IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 19, 2019, accepted September 7, 2019, date of publication September 20, 2019, date of current version October 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942608

A Wearable Sleep Position Tracking System Based on Dynamic State Transition Framework

SANGHOON JEON[1], TAEJOON PARK[2], (Member, IEEE), ANAND PAUL[3], (Senior Member, IEEE), YANG-SOO LEE[4], AND SANG HYUK SON[1], (Fellow, IEEE)

[1]Department of Information and Communication Engineering, DGIST, Daegu 42988, South Korea
[2]Department of Robotics Engineering, Hanyang University, Seoul 15588, South Korea
[3]School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, South Korea
[4]Department of Physical Medicine and Rehabilitation, School of Medicine, Kyungpook National University, Daegu 41566, South Korea

Corresponding authors: Taejoon Park (taejoon@hanyang.ac.kr) and Sang Hyuk Son (son@dgist.ac.kr)

This work was supported in part by the Global Research Laboratory Program under Grant 2013K1A1A2A02078326 through NRF, in part by the ICT Research and Development program of MSIP/IITP under Grant 2014-0-00065, (Resilient Cyber-Physical Systems Research), and in part by the DGIST Research and Development Program of the Ministry of Science, ICT and Future Planning under Grant 18-EE-01.

ABSTRACT Sleep monitoring is vital as sleep plays an important role in recovering physical and mental health. To have a sound sleep, one has to avoid bad sleep positions associated with personal health conditions. However, most of the existing sleep trackers merely show quantitative information about sleep patterns and duration at each sleep stage, overlooking the importance of sleep positions upon sleep quality. To accurately keep track of sleep positions, we propose a wearable sleep position tracking system consisting of two wristbands and one chest-band. We suggest a two-level classifier specialized for sleep motion based on Dynamic State Transition (DST)-framework. The DST-framework is designed to process the spatio-temporal sleep motion data collected via accelerometer/gyro sensing and classify twelve sleep position (SP) motions from four sleep positions. Our experimental results demonstrate that the proposed system effectively and accurately classify twelve SP motions for tracking sleep positions, and hence, serves as a key building block for comprehensive sleep care applications related to sleep positions.

INDEX TERMS Sleep position, sleep quality, sleep monitoring, wearable devices.

I. INTRODUCTION

Sleep is one of the major activities of daily living and occupies one-third of our life. Sound sleep is essential for us since it plays a vital role in recovering physical and mental health [1]. Polysomnography (PSG) is a well-known standard sleep diagnosis tool that figures out sleep disorders by using a number of physiological sensors including electroencephalogram (EEG), electrooculogram (EOG), electromyogram (EMG), electrocardiogram (ECG), and pulse oximetry [2]. However, PSG may not be practically useful because sleeping with many sensors is inconvenient and undergoing sleep experiments in the dedicated sleep facility may not precisely reflect real sleep problems.

Recent advances in sensing, wireless communications, and data analytics have opened up new possibilities for using wearable devices like fitness trackers to analyze sleep disorders, i.e., a portable PSG. Besides, smart beds/mattresses, and external bedside monitors such as camera, light, noise, temperature optimize the sleep environment for sound sleep. These devices are ultimately aimed at enhancing sleep by measuring sleep quantity and quality.

The National Sleep Foundation (NSF) addressed sleep quantity recommendations across the lifespan, e.g., 10 to 13 hours for teenagers and 7 to 9 hours for adults [3]. Also, sleep variables such as sleep latency, number of awakenings>6 minutes, wake after sleep onset and sleep efficiency are used as indicators of sleep quality [4].

Sleep disorders regularly interfere with sleep, causing fragmented sleep and excessive daytime sleepiness, which eventually leads to poor quality of life. There are many types of sleep disorders: insomnia, sleep-related breathing disorders, central disorders of hypersomnolence, circadian rhythm sleep-wake disorders, sleep-related movement disorders, and

The associate editor coordinating the review of this manuscript and approving it for publication was Kemal Polat.

parasomnias [5]. The first step in overcoming sleep disorders is to identify and track sleep patterns. Self-help remedies, such as a sleep diary and lifestyle adjustments, can help. However, severe sleep disorders require medical treatments such as sleeping pills and surgery by a sleep specialist.

Most sleep gadgets are designed to monitor and regulate 24-hour sleep-wake cycles, a.k.a., circadian rhythms, lacking a function of tracking sleep positions. However, sleep positions are directly related to some sleep disorders. For example, a supine position causes unfavorable airway geometry and reduces lung volume, often leading to obstructive sleep apnea syndrome (OSAS) [6]. In contrast, infants placed supine is recommended because prone and side positions are vulnerable to sudden infant death syndrome (SIDS) [7]. Pregnant women are advised to sleep on their side, especially on the left side for maximum blood flow. Besides, a pressure ulcer means a skin wound due to prolonged pressure on the skin. Patients with pressure ulcer can relieve the constant pressure on the same areas of the body by repositioning regularly [8].

Clearly, the specific sleep position depending on sleep disorders has a negative influence on one's health, thus should be avoided. As a simple and effective means of therapy, positional therapy (PT) is given to mild OSAS or snoring patients [9]. PT uses a pajama with a tennis ball on the back, making a patient uncomfortable to sleep in a supine position. However, a sleep tracking system is still needed to effectively manage sleep positions by monitoring changes in sleep positions during the night.

For a practical and accurate solution of tracking sleep positions, it is important to address the following motion characteristics to deal with the reality associated with sleep behavior. First, sleep motions show *discontinuity* characteristics. Sleep habits such as scratching body and hand stretching are observed discontinuously during sleep, and these motions exacerbate the background noise. Second, typical sleep motions have *uncertainty* characteristics in motion patterns due to unconsciousness. General activity recognition approaches are not appropriate to deal with unconscious sleep motions. Finally, sleep motions have *time-variability* characteristics in the sense that the duration of sleep motions can vary in a wide range. In short, classification of spatio-temporal Sleep Position (SP) motion is challenging since unconscious sleep motion includes much noise such as sleep habits before and after changes in sleep positions.

The key idea to overcome these challenges is to develop a spatio-temporal state transition model that fits well with a sequence of sleep motions by exploiting the fact that there exists a limited range of hand and body movement directions. Based on this idea, we propose a *wearable Sleep POsition Tracking System* (wSPOTS) that utilizes wearable devices to analyze the sleep motion data of both hands and a chest in real-time and infers instantaneous sleep position changes.

In particular, wSPOTS: 1) collects directional features from wearable devices consisting of two wristbands and one chest-band, 2) aggregates these directional features over time, and 3) determines/tracks final sleep positions by systematically processing the spatio-temporal sleep motion information via a hybrid learning algorithm, called Dynamic State Transition (DST)-framework. DST-framework is specially designed to distinguish sleep positions using noisy sleep data effectively. DST-framework divides the sleep motion data into three consecutive states and considers both inter- and intra-state directions for the spatio-temporal classification.

The major contributions of this paper are as follows:

- We propose a system, called wSPOTS, for tracking sleep positions in real-time by classifying sleep motions.
- We design twelve Sleep Position (SP) motion models from four sleep positions, and develop a DST-framework for the spatio-temporal motion classification.
- We propose a two-level classifier based on DST-framework which is a core part of wSPOTS. Depending on Sleep Position Change (SPC) group and NonSPC group at the first level, detailed *SP* motions are classified in each group at the second level. This two-level classifier is designed to improve the classification performance.
- Comprehensive evaluation results in both pilot and onsite experiments show that DST-framework is practical and effective in classifying *SP* motions. In addition, we compare the performance of various combinations of wearable devices.

We envision that wSPOTS will be used as a supplement to existing sleep diagnosis solutions such as PSG, serving as a cost-effective 24/7 monitoring tool for sleep disorders and other sleep applications.

The rest of the paper is organized as follows. Section II reviews the related work. Section III presents the proposed system, called wSPOTS. Section IV evaluates wSPOTS. Section V discusses the limitations and future work. Finally, the paper concludes with Section VI.

II. RELATED WORK

Many types of devices have been developed for tracking sleep positions to help sleep. Research on sleep position tracking mainly uses three device types: smart bed, camera, and wearable device.

First, most of the studies have used smart bed-type devices in the form of sensors installed on or near the mattress: a dense array of the pressure sensor, Inertial Measurement Unit (IMU) sensor, and wireless sensors (WiFi and RFID).

A high-resolution pressure map using Force Sensing Resistor (FSR) sensors was used for classifying three sleep positions such as supine, left lying, and right lying [10]. Three positions except for a prone position were further classified into a total of six postures [11], or four main sleep positions were estimated with Deep Neural Network (DNN) using the pressure map [12]. For specific application use, infant postures (such as prone, supine and seated) were estimated by using a modular pressure-sensitive mat [13]. In the application of pressure ulcers, Pouyan et al. classified eight in-bed postures with a binary pattern matching technique [14] and

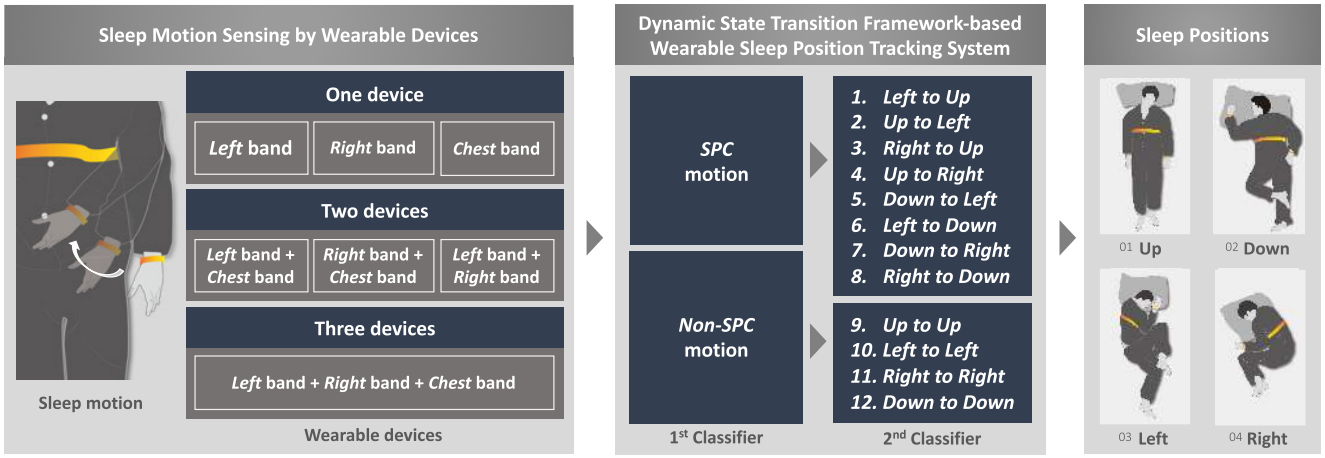

FIGURE 1. Overview of wearable sleep position tracking system (wSPOTS).

three sleep positions with deep learning [15]. Besides, three sleep positions were recognized by using FSR sensor array in the form of a pillow [16].

