BIA: Behavior Identification Algorithm Using Unsupervised Learning Based on Sensor Data for Home Elderly

Cuijuan Shang, Chih-Yung Chang, Guilin Chen, Shenghui Zhao, and Haibao Chen

***Abstract*—Behavior identification plays an important role in supporting homecare for the elderly living alone. In literature, plenty of algorithms have been designed to identify behaviors of the elderly by learning features or extracting patterns from sensor data. However, most of them adopted probabilistic models or supervised learning to identify behaviors based on labeled sensor data. This paper proposes a behavior identification algorithm (*BIA*) using unsupervised learning based on unlabeled sensor data for the elderly living alone in smart home. This paper presents the observation of elder behaviors with three features: *Event Order*, *Time Length Similarity* and *Time Interval Similarity* features. Based on these features of behavior observations, two properties of behaviors, including the Event Shift and Histogram Shape Similarity properties, are presented. According to these properties, the proposed *BIA* is developed. Finally, performance results show that the proposed *BIA* outperforms the existing unsupervised machine learning mechanisms in terms of the behavior identification precision and recall.**

***Index Terms*—Behavior identification, histogram, homecare, unsupervised learning, sensor data.**

I. Introduction

HOMECARE is an important research to deal with the aging issue [1], [2], due to the insufficient medical resources and the elderly's preference of their own homes. Nevertheless, the elderly living alone at home has great potential safety hazard, such as falls and fires. Because of the shortage of human resources, there are not enough care givers to provide services to the elderly. Therefore, plenty of technologies are required to support the homecare.

Manuscript received August 4, 2019; revised September 9, 2019; accepted September 19, 2019. Date of publication September 24, 2019; date of current version June 5, 2020. This work was supported in part by the Natural Science Foundation of Anhui Higher Education under Grant KJ2019ZD44, Grant KJ2019A0648, and Grant KJ2018B01, and in part by the Excellent Young Talents of Anhui Higher Education under Grant gxyq2018094. *(Corresponding author: Chih-Yung Chang.)*

C. Shang, G. Chen, S. Zhao, and H. Chen are with the School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China (e-mail: shangcuijuan@chzu.edu.cn; glchen@chzu.edu.cn; zsh@chzu.edu.cn; chb@chzu.edu.cn).

C.-Y. Chang is with the Department of Computer Science and Information Engineering, Tamkang University, Taipei 25137, Taiwan (e-mail: cychang@ mail.tku.edu.tw).

Digital Object Identifier 10.1109/JBHI.2019.2943391

One important part and parcel of homecare is to access the health state of the elderly living home alone. Studies [3] and [4] have shown that behaviors are associated with the healthy state of a person, especially an elder who usually lives regularly. Therefore, plenty of researchers [5], [6] have proposed lots of mechanisms to identify behaviors of the elderly based on a variety of devices from different perspectives.

Based on the types of data collection devices, behavior identification algorithms can be divided into three types, including video-based [6]–[10], wearable-based [5], [11]–[14] and unobtrusive sensor-based [4], [14]–[20], respectively. The video-based category mainly adapted cameras to monitor the behaviors of occupant and the changes of environment in smart homes [6]–[10]. The collected data are video sequences or digitized visual data. Algorithms fallen in vision-based type usually make use of the computer vision techniques for recognizing daily behaviors of the elderly. However, video-based behavior identification algorithms potentially result in the invasion of inhabitant's privacy, which leads to the situation that they are unacceptable.

The second category depends on the wearable devices, such as smart watches and smart rings, to collect the movement and location information of the elderly [12], [13]. However, it's inconvenient for the elderly because wearable devices require users to wear them every day. The results are mainly affected by the cooperation of users.

The unobtrusive sensor-based category depends on the technologies of wireless sensor networks which are applied to monitor daily behaviors of the elderly in smart homes [4], [15]. These sensors, such as pressure, infrared, water flow as well as door magnetic sensors, are commonly deployed in smart homes to identify behaviors of the elderly. The advantages of unobtrusive sensors include low cost, ease of device deployment as well as the least privacy invasion compared to cameras. Therefore, algorithms based on unobtrusive sensors are considered as a promising research issue to deal with the problem of behavior identification in the smart home.

Extensive existing behavior identification studies have been proposed based on unobtrusive sensors [4], [15], [19]. Most of them employed probabilistic models such as naive Bayesian Network, Dynamic Bayesian Network, Markova model, maximum-likelihood estimation or queuing theory, to describe the states and probabilities for the transition from one state to another. These models are usually trained to fit the specific problem.

