyungrim Shin, and Chol Shin. 2006. Development of the sleep quality scale. *J. Sleep Res.* 15, 3 (2006), 309–316. DOI : http://dx.doi.org/10.1111/j.1365-2869.2006.00544.x

Zeo. 2003. Zeo, Inc. Retrieved from http://www.myzeo.com/sleep.

Jin Zhang, Qian Zhang, Yuanpeng Wang, and Chen Qiu. 2013. A real-time auto-adjustable smart pillow system for sleep apnea detection and treatment. In *Proceedings of the 2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'13)*. 179–190. DOI : http://dx.doi.org/10.1109/IPSN.2013.6917584

Received May 2017; revised January 2018; accepted February 2018