IMU sensor installed on the mattress is used to analyze sleep motion patterns. Hoque et al. proposed a sleep monitoring system based on Wireless Identification and Sensing Platform (WISP) equipped with accelerometers [17]. The system distinguished four sleep positions by using the y-axis accelerometer readings. MediSense [18] did not directly estimate sleep positions, but inferred patient's motion activities (stay still, arm wave, body rotate, and body shake) from noisy bed motions by using z-axis gyroscope readings.

Wi-Sleep [19] classified a person's respiration, six sleep positions, and rollovers by leveraging WiFi signals, i.e., channel state information (CSI), from a pair of TX-RX. TagSheet [20] used RFID tags on a bedsheet. By using the backscattered radio frequency signals from the bed sheet to an RFID reader, TagSheet identified six sleep positions and estimated respiration rate.

Second, infrared and depth cameras are used to extract high-resolution image data of body posture and movement during sleep and to track sleep positions.

RTPose [21] used an infrared camera and estimated upper leg pose on diverse sleep positions by tracking relative locations of a head and torso on 2D images. On the other hand, a depth camera produces more 3D images than the infrared camera. Thus, feature extraction methods play an important role in the classification performance. From 3D scanned human body images, Scale-Invariant Feature Transform (SIFT) [22] and frequency-based feature selection [23] are used for estimating six sleep positions and two sleep positions, respectively. Grimm *et al.* [24] extracted the Bed Aligned Map from the depth image and classified three sleep positions with Convolutional Neural Networks (CNN).

Lastly, wearable devices that include an IMU sensor are useful for monitoring specific body parts by directly detecting body movements. A straightforward way to monitor sleep positions is to use a chest-band since sleep positions are determined by body posture. Sleep positions were easily classified by setting a simple threshold [25], [26], i.e., the angle range of chest-band. In addition to studies using chest-band, there were also studies using smartwatch-type devices. SleepMonitor [27] was designed to estimate the respiratory rate and four sleep positions by using accelerometer sensors on a smartwatch. It used only static windows and discarded windows containing body movements, and it used three dimensional tilt angles to detect four sleep positions. Similar to SleepMonitor, SleepGuard [28] estimated four sleep positions when a monitoring window is static, i.e., no motion in the window, by using both tilt sensor data and orientation data.

However, existing approaches have the following limitations. First, the accuracy of bed-installed sensors depends heavily on installation conditions such as how many and where the sensors are on the bed as well as on environmental conditions such as movements of a bed partner. Second, camera-based systems have privacy and installation concerns. Lastly, most researches using wearable devices used only one sensor, which limits data collection from multiple parts of the body.

Most related to our approach, SleepMonitor and SleepGuard captured the direction of arm posture according to sleep positions by using one smartwatch. Although we also use similar observations of the fixed sleeping direction of the chest and wrists, we focus on the area of activity, i.e., motion-based recognition. Different from the previous approaches, our wSPOTS aims to classify twelve motions associated with sleep positions from noisy motion data while other approaches aim to estimate sleep positions in static areas. In addition, we compare the performance depending on the combinations of three wearable devices.

III. PROPOSED SYSTEM

wSPOTS is designed to track four sleep positions for enhancing sleep quality. wSPOTS uses wearable devices consisting of three sensors (two wrists and one chest sensors)

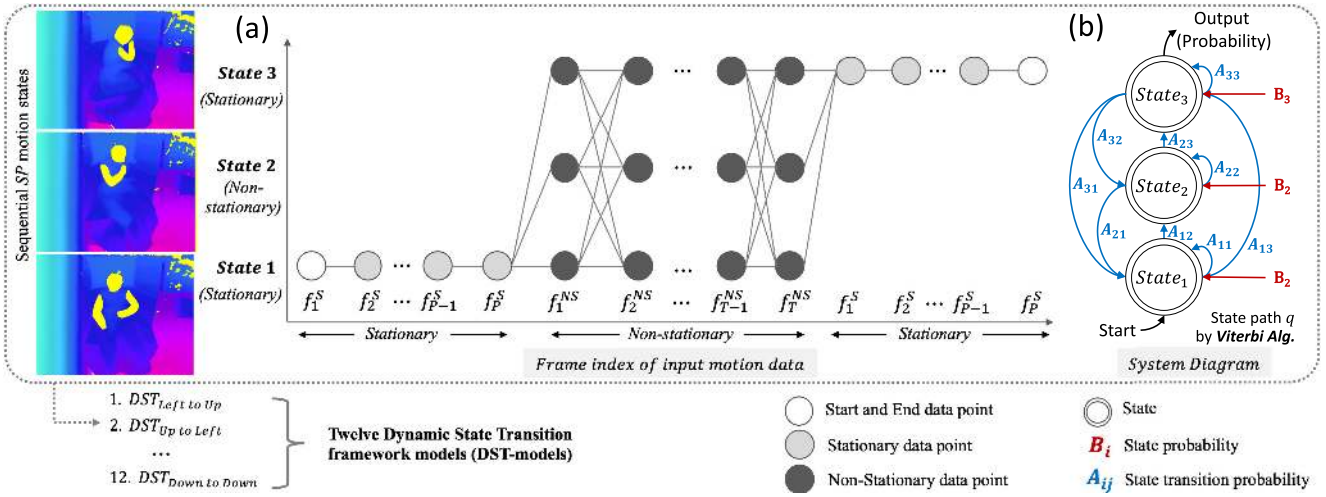

FIGURE 2. Spatio-temporal *SP* motion classification based on Dynamic State Transition (DST)-framework.

to sense sleep motions, classifies the motions based on DST-framework, and then tracks sleep positions as shown in Fig. 1. In this paper, we propose a DST-framework-based classifier specialized for twelve *SP* motions. We also evaluate the classification performance for a total of seven combinations of the three sensors to determine the suitability of the combination.

A. BACKGROUND

Spatio-temporal data refers to data having spatial and temporal information. For the spatio-temporal pattern classification, Hidden Markov Model (HMM) has been widely used [29] and has demonstrated its effectiveness in applications such as speech [30], DNA sequencing [31], and handwritten digits [32].

ST-SVM algorithm was proposed for video-based identification of human behavior [33]–[35]. While traditional HMM requires the recognition target to be precisely described by the observation sequences, ST-SVM is flexible enough to trace the likelihood of sequential states via Support Vector Machine (SVM). Another strength of ST-SVM is to allow a smooth state transition via Markov Random Field (MRF), resulting in much simpler system design yet high performance of classification. ST-SVM basically adopts a left-to-right model so that a state index does not decrease as time goes on.

In this work, we have modified the network parameters (state probability and state transition probability) and the network model of ST-SVM to be suitable for motion data and to improve classification performance. We propose two frameworks, i.e., ST-framework and DST-framework, that differ from ST-SVM.

For ST-framework, we modify network parameters to fit IMU motion data. In more detail, we add a new Random Forest (RF) algorithm to state probability estimation. Also, we use the historical features of the IMU data instead of the localized contour sequence (LCS) features, an image feature, to calculate the state transition probability using the IMU data. The network model of ST-framework follows the network model of ST-SVM, i.e., a left-to-right model.

In DST-framework, we extend the network model in ST-framework to effectively classify actual sleep motions. The network model plays a vital role in calculating an accumulated probability across the network by using network parameters. As shown in Fig. 2(a), we propose a conditioned fully-connected model, and this model is used as a network model in DST-framework. While a fully-connected model enters any other states, the conditioned fully-connected model can reach pre-determined states depending on the conditions of the activity periods. For example, this network model enters only the initial state in a stationary period and can reach any other states (same to the fully-connected model) during a non-stationary period. And then, it arrives the end state in a stationary period.

The conditioned fully-connected of DST-framework has two advantages. First, a fully-connected model during the non-stationary period can collect all possible directional features from all states of a *SP* motion. Actual sleep motions containing sleep habits can be easily transitioned between adjacent states, which makes the fully-connected model more effective than the left-to-right model. Second, a network design with the fixed state during the non-stationary period not only makes a natural flow of the *SP* motion consisting of three states but also reduces confusion between similar *SP* motions, e.g., Up-to-Left and Left-to-Up. The directional features of the two *SP* motions are similar, but the start and end feature are distinctly different.

In our previous work, SleePS [36] tracks sleep positions using deep learning approach (data-driven method) except for *Down* sleep position. We extend this work with a Machine learning approach (model-driven method) to design a sophisticated sleep position tracking algorithm, called DST-framework. Note that the previous work does not consider *Down* sleep position and has limitations such as difficulty in tracking all sleep positions.

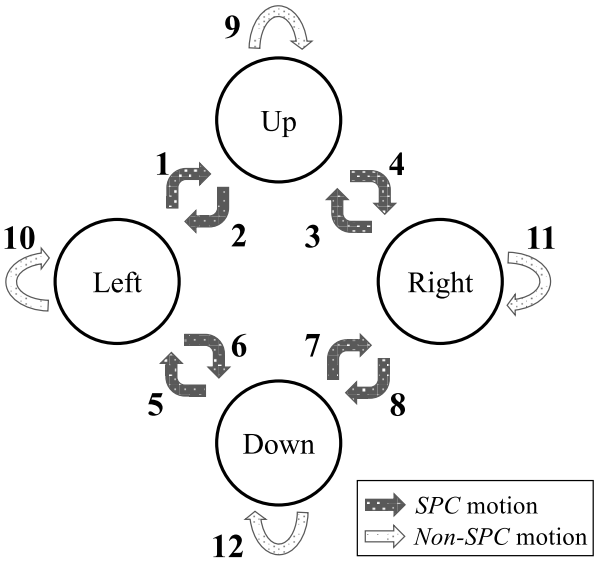

FIGURE 3. Twelve *SP* motion models.

B. DYNAMIC STATE TRANSITION FRAMEWORK

We define twelve *SP* motion models from four sleep positions, as shown in Fig 3. To classify the twelve *SP* motions, DST-framework models belonging to SPC and NonSPC groups are generated according to the decision of the SPC and the NonSPC groups, respectively. Then, the final *SP* motion is determined by comparing the accumulated probabilities across the DST-framework models of the determined group. DST-framework consists of three primary components: Machine Learning (ML)-based state probability, Markov Random Field (MRF)-based transition probability, and Viterbi detection.

To illustrate DST-framework more clearly, we introduce a system diagram of DST-framework. Three components of DST-framework are used for calculating the probability of *SP* motion as shown in Fig. 2(b). Before calculating the probability of *SP* motion, we extract the state transition probability A. After that, given input motion features, the state probability B is estimated. The probability of *SP* motion is calculated by considering two network parameters (A and B) on the network model, and the optimal path q is determined by Viterbi algorithm.

1) ML-BASED STATE PROBABILITY

For spatial state classification, we employ two well-known ML algorithms, i.e., SVM and RF, for the state probability. The posterior probability from the ML algorithms is used as the state probability of the DST-framework. There are four classes in each state, and the posterior probability of the selected class in each *SP* motion is used in DST-framework.

Let c and x denote a state class of DST-framework and 3-axis Gravity Vector (GV), respectively. GV represents the directional features from 9-axis Inertial Measurement Unit (IMU) data. Also let m denote a binary label where $m = 1$ indicates a desired state while $m = -1$ means other states. With input frames T and states N in *Non-Stationary* period, a state probability is specified by a $T \times N$ matrix where each element is given by $Pr(c|x)$. In this work, N is designed as three.

DST-framework uses a conditioned fully-connected model. The first and the end state data of *SP* motion in the non-stationary period are appended at the initial and the ending stationary periods, respectively, using a padding method. This design makes a natural flow between the three states even it has a fully-connected model in the middle network. DST-framework focuses on figuring out *SP* motions where motion occurs, i.e., *Non-Stationary* period. Thus, we set the padding size P to 10% of an input frame size T. State probability with $(P+T+P) \times N$ matrix is generated in DST-framework. To make the state probability using a posterior probability from ML, we use two ML algorithms: SVM and RF. We briefly discuss the two ML algorithms as follows.

We use 2-class SVM to calculate a posterior probability. SVM calculates the state probabilities from incoming feature data by comparing how similar each sleep motion feature is to the trained model [37]. That is, SVM calculates the state probability using a ratio, R of the distance between x and a hyperplane to the marginal distance as follows:

$$R(x) = \beta x + b \tag{1}$$

where β and b are SVM parameters to specify the decision boundary. The posterior probability of SVM given observation x for desired state class c is calculated by converting R into a value between 0 and +1 through the following sigmoid function:

$$Pr(c_{m=+1}|x) = \frac{1}{1 + e^{-R(x_t)}} \tag{2}$$

As one of the highly accurate classifiers, RF is an ensemble classifier consisting of many decision trees. RF compensates the overfitting problem in a decision tree algorithm by a random sampling method called bagging [38]. To implement RF, we use the TreeBagger function in MATLAB and set the number of trees to 200.

$$Pr(c|x) = \frac{1}{\sum_{e=1}^{E} wI(e \in S)} \sum_{e=1}^{E} w\widehat{P}_e(c|x) I(e \in S) \tag{3}$$

where w, e, and E denote the tree weight, tree, and the set of indices in the selected tree that make up the prediction, respectively. $I(e \in S)$ is 1 if e is in the set S, and 0 otherwise. $\widehat{P}_e(c|x)$ is the estimated posterior probability of class c given observation x. We calculate the posterior probability of RF by the weighted average of the class posterior probabilities over the selected trees, as shown in Eq. 3.

2) MRF-BASED STATE TRANSITION PROBABILITY

The state-transition probability quantifies how adjacent motion features belonging to the same state differ from those belonging to other states. When a motion is composed of several consecutive states, it is required that state transitions naturally take place among adjacent states. To meet this requirement for a smooth state transition, we use Markov

Random Field (MRF) to compute the state-transition probabilities such that adjacent and close states have higher state transition probability than faraway states.

In order to consider this smooth flow between the three states, we extract the state transition probability using the histogram H of directional features in each state. The histogram H is extracted in each device according to a total of seven combinations of three wearable devices. For example, a histogram feature $H = [H^{Left}, H^{Right}, H^{Chest}]$ in a combination consisting of three devices has three histogram features.

MRF-based state-transition probability, a_{ij} (for the transition from state i to j) measures the similarity between two states for randomly selected pairs of H_i and H_j, which are made of the same number of GV in each state. Hence, a_{ij} is computed as:

$$\alpha_{ij} = \frac{1}{Z} \prod_{H_i \in i, H_j \in j} \varphi(X_{H_i H_j}) \tag{4}$$

where

$$Z = \sum_{j} \prod_{H_i \in i, H_j \in j} \varphi(X_{H_i H_j}) \tag{5}$$

and $\sum_j \alpha_{ij} = 1$. Also, a potential function $\varphi(X_{H_i H_j})$ is defined by $\varphi(X_{H_i H_j}) = e^{-D(H_i, H_j)}$ where $D(H_i, H_j)$ denotes the absolute difference between two histogram features. The 3×3 matrix of state-transition probabilities is generated in our framework by applying Eq. 5 on the training data.

3) VITERBI DETECTION

DST-framework applies the Viterbi algorithm to find an optimal path within the state diagram shown in Fig. 2. Initial probability in state i is given by $\mathbf{\Pi} = [\Pi_i]$ where $\Pi_i = Pr(q_i|t = 1)$ and a path q_i starts from state 1. Given an input motion feature sequence I, a path q, and DST-framework parameters $\{A, B, \mathbf{\Pi}\}$, a normalized accumulated probability along the optimal path is defined by Eq. 6. In Eq. 6, A and B indicate the state transition and state probabilities, respectively.

$$\underset{\forall\ path\ q}{arg\ max}\{-\frac{log(Pr(I, q|A, B, \mathbf{\Pi}))}{T}\} \tag{6}$$

where a joint probability $Pr(I, q|A, B, \mathbf{\Pi})$ is as follows:

$$Pr(I, q|A, B, \mathbf{\Pi}) = Pr(I|A, B, \mathbf{\Pi}) \cdot Pr(q|A, B, \mathbf{\Pi}) \tag{7}$$

Each DST-framework model calculates the probability of maximum likelihood path from the incoming sleep motion data.

The final decision of *SP* motion models is determined by selecting one of DST-framework models with the highest accumulated probability as shown in Eq. 8.

$$output = \underset{\eta}{arg\ max}\{DST_\eta\} \tag{8}$$

where η is the class of *SP* motion models.

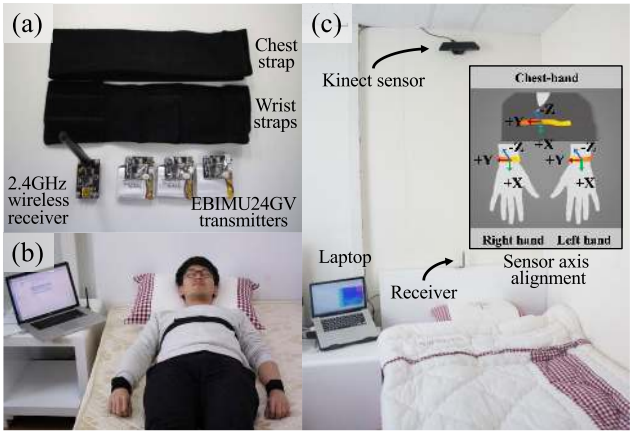

FIGURE 4. System setup: (a) Sensor set (b) A participant wearing sensor sets (c) Sleep experimental room.

C. WEARABLE SLEEP POSITION TRACKING SYSTEM

In this section, we describe the configuration of wSPOTS, pre-processing, and a two-level classifier. The two-level classifier consists of the first classifier (C^1) and the second classifier (C^2).

1) SYSTEM SETUP

Fig. 4 shows the system setup. The hardware of wSPOTS consists of three wearable devices, i.e., two for wrists and one for chest, as shown in Fig. 4(a). The two wrist-bands and one chest-band use the same 9-axis IMU sensor containing 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer sensors for collecting sleep data. The IMU sensor in our system is an AHRS EBIMU24GV module with a sampling rate of 100Hz from E2BOX Company in Korea, an accelerometer sensor range of 2g, and a gyroscope range of 2000dps. The sensed data from IMU sensors are sent to a Laptop through 2.4GHz wireless communication.

A participant wears three IMU sensors that are placed according to a pre-defined orientation such that a y-axis in the sensor coordinate system points to the fingertip while z-axis points the downward direction in the back of a hand. The chest sensor is also aligned such that its y- and z-axis points to legs and the downward of the chest, respectively, as shown in Fig. 4(b)(c). With these wearable devices, we evaluate the performance of wSPOTS by analyzing datasets from both pilot and onsite experiments.

The pilot experiment is an experiment that repeats pre-classified *SP* motions with a fixed interval time in a controlled environment. We recruited five participants and asked them to perform the pre-classified *SP* motion slightly differently at a fixed time interval about 100 times.

The onsite experiment is an actual night sleep experiment. We have built a sleep test-bed in our lab and collected night-time sleep data for a total of eleven participants. All participants were composed of young men in their 20s. An infrared camera sensor (Kinect) was installed on top of the sleep test-bed for ground-truth labeling as shown in Fig. 4(c), and all

SP motions in the onsite experiments were categorized by manually identifying the sleep position change in the video recording of the infrared camera sensor. Note that sleep experiments of the participants were approved by the Institutional Review Board (IRB) at DGIST.

2) PRE-PROCESSING

In this section, we will explain how to handle pre-processing procedures: motion detection, motion grouping, and generation of directional features.

First, *motion detection* is used to find activity periods to analyze human activities. The activity periods are useful to eliminate cumulative errors using a method called a zero velocity update in navigation and tracking applications [39]. To consider both the acceleration and the angular rate in human motion, we consider the magnitude of both the accelerometer and the gyroscope sensor. Also, we apply a high pass filter to the accelerometer sensor $\mathbf{A}^{hpf}$ to consider only the change of the accelerometer sensor without a gravity component. The magnitude of two sensors, $\mathbf{m}^A$ and $\mathbf{m}^G$, are defined with Eq. 9 and Eq. 10.

$$\mathbf{m}^A = (\mathbf{A}_x^{hpf})^2 + (\mathbf{A}_y^{hpf})^2 + (\mathbf{A}_z^{hpf})^2 \tag{9}$$

$$\mathbf{m}^G = (\mathbf{G}_x)^2 + (\mathbf{G}_y)^2 + (\mathbf{G}_z)^2 \tag{10}$$

Thereafter, we normalize the two magnitude values and define the sum of the two normalized values as $\mathbf{V}_i$ with Eq. 11. We define the value of $\mathbf{V}_i$ at an instance i as a stationary if it is smaller than a heuristic threshold γ, and as a non-stationary otherwise. For the heuristic threshold γ, we extract a *SP* log from three wearable sensors by changing the threshold. And then, we select an optimal threshold that the *SP* log from the wearable sensors and a *SP* log from Kinect camera (ground truth) are the most consistent.

$$\mathbf{V}_i = \frac{\mathbf{m}^A(i)}{max(\mathbf{m}^A)} + \frac{\mathbf{m}^G(i)}{max(\mathbf{m}^G)} \tag{11}$$

By using the timestamps of the non-stationary period, we confirm when and where the motion has occurred. The *motion detection* uses all three devices to compare and analyze the performance of the proposed algorithm, i.e., DST-framework. Note that *SP* tracking performance would be decreased in the case of fewer than three devices due to missing of detecting sleep motions.

Second, we use a *motion grouping*. It is designed to consider adjacent motions that they are scattered but correlated to the sleep position change. The *motion grouping* connects adjacent motions as one motion within a pre-defined time by observing timestamps. Adjacent motions are grouped into one motion if the motions occur within the pre-defined time φ between them as shown in Alg. 1, where we set φ to one minute.

Lastly, *directional features*, i.e., GV, are extracted using orientation of the IMU sensor. We generate the orientation, quaternion q, by an accelerometer/gyro sensor fusion algorithm, i.e., Madgwick AHRS algorithm [40]. Thereafter,

Algorithm 1 Motion Grouping

Result: Filtered motion timestamps in a queue Q
$\tau, \eta, \varphi \leftarrow$ Timestamp, temp variable, and threshold
$Q_{start}, Q_{end} \leftarrow$ queues for start timestamp, and end timestamp
$\varphi = 1$ minute // Initialization process
$\eta_{start} = \tau_{start}(m_1)$
$\eta_{end} = \tau_{end}(m_1)$
$Q_{start} = \text{EnQueue}(\eta_{start})$
while *Incoming new Motion* m_i **do**
  $\zeta = \tau_{start}(m_i) - \eta_{end}$
  if $\zeta < \varphi$ **then**
    $\eta_{end} = \tau_{end}(m_i)$
  else
    $Q_{end} = \text{EnQueue}(\eta_{end})$
    $\eta_{start} = \tau_{start}(m_i)$
    $Q_{start} = \text{EnQueue}(\eta_{start})$
  end
end

directional features are calculated by quaternion rotation mathematics.

According to [41], the current IMU sensor orientation, quaternion q, is defined by three imaginary numbers and one real number as

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k \tag{12}$$

The GV is calculated by applying to the current IMU orientation $\mathbf{q}$ by tacking the conjugation of $\mathbf{r}$ by $\mathbf{q}$

$$r' = \mathbf{q} \cdot r \cdot \mathbf{q}^* \tag{13}$$

where $\mathbf{r}$ is a unit vector [0, 0, 1] representing the global z-axis, and r' representing GV is a rotated vector in the global axes. The rotated vector r' lies on a unit sphere. We compute GV for every sensor reading, and then apply to the next process.

The direction of sleep motions is influenced by the orientation of bed, and sometimes one sleeps in an arbitrary direction different from the bed direction owing to sleeping habits. This causes the direction of sleep motion feature to deviate from the predefined one. To address this challenge, we transform the quaternion data into a GV that represents a z-axis (gravity force) component of the sensor to extract an orientation-invariant sleep motion feature while losing horizontal (i.e., x- and y-axes) information of the sensors. GV is used as directional features in our system. Note that gravity pulls everything towards the center of the earth and is captured by 3-axis accelerometer sensors with $1g$ gravity in a static condition.

3) FIRST CLASSIFIER

The directional features of *Up* sleep position occupy most of a unit sphere due to the high degree of freedom. By considering this unevenly distributed feature characteristic, we divide all *SP* motions into two groups (*SPC* group and *NonSPC* group)

TABLE 1. Feature design for the first classifier C^1.

Feature Name	Type	# Feature	Component	# Total
Mean	Time-domain	1	(GV_x, GV_y, GV_z)	3
Root mean square	Time-domain	1	(GV_x, GV_y, GV_z)	3
Covariance	Time-domain	3	(GV_x, GV_y, GV_z)	9
Spectral peaks	Frequency-domain	12	(GV_x, GV_y, GV_z)	36
Spectral power	Frequency-domain	5	(GV_x, GV_y, GV_z)	15

TABLE 2. DST-framework model components in *SPC* group.

State 1 (Stationary)	State 2 (Non-stationary)	State 3 (Stationary)
Up	Up→Left ‖ Left→Up	Up
Left	Up→Right ‖ Right→Up	Left
Right	Down→Left ‖ Left→Down	Right
Down	Down→Right ‖ Right→Down	Down

and classify *SP* motions separately in each group. For example, *Up-to-Up* motion in *SPC* group tends to have a higher accumulated probability than *Up-to-Left* motion in *NonSPC* group due to the feature distribution.

The first classifier C^1 plays the role of classifying *SPC* group and *NonSPC* group. The two groups also show imbalanced datasets since raw sleep datasets are imbalanced. We adopt a Random Under Sampling Boost (RUSBoost) algorithm since it is one of the effective ML algorithms for classification of imbalanced datasets [42], [43]. The logic behind the RUSBoost algorithm is undersampling by taking the minimum sample number N in every class and taking an optimal subset of data with N in each class through an adaptive boosting.

For feature extraction, we use a 3-axis GV and generate a total of 66 features in the time-domain and frequency-domain as shown in Tab. 1. These features are used for human physical activity recognition [44], [45]. Performance analysis for this first classifier C^1 will be covered in detail in Section IV-C.1. Note that all programming and analysis were done by using MATLAB.

4) SECOND CLASSIFIER

After classifying *SPC* and *NonSPC* group, wSPOTS performs detailed *SP* motion classification in a second classifier C^2 according to the group. Eight *SP* motions are in the *SPC* group while four *SP* motions are in the *NonSPC* group.

For classification of twelve *SP* motions, wSPOTS generates DST-framework models belonging to each group (SPC or NonSPC) by using components for each state as shown in Table 2 and Table 3. For example, *Up* to *Left* motion model is made by a combination of features corresponding to *Up* in State 1, $Up \rightarrow Left || Left \rightarrow Up$ in State 2, and *Left* in State 3. The $Up \rightarrow Left$ and $Left \rightarrow Up$ are considered as the same directional features since they have the same directional features, i.e., GV, when considering only the directional components. Detailed state transition probability, state probability, and the final decision for DST-framework are described in Section III-B above.

We use SVM and RF as conventional ML algorithms in the second classifier C^2. 66 features in the first classifier

TABLE 3. DST-framework model components in *NonSPC* group.

State 1 (Stationary)	State 2 (Non-stationary)	State 3 (Stationary)
Up	Up→Up	Up
Left	Left→Left	Left
Right	Right→Right	Right
Down	Down→Down	Down

C^1 are used equally for the two algorithms as input features in the second classifier C^2. Besides, we also evaluate ST-framework that uses a left-to-right model. The ST-framework is also a newly proposed algorithm, but DST-framework is a more enhanced framework tailored *SP* motions by using conditioned fully connected model.

IV. EVALUATION

In this section, we describe the datasets that we collected for system performance evaluation of wSPOTS. We first evaluate two classifiers of wSPOTS separately in the algorithm-level evaluation. In addition, we evaluate the performance of wSPOTS considering the two classifiers together for the system-level evaluation.

A. EVALUATION DATASETS

We evaluate the system performance of wSPOTS through two experiments: Pilot and Onsite experiment.

The pilot experiment means to perform specific *SP* motion in a controlled environment. Through this experiment, we can make each *SP* motion models based on DST-framework and evaluate the classification performance experimentally. Participants are required to move randomly about a hundred times in each *SP* motion to obtain data under the controlled conditions. We recruited five participants and collected a total of 6,256 balanced datasets.

The onsite experiment refers to actual overnight sleep data. This experiment is designed to confirm the classification performance of the proposed algorithm in *SP* motions of actual sleep. We recruited eleven participants, and a total of 516 imbalanced datasets were collected. At this time, the ground truth label for each *SP* motion is labeled manually by observing the video record of an infrared camera (Kinect sensor). A detailed description of collected datasets for the two experiments is shown in Table 4.

B. EVALUATION METRICS

For classification performance evaluation, a confusion matrix is widely used. The confusion matrix sorts all cases results from the classification model into categories by determining whether the predicted value matches the actual value. In the confusion matrix, we can compute basic metrics such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). In our system, we use two performance metrics by using the confusion matrix and basic metrics: overall *Accuracy* and average *F1-score*.

Overall *Accuracy* is a good performance metric for overall classification performance when the entire data set is

TABLE 4. Datasets description in Pilot experiment and Onsite experiment.

Motion type	# Pilot experiment	# Onsite experiment
SPC group	**4,096**	**80**
1) Left→Up	554	16
2) Up→Left	562	16
3) Right→Up	567	21
4) Up→Right	569	25
5) Down→Left	452	0
6) Left→Down	451	0
7) Down→Right	471	1
8) Right→Down	470	1
NonSPC group	**2,160**	**436**
9) Up→Up	547	374
10) Left→Left	487	33
11) Right→Right	572	29
12) Down→Down	554	0

nearly balanced. This metric is calculated as the total number of correctly classified items divided by the total number of test items in a confusion matrix M as shown in Eq. 15. However, actual sleep motion data shows imbalanced datasets, so it is not enough to evaluate the classification performance with only the overall *Accuracy*.

Therefore, we also use another performance metric, i.e., *F1-score*, which is an effective metric for the imbalanced datasets [46]. *F1-score* is a harmonic average of precision and recall with Eq. 17. Precision and Recall have different purposes in system performance evaluation. Precision with Eq. 15 seeks to reduce FP while Recall with Eq. 16 tries to reduce FN. We calculate F1-score for every class and use the average overall classes, i.e., average *F1-score*, as the performance metric with Eq. 18, where c and m denote class and the number of class, respectively.

$$Overall\ Accuracy = \frac{\sum M_{i,j}(x|i == j)}{\sum M_{i,j}(x)} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1score = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{17}$$

$$Average\ F1score = \frac{1}{m}\sum_{c=1}^{m} F1score(c) \tag{18}$$

C. ALGORITHM-LEVEL EVALUATION

In this section, we evaluate the classification performance of two classifiers at each level according to both pilot and onsite experiments. In addition, we evaluate the performance by considering combinations of three wearable devices.

1) FIRST CLASSIFIER EVALUATION

The first process of wSPOTS is to classify motion groups: *SPC* and *NonSPC*. This is called a first classifier C^1. We use RUSBoost algorithm for C^1. Note that RUSBoost algorithm is selected as a simple filter. Comparing with other algorithms is beyond the scope of this work. We focus on the second classifier C^2 based on DST-framework.

TABLE 5. First classifier (C^1) evaluation by using RUSBoost algorithm.

Combination	*Pilot experiment*		*Onsite experiment*	
	Accuracy	*F1-score*	*Accuracy*	*F1-score*
L-band	0.86	0.84	0.82	0.70
R-band	0.86	0.84	0.83	0.72
C-band	1.00	1.00	0.96	0.93
LC-band	0.99	0.99	0.96	0.92
RC-band	0.99	0.99	0.95	0.90
LR-band	0.88	0.86	0.84	0.74
LRC-band	0.99	0.99	0.96	0.92

Accuracy: overall Accuracy, *F1-score*: average F1-score
Train/Test dataset in *Pilot exp.*: Pilot with 5-fold cross-validation
Train/Test dataset in *Onsite exp.*: Onsite with 5-fold cross-validation

We have labeled datasets for *SPC* group and *NonSPC* group as shown in Tab. 4. By using these labeled datasets, we conduct 5-fold cross validation in each experiment. Table 5 shows the classification performance of *SPC* and *NonSPC* according to a total of seven combinations of three devices in a pilot experiment and an onsite experiment.

In the pilot experiment, all combinations show good performance with higher than overall Accuracy of 0.86 and average F1-score of 0.84. Combinations using only wristbands such as *L*-band, *R*-band, and *LR*-band show relatively low performance compared with other combinations. There is no significant difference when comparing overall Accuracy and average F1-score since the datasets of the pilot experiment are balanced datasets.

In the onsite experiment, classification performance shows a tendency similar to the pilot experiment, but the overall performance is slightly decreased due to sensing noise caused by sleep habits. Although the overall Accuracy represents the overall classification performance, the average F1-score is more effective performance metric for imbalanced datasets [46]. Combinations including a chest-band such as *C*-band, *LC*-band, *RC*-band, and *LRC*-band, show good performance with higher than overall Accuracy of 0.95 and average F1-score of 0.92.

Experimental results of the first classifier C^1 demonstrate that four combinations including a chest-band achieve high performance in both pilot and onsite experiments. A combination using only a chest-band, i.e., *C*-band, shows the best performance in the first classifier C^1. It is because many features from multiple sensors may not be effective in RUSBoost. However, the *C*-band shows relatively low performance than the other combinations in a second classifier C^2. We will discuss a detailed performance evaluation for the second classifier C^2 in the next section.

2) SECOND CLASSIFIER EVALUATION

Second classifier C^2 performs detailed *SP* motion classification depending on *SPC* and *NonSPC*. To evaluate the second classifier C^2, we use six algorithms: SVM, RF, ST-SVM, ST-RF, DST-SVM, and DST-RF. To train and test *SP* motion models with multiple algorithms, we conduct 5-fold cross-validation in the pilot experiment. To evaluate the second classifier C^2 using onsite datasets, we use the datasets

TABLE 6. Algorithm-level evaluation: Overall *Accuracy* of second classifier (C^2) considering multi-level approach.

Experiment	Algorithm	*L*-band		*R*-band		*C*-band		*LC*-band		*RC*-band		*LR*-band		*LRC*-band	
		1-Lv	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*
Pilot experiment	SVM	0.61	0.69	0.58	0.67	0.96	0.97	0.96	0.97	0.97	0.98	0.73	0.81	0.98	0.98
	RF	0.67	0.77	0.65	0.75	0.99	0.99	0.99	0.99	0.99	0.99	0.78	0.85	0.99	0.99
	ST-SVM	0.39	0.60	0.40	0.60	0.76	0.90	0.68	0.97	0.65	0.98	0.58	0.78	0.67	0.98
	ST-RF	0.39	0.60	0.41	0.60	0.77	0.92	0.63	0.97	0.65	0.98	0.56	0.79	0.65	0.99
	DST-SVM	0.33	0.53	0.35	0.55	0.75	0.90	0.68	0.97	0.71	0.97	0.55	0.79	0.70	0.99
	DST-RF	0.32	0.58	0.35	0.59	0.76	0.88	0.64	0.95	0.69	0.96	0.54	0.80	0.69	0.99
Onsite experiment	SVM	0.10	0.32	0.09	0.32	0.62	0.71	0.57	0.67	0.47	0.72	0.07	0.44	0.36	0.71
	RF	0.15	0.57	0.14	0.48	0.85	0.96	0.78	0.98	0.79	0.96	0.18	0.69	0.69	0.98
	ST-SVM	0.10	0.49	0.09	0.44	0.13	0.87	0.13	0.90	0.14	0.94	0.09	0.60	0.14	0.94
	ST-RF	0.10	0.52	0.10	0.42	0.13	0.96	0.14	0.97	0.14	0.97	0.09	0.64	0.14	0.98
	DST-SVM	0.09	0.53	0.08	0.47	0.18	0.88	0.14	0.92	0.23	0.95	0.10	0.63	0.15	0.95
	DST-RF	0.06	0.56	0.09	0.49	0.16	0.96	0.15	0.97	0.20	0.98	0.09	0.66	0.16	0.99

1-Lv: 1-Level Classifier, *2-Lv*: 2-Level Classifier
Train/Test dataset in *Pilot experiment*: Pilot with 5-fold cross-validation
Train/Test dataset in *Onsite experiment*: Pilot/Onsite

TABLE 7. Algorithm-level evaluation: Average *F1-score* of second classifier (C^2) considering multi-level approach.

Experiment	Algorithm	*L*-band		*R*-band		*C*-band		*LC*-band		*RC*-band		*LR*-band		*LRC*-band	
		1-Lv	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*	*1-Lv*	*2-Lv*
Pilot experiment	SVM	0.60	0.69	0.58	0.67	0.96	0.96	0.96	0.97	0.97	0.98	0.72	0.80	0.97	0.98
	RF	0.69	0.76	0.64	0.75	0.99	0.99	0.99	0.99	0.99	0.99	0.78	0.85	0.99	0.99
	ST-SVM	0.36	0.59	0.38	0.59	0.70	0.90	0.61	0.96	0.57	0.98	0.54	0.78	0.58	0.98
	ST-RF	0.37	0.59	0.39	0.60	0.70	0.91	0.54	0.97	0.57	0.98	0.52	0.79	0.55	0.99
	DST-SVM	0.30	0.51	0.32	0.54	0.69	0.89	0.63	0.97	0.64	0.97	0.51	0.78	0.64	0.99
	DST-RF	0.28	0.56	0.32	0.58	0.69	0.87	0.58	0.95	0.61	0.95	0.51	0.80	0.63	0.98
Onsite experiment	SVM	0.09	0.20	0.09	0.21	0.43	0.62	0.28	0.45	0.28	0.48	0.07	0.22	0.23	0.51
	RF	0.13	0.25	0.11	0.23	0.58	0.66	0.43	0.72	0.50	0.54	0.16	0.39	0.49	0.64
	ST-SVM	0.10	0.29	0.05	0.24	0.26	0.65	0.30	0.71	0.34	0.73	0.07	0.34	0.33	0.66
	ST-RF	0.08	0.29	0.07	0.23	0.29	0.70	0.28	0.87	0.38	0.90	0.08	0.39	0.38	0.91
	DST-SVM	0.10	0.27	0.07	0.27	0.36	0.65	0.33	0.73	0.53	0.79	0.09	0.40	0.36	0.79
	DST-RF	0.05	0.31	0.08	0.28	0.50	0.82	0.32	0.87	0.62	0.91	0.10	0.40	0.43	0.93

1-Lv: 1-Level Classifier, *2-Lv*: 2-Level Classifier
Train/Test dataset in *Pilot exp.*: Pilot with 5-fold cross-validation
Train/Test dataset in *Onsite exp.*: Pilot/Onsite

collected in the pilot experiment as a training dataset to train each *SP* motion, and use the datasets of the onsite experiment as a test dataset.

Table 6 and Table 7 show detailed classification performance evaluation of the six algorithms in the two experiments. In this evaluation, we take three key aspects into consideration: a design of two-level classifier, both pilot and onsite experiments, and combinations of three wearable devices.

First, the evaluation results demonstrate that two-level classifier as designed achieves better classification performance than one-level classifier. The one-level classifier directly classifies twelve *SP* motions without a filtering process for *SPC* and *NonSPC*. All algorithms in both pilot and onsite experiments show consistent results, demonstrating that our design of two-level classifier is effective in classification performance.

Second, we compare the performance of six algorithms in the two experiments. The proposed algorithms based on the DST-framework in the pilot experiment show acceptable performance, but conventional ML algorithms such as SVM and RF show slightly better performance. The pilot experiment collects *SP* motions in a controlled motion pattern with a fixed time interval, which generates regularized static patterns. The ML algorithms effectively classify these static patterns. However, the ML algorithms show significant performance degradation in the onsite experiment. The actual sleep data include many sensing noises caused by sleep habits and time-varying data, resulting in motion patterns different from those of the pilot experiment. The proposed algorithms show more robust and accurate performance in classifying actual sleep motions.

Lastly, we evaluate the combinations of wearable devices. Obviously, with more sensors, the classification performance improves. In order to achieve high classification performance, using only one chest-band is acceptable but it shows low performance compared to combinations of one chest-band and at least one wristband.

Fig. 5 shows a summary plot of the experimental results. Experimental results demonstrate that algorithms based on DST-framework performs well in the pilot experiment and outperforms the other algorithms in the onsite experiment. Specifically, DST-RF algorithm provides the most robust and accurate classification performance when considering two experiments and seven combinations of three wearable devices. In the evaluation of actual sleep data, we confirm that RF is better than SVM, ST-framework is better than RF, and DST-framework is slightly better than ST-framework.

The results indicate that conventional ML algorithms (SVM and RF) perform effectively in the proposed frameworks, and we achieve even further performance improvement by modifying the network model, i.e., DST-framework. We select the DST-RF algorithm for C^2 in wSPOTS.

D. SYSTEM-LEVEL EVALUATION

wSPOTS is a two-level classifier consisting of a first classifier C^1 and a second classifier C^2 at each level for classifying twelve *SP* motions. According to the decision of the first classifier C^1, the second classifier C^2 based on DST-framework performs detailed *SP* motion classification. In the algorithm-level evaluation, two classifiers are evaluated independently, and the optimal algorithm in each classifier is selected. On the other hand, the system-level evaluation implies that two classifiers are considered together. That is, the prediction results from the first classifier C^1 influences on the second classifier C^2. We use RUSBoost for the first classifier C^1 and DST-RF for the second classifier C^2, and the results of the system-level evaluation are shown in Table 8.

In the pilot experiment, classification performance results show performance degradation about 0∼9% and 0∼10% in overall Accuracy and average F1-score, respectively, due to false prediction results from the first classifier C^1. There is no big difference between two performance metrics and show high performance over 0.88 in overall Accuracy and 0.87 in average F1-score, except for combinations that only use wristbands.

On the other hand, the onsite experiment shows even worse performance degradation than the pilot experiment, decreasing about 3∼10% and 0∼18% in overall Accuracy and average F1-score, respectively. As discussed earlier, the datasets of the onsite experiment are imbalanced datasets. The datasets cause significant performance degradation if the second classifier C^2 leads to false prediction in the minority class. Combinations having one chest-band and at least one wristband show acceptable performance over 0.94 in overall Accuracy and 0.79 in average F1-score.

The evaluation results indicate that for practical and accurate monitoring applications, one chest-band is possible, but combinations consisting of one chest-band and at least one wristband are more desired.

TABLE 8. System-level evaluation using RUSBoost (C^1) and DST-RF (C^2).

Experiment	Combination	RUSBoost (C^1) + DST-RF (C^2)	
		Accuracy	*F1-score*
Pilot experiment	*L*-band	0.50	0.49
	R-band	0.51	0.51
	C-band	0.88	0.87
	LC-band	0.95	0.94
	RC-band	0.95	0.95
	LR-band	0.71	0.70
	LRC-band	0.98	0.98
Onsite experiment	*L*-band	0.47	0.25
	R-band	0.41	0.18
	C-band	0.92	0.64
	LC-band	0.94	0.87
	RC-band	0.94	0.79
	LR-band	0.56	0.31
	LRC-band	0.95	0.91

Accuracy: overall Accuracy, *F1-score*: average F1-score
Train/Test dataset in *Pilot exp.*: Pilot with 5-fold C.V. for C^1 and C^2
Train/Test dataset in *Onsite exp.*: Onsite with 5-fold C.V. for C^1 and Pilot/Onsite for C^2, C.V.: Cross-Validation

V. DISCUSSION

wSPOTS aims to classify *SP* motions from the motion noise during sleep caused by the unconsciousness. Based on our observations on actual sleep motions, we note there exist so many noises such as scratching body and stretching. Moreover, most motions related to *SP* change include motion noise. It makes it challenging to generate regulated motion patterns for motion recognition, which affects tracking system performance. To address this real-realm tracking issue, we designed DST-framework, which is more robust to sensing noise. We demonstrate the effectiveness of the DST-framework in the evaluation section. Further, we evaluated combinations of wearable devices for finding the best solution in *SP* tracking problem.

There are some limitations in this work. First, motion noise can be caused by the loosening of a device during sleep. SmartMove [47] tested the mount conditions such as tightly mounted and loosely mounted. The loosely mounted device showed a high variation in sensing values than that of the tightly mounted device. We also observed that some participants' mount position is a little bit moved due to roll over during sleep. This loosely wearing condition might be one of the factors leading to poor performance in using a single device during sleep.

Second, unfortunately, there are missing data in the onsite experiment due to characteristics of imbalanced sleep motions, e.g., Down-to-Down motion. Thus, there are limitations verifying the classification performance of all *SP* motion classes in detail. To overcome the limitation, we train DST-framework based on a model-based approach with pilot experiment dataset and test with onsite experiment dataset for evaluation using actual sleep data. In this way, we can claim the overall superiority of the proposed DST-framework by comparing ML algorithms and device combinations.

Lastly, wSPOTS does not consider real-time issues. Although we developed a motion-based *SP* motion classification, our system gives passive feedback that displays the user's *SP* log after the user wakes up.

For future work, we plan to extend the current work in several directions for a comprehensive sleep care system. First, with the advance of deep learning techniques, it is possible to study more specialized classifier and personalization. Current wSPOTS is based on a model-based *SP* classifier. We will conduct data-driven research related to personalization based on big data from many users. Besides, sleep characteristics according to age, gender, and sleep disorders with deep learning will also be analyzed for the personalization study.

Second, eXplainable Artificial Intelligence (XAI) is essential in medical applications [48]. Successful AI systems are usually applied in a black-box manner, but they should be interpreted and verified by medical experts. In actual

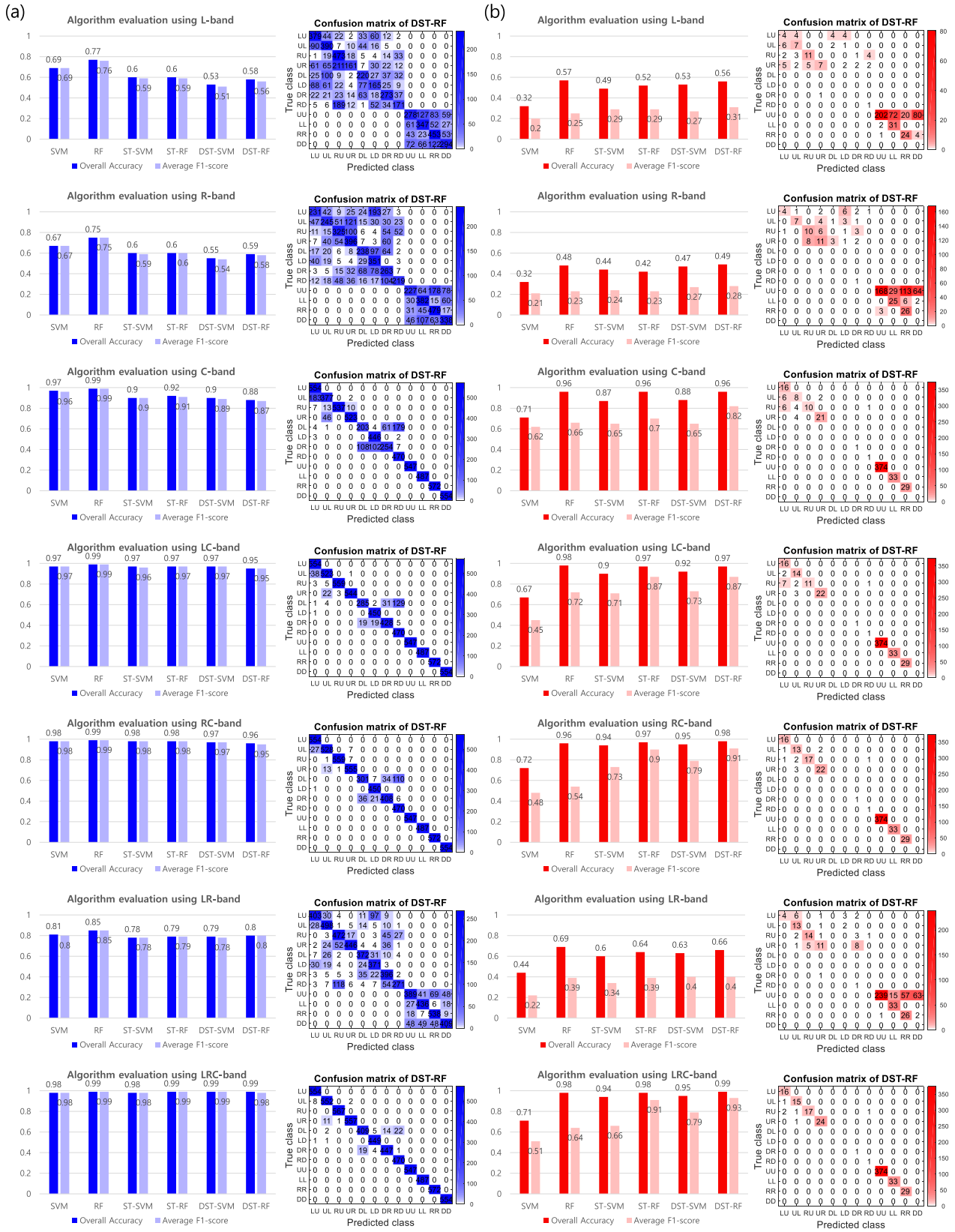

FIGURE 5. Second classifier (C^2) performance evaluation depending on seven combinations of three devices: (a) Pilot experiment, (b) Onsite experiment.

sleep data, we observed much noise that resulted from sleep behaviors, such as scratching body, stretching, hand moving, arm raising, and body trembling. In addition to tracking sleep positions, we will research the automatic classification of these microscopic sleep behaviors for XAI in sleep applications.

Lastly, Human-in-the-Loop Cyber-Physical Systems (HiLCPS) considers human as a component in the loop and gives appropriate actions [49]. Many off-the-shelf sleep trackers just passively give sleep information to the user. However, a system that gives active feedback to the user can further improve sleep management. We have evaluated the reaction of users by using chest band-based vibration feedback system in real-time, which is called SleepManager [50]. SleepManager uses a simple threshold to classify two sleep positions and gives vibration feedback to the user when it detects a supine position. We will further investigate issues related to real-time constraints and user adaptability of feedback in HiLCPS.

VI. CONCLUSION

Tracking sleep positions is challenging since unconscious sleep behavior is neither consistent nor predictable. To address this challenge, we propose a two-level classifier specialized for twelve *SP* motions based on DST-framework. The proposed wSPOTS, together with a spatio-temporal classification algorithm, i.e., DST-framework, aims to be used as an effective tracking system in positional sleep treatment.

Experimental results demonstrate that proposed algorithms based on DST-framework in wSPOTS show good classification performance in a pilot experiment and outperforms other conventional ML algorithms such as SVM and RF in an onsite experiment. In addition, we evaluate seven combinations of three wearable devices to check the suitability for practical and effective sleep monitoring applications.

The results support that wSPOTS is capable of tracking sleep positions effectively. We expect our system to be one of the key building blocks for comprehensive sleep care.

REFERENCES

[1] A. Sasidharan, S. Sulekha, and B. Kutty, "Current understanding on the neurobiology of sleep and wakefulness," *Int. J. Clin. Exp. Physiol.*, vol. 1, no. 1, p. 3, 2014.

[2] K. E. Bloch, "Polysomnography: A systematic review," *Technol. Health Care*, vol. 5, no. 4, pp. 285–305, 1997.

[3] M. Hirshkowitz, K. Whiton, S. M. Albert, C. Alessi, O. Bruni, L. DonCarlos, N. Hazen, J. Herman, E. S. Katz, L. Kheirandish-Gozal, N. Hazen, J. Herman, E. S. Katz, L. Kheirandish-Gozal, D. N. Neubauer, A. E. O'Donnell, M. Ohayon, J. Peever, R. Rawding, R. C. Sachdeva, B. Setters, M. V. Vitiello, J. C. Ware, and P. J. A. Hillard, "National Sleep Foundationâ Źs sleep time duration recommendations: Methodology and results summary," *Sleep Health*, vol. 1, no. 1, pp. 40–43, 2015.

[4] M. Ohayon, E. M. Wickwire, M. Hirshkowitz, S. M. Albert, A. Avidan, F. J. Daly, Y. Dauvilliers, R. Ferri, C. Fung, D. Gozal, N. Hazen, A. Krystal, K. Lichstein, M. Mallampalli, G. Plazzi, R. Rawding, F. A. Scheer, V. Somers, and M. V. Vitiello, "National sleep foundation's sleep quality recommendations: First report," *Sleep Health*, vol. 3, no. 1, pp. 6–19, 2017.

[5] M. J. Sateia, "International classification of sleep disorders-third edition," *Chest*, vol. 146, no. 5, pp. 1387–1394, 2014.

[6] S. A. Joosten, D. M. O'Driscoll, P. J. Berger, and G. S. Hamilton, "Supine position related obstructive sleep apnea in adults: Pathogenesis and treatment," *Sleep Med. Rev.*, vol. 18, no. 1, pp. 7–17, 2014.

[7] E. A. Mitchell, B. T. Thach, J. M. Thompson, and S. Williams, "Changing Infants'Sleep position increases risk of sudden infant death syndrome," *Arch. Pediatrics Adolescent Med.*, vol. 153, no. 11, pp. 1136–1141, 1999.

[8] Z. Moore, S. Cowman, and R. M. Conroy, "A randomised controlled clinical trial of repositioning, using the 30° tilt, for the prevention of pressure ulcers," *J. Clin. Nursing*, vol. 20, nos. 17–18, pp. 2633–2644, 2011.

[9] W. Richard, D. Kox, C. den Herder, M. Laman, H. van Tinteren, and N. de Vries, "The role of sleep position in obstructive sleep apnea syndrome," *Eur. Arch. Oto-Rhino-Laryngol. Head Neck*, vol. 263, no. 10, pp. 946–950, 2006.

[10] C. C. Hsia, K. J. Liou, A. P. W. Aung, V. Foo, W. Huang, and J. Biswas, "Analysis and comparison of sleeping posture classification methods using pressure sensitive bed system," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2009, pp. 6131–6134.

[11] J. J. Liu, Q. Wu, M.-C. Huang, N. Alshurafa, M. Sarrafzadeh, N. Raut, and B. Yadegar, "Sleep posture analysis using a dense pressure sensitive bedsheet," *Pervas. Mobile Comput.*, vol. 10, pp. 34–50, Feb. 2014.

[12] Y. Enokibori and K. Mase, "Data augmentation to build high performance DNN for in-bed posture classification," *J. Inf. Process.*, vol. 26, pp. 718–727, Oct. 2018.

[13] M. Donati, F. Cecchi, F. Bonaccorso, M. Branciforte, P. Dario, and N. A. Vitiello, "A modular sensorized mat for monitoring infant posture," *Sensors*, vol. 14, no. 1, pp. 510–531, 2013.

[14] M. B. Pouyan, S. Ostadabbas, M. Farshbaf, R. Yousefi, M. Nourani, and M. D. M. Pompeo, "Continuous eight-posture classification for bedbound patients," in *Proc. 6th Int. Conf. Biomed. Eng. Inform.*, Dec. 2013, pp. 121–126.

[15] M. B. Pouyan, J. Birjandtalab, M. Heydarzadeh, M. Nourani, and S. A. Ostadabbas, "A pressure map dataset for posture and subject analytics," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Inform. (BHI)*, Feb. 2017, pp. 65–68.

[16] S. Lokavee, N. Watthanawisuth, J. P. Mensing, and T. Kerdcharoen, "Sensor pillow system: Monitoring cardio-respiratory and posture movements during sleep," in *Proc. 4th Biomed. Eng. Int. Conf.*, Jan. 2012, pp. 71–75.

[17] E. Hoque, R. F. Dickerson, and J. A. Stankovic, "Monitoring body positions and movements during sleep using WISPs," in *Proc. Wireless Health*, 2010, pp. 44–53.

[18] J. Park, W. Nam, J. Choi, T. Kim, D. Yoon, S. Lee, J. Paek, and J. Ko, "Glasses for the third eye: Improving the quality of clinical data analysis with motion sensor-based data filtering," in *Proc. 15th Conf. Embedded Netw. Sensor Syst.*, 2017, p. 8.

[19] X. Liu, J. Cao, S. Tang, and J. Wen, "Wi-sleep: Contactless sleep monitoring via WiFi signals," in *Proc. IEEE Real-Time Syst. Symp.*, Dec. 2014, pp. 346–355.

[20] J. Liu, X. Chen, S. Chen, X. Liu, Y. Wang, and L. Chen, "TagSheet: Sleeping posture recognition with an unobtrusive passive tag matrix," in *Proc. IEEE Conf. Comput. Commun.*, Apr./May 2019, pp. 874–882.

[21] C.-W. Wang, A. Hunter, N. Gravill, and S. Matusiewicz, "Real time pose recognition of covered human for diagnosis of sleep apnoea," *Computerized Med. Imag. Graph.*, vol. 34, no. 6, pp. 523–533, 2010.

[22] A. Ren, B. Dong, X. Lv, T. Zhu, F. Hu, and X. Yang, "A non-contact sleep posture sensing strategy considering three dimensional human body models," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 414–417.

[23] M. S. D. Rasouli and S. Payandeh, "A novel depth image analysis for sleep posture estimation," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 5, pp. 1999–2014, 2019.

[24] T. Grimm, M. Martinez, A. Benz, and R. Stiefelhagen, "Sleep position classification from a depth camera using bed aligned maps," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 319–324.

[25] K.-M. Chang and S.-H. A. Liu, "Wireless portable electrocardiogram and a tri-axis accelerometer implementation and application on sleep activity monitoring," *Telemed. E-Health*, vol. 17, no. 3, pp. 177–184, 2011.

[26] A. R. Fekr, K. Radecka, and Z. A. Zilic, "Design of an e-health respiration and body posture monitoring system and its application for rib cage and abdomen synchrony analysis," in *Proc. IEEE Int. Conf. Bioinf. Bioeng.*, Nov. 2014, pp. 141–148.

[27] X. Sun, L. Qiu, Y. Wu, Y. Tang, and G. Cao, "Sleepmonitor: Monitoring respiratory rate and body position during sleep using smartwatch," *ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 3, p. 104, 2017.

[28] L. Chang, J. Lu, J. Wang, X. Chen, D. Fang, Z. Tang, P. Nurmi, and Z. Wang, "SleepGuard: Capturing rich sleep information using smartwatch sensing data," *ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 3, p. 98, 2018.

[29] K. H. Fielding and D. W. Ruck, "Spatio-temporal pattern recognition using hidden Markov models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 4, pp. 1292–1300, Oct. 1995.

[30] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[31] B.-J. Yoon, "Hidden Markov models and their applications in biological sequence analysis," *Current Genomics*, vol. 10, no. 6, pp. 402–415, 2009.

[32] N. Liu, B. C. Lovell, P. J. Kootsookos, and R. I. A. Davis, "Model structure selection & training algorithms for an HMM gesture recognition system," *Proc. 9th Int. Workshop Frontiers Handwriting Recognit.*, Oct. 2004, pp. 100–105.

[33] C.-Y. Chen, T.-C. Wang, J. F. Wang, and L. P. Shieh, "SVM-based state transition framework for dynamical human behavior identification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 2009, pp. 1933–1936.

[34] A. Paul, B.-W. Chen, K. Bharanitharan, and J.-F. Wang, "Video search and indexing with reinforcement agent for interactive multimedia services," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 2, p. 25, 2013.

[35] B.-W. Chen, C.-Y. Chen, and J.-F. Wang, "Smart homecare surveillance system: Behavior identification based on state-transition support vector machines and sound directivity pattern analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 6, pp. 1279–1289, Nov. 2013.

[36] S. Jeon, A. Paul, H. Lee, Y. Bun, and S. H. Son, "SleePS: Sleep position tracking system for screening sleep quality by wristbands," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Oct. 2017, pp. 3141–3146.

[37] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[39] A. Olivares and J. Ramírez, J. M. Górriz, G. Olivares, and M. Damas, "Detection of (In)activity periods in human body motion using inertial sensors: A comparative study," *Sensors*, vol. 12, no. 5, pp. 5791–5814, 2012.

[40] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, Jun./Jul. 2011, pp. 1–7.

[41] K. W. Spring, "Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review," *Mechanism Mach. Theory*, vol. 21, no. 5, pp. 365–373, 1986.

[42] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: Improving classification performance when training data is skewed," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 8–11.

[43] J. A. Blackard and D. J. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," *Comput. Electron. Agricult.*, vol. 24, no. 3, pp. 131–151, 1999.

[44] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, J. Bravo, R. Hervás, and M. Rodríguez, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Ambient Assisted Living and Home Care*. Berlin, Germany: Springer, 2012, pp. 216–223.

[45] G. Bunkheila. Signal processing and machine learning techniques for sensor data analytics. Mathworks. Accessed: Sep. 21, 2019. [Online]. Available: https://kr.mathworks.com/videos/signal-processing-and-machine-learning-techniques-for-sensor-data-analytics-107549.html

[46] G. Hoang, A. Bouzerdoum, and S. Lam, "Learning pattern classification tasks with imbalanced data sets," in *Pattern Recognition*. 2012, pp. 193–208.

[47] M. Haescher, J. Trimpop, G. Bieber, and B. Urban, "SmartMove: A smartwatch algorithm to distinguish between high- and low-amplitude motions as well as doffed-states by utilizing noise and sleep," in *Proc. 3rd Int. Workshop Sensor-Based Activity Recognit. Interact.*, 2016, p. 1.

[48] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," 2017, *arXiv:1708.08296*. [Online]. Available: https://arxiv.org/abs/1708.08296

[49] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin, "Cyber physical system challenges for human-in-the-loop control," in *Proc. 8th Int. Workshop Feedback Comput.*, San Jose, CA, USA, 2013.

[50] S. Jeon, A. Paul, S. H. Son, and Y. Eun, "Sleep position management system for enhancing sleep quality using wearable devices," in *Proc. 16th ACM Conf. Embedded Networked Sensor Syst.*, 2018, pp. 365–366.

SANGHOON JEON received the B.S. degree from the School of Electronics Engineering, Kyoungpook National University, Daegu, South Korea, in 2012, and the M.S degree from the Department of Information and Communication Engineering, DGIST, Daegu, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include wearable computing, human computer interface, the healthcare & IoT applications, embedded systems, and cyber-physical systems.

TAEJOON PARK received the B.S. degree *(summa cum laude)* in electrical engineering from Hongik University, Seoul, in 1992, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, South Korea, in 1994, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2005, under the supervision of Prof. Kang G. Shin. He was an Associate Professor with the Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea, from February 2011 to February 2015, an Assistant Professor with Korea Aerospace University, Gyeonggi-do, from September 2008 to February 2011, a Principal Research Engineer with Samsung Electronics, Gyeonggi-do, from April 2005 to April 2008, and a Research Engineer with LG Electronics, Seoul, South Korea, from February 1994 to June 2000 (promoted to a Senior Research Engineer, in 2000). He is currently a Full Professor and the Chair of the Department of Robotics Engineering and the Director of the Collaborative AI-Robotics in Engineering (CARE) Center, Hanyang University, Gyeonggi-do, South Korea. He has authored or coauthored over 130 articles/patents, including essential patents for the DVD standard, six of which were cited over 100 times. He has an h-index of 19 with over 1900 cumulative citations according to Google scholar. His current research interests include cyber-physical systems (CPS) and artificial intelligence (AI) with emphasis on deep learning, and their applications to robots, vehicles, and factories. He is a member of ACM.

ANAND PAUL (SM'15) received the Ph.D. degree in electrical engineering with National Cheng Kung University, Taiwan, China, in 2010. He is currently an Associate Professor with the School of Computer Science and Engineering, Kyungpook National University, South Korea. He has done extensive work on big data/IoT based smart cities and had done data analytics using machine learning techniques. He is a delegate representing South Korea for M2M focus group, from 2010 to 2012. He is also an MPEG Delegate representing South Korea. His research interests include big data analytics/IoT and AI/machine learning. He is serving as an Editor in Chief of *International Journal of Smart Vehicles and Smart Transportation* (IGI Global), and also an Associate Editor in IEEE Access, *IET Wireless Sensor Systems*, *ACM Applied Computing Reviews*, *Cyber Physical Systems* (Taylor and Francis), *Human Behavior and Emerging Technology* (Wiley), and *Journal of Platform Technology*. He has also guest edited various International journals. He is the Track Chair for smart human computer interaction in ACM SAC 2014-2019, and the General Chair for the 8th International Conference on Orange Technology (ICOT 2020).

YANG-SOO LEE received the B.S. and M.S. degrees from the School of Medicine, Kyungpook National University, Daegu, South Korea, in 1986 and 1988, respectively, and the Ph.D. degree in physiology from Yeungnam University, Daegu, in 2001. From 1990 to 1994, he completed Rehabilitation Medicine course from Kangdong Sacred Heart Hospital. Since 1994, he has been a Professor with the Department of Rehabilitation Medicine, School of Medicine, Kyungpook National University.

SANG HYUK SON received the B.S. degree in electronics engineering from Seoul National University, the M.S. degree from KAIST, and the Ph.D. degree in computer science from the University of Maryland, College Park. He has been a Professor with the Computer Science Department, University of Virginia, and a WCU Chair Professor with Sogang University. He has been a Visiting Professor with KAIST, the City University of Hong Kong, the *École* Centrale de Lille, France, Linköping University, Sweden, and the University of Skövde, Sweden. He is currently the Chair professor with the Department of Information and Communication Engineering, DGIST. He has authored or coauthored over 340 articles and edited/authored four books in these areas. His research interests include cyber physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He is a member of the Korean Academy of Science and Technology and the National Academy of Engineering of Korea. He is a Founding Member of the ACM/IEEE CPS Week and a member of the Steering Committee for the IEEE RTCSA and Cyber Physical Systems Week. He received the Outstanding Contribution Award from the Cyber Physical Systems Week, in 2012. He was on the Editorial Board of *ACM Transactions on Cyber Physical Systems*, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and *Real-Time Systems* journal.