However, to learn the model, a training dataset has to be collected and annotated, which is expensive and time consuming.

This paper also falls in the unobtrusive sensor-based category. The proposed *Behavior Identification Algorithm* (*BIA*) adopts unsupervised learning based on sensor data for the elderly living alone in smart home. The *Regularity* and *Histogram Shape Similarity Properties* are proposed based on the observation on basic features of the elderly living alone. According to the proposed properties of behaviors, the *BIA* firstly partitions the sensor events into multiple groups each of which will be a collection of sensor events which are generated from the same sensor associated with the same action in approximate time. Then, the *BIA* further treats those groups as primitive elements and groups them into several clusters each of which is associated with a behavior and labeled with a behavior name. The main contributions of this work are listed as follows.

1) *Formalization of problem and constraints:* This paper establishes a problem formulation system to represent the problem and goal of behavior identification for an elderly in home. Four constraints of behavior are also proposed, including *Space*, *Order*, *Time Interval and Time Difference constraints*.
2) *Presenting time regularity observation of behaviors:* Based on the observation on basic features of the elder living alone, the time regularity of behaviors is presented, which includes three features: *Event Order, Time Length Similarity* and *Time Interval Similarity features*. According to these features, new properties are proposed to analyze and explore the relations between sensory data and behaviors.
3) *Proposing Event Shift and Histogram Shape Similarity properties:* The histogram is used to present the distribution of sensor events in this paper. The *Event Shift* and *Histogram Shape Similarity* properties of behaviors are proposed to exploit the implicit relation between sensor events associated with the same behavior. Based on the relations between histograms, the proposed *BIA* takes into account the statistical property of sensor event data and provides elasticity and flexibility in behavior recognition.
4) *Developing Behavior Identification Algorithm:* The proposed *BIA* algorithm uses unsupervised clustering learning to identify behaviors of the elderly living alone based on unlabeled sensor data. The proposed *BIA* takes advantage of statistical properties of sensor data and utilizes unsupervised clustering algorithms to improve the performance of behavior recognition, as compared with the algorithm proposed in [20].

The rest of this paper is organized as follows. Section II summarizes related work. Section III formalizes the assumptions and problem statement of the investigated behavior identification issue. Section IV proposes *Regularity* and *Histogram Shape Similarity* properties of behaviors while Section V proposes the *Behavior Identification Algorithm*(*BIA*). Section VI investigates the performance improvement of the proposed *BIA* against the related work. Section VII gives conclusions and future work of this study.

II. Related Work

In literature, plenty of researches have paid attention to behaviors identification. In general, these researches can be classified into three categories based on the data collection devices, including vision-based, wearable-based and unobtrusive sensor-based classes. The following briefly reviews these studies.

A. Vision-Based Algorithms

Vision-based algorithms [6]–[10] aimed to identify behaviors by processing video sequences, which is an important research field of computer vision. Generally, the behavior identification algorithms took the video as inputs and directly identified the behaviors by using image processing methods [6], [7]. Some vision-based researches assumed that a camera is worn on the user to capture the information about the context. Algorithms applied to the video analysis indirectly identified behaviors from video sequences by comparing changes of the context [8], [9]. In general, vision-based algorithms can reach high accuracy and have made positive contributions in the field of human behavior identification. However, they exhibited shortcomings such as the invasion of privacy and the requirement of computing resources.

B. Wearable-Based Algorithms

There were several wearable-based algorithms [5], [11]–[14] proposed to identify behaviors mostly according to the motion characteristics of the inhabitant. The sensors, such as accelerometer, physiological sensors, magnetometer as well as gyroscope were assumed to be embedded in the wearable devices and used to collect motion data. The features of data are analyzed and used to identify behaviors. These algorithms [12], [13] have shown satisfactory results in both laboratory/clinical and free-living environment settings. However, wearable sensors need to be attached to inhabitants, which result in lots of discomfort to inhabitants.

C. Unobtrusive Sensor-Based Algorithms

The unobtrusive sensors, such as sound sensors, pressure sensors and temperature sensors, were deployed in smart homes and used to measure the appearances and movements of objects [4], [15]–[18]. Researchers can take advantage of unobtrusive sensors for capturing interactions between inhabitants and the smart home environment, to identify behaviors of the inhabitant.

Most of these researches employed probabilistic models to identify the behaviors. Study [19] designed algorithms that automatically learned Markov models for each class of activities. These models were used to identify behaviors that were performed in a smart home. Further, study [15] used the hidden Markov model (HMM) in behavior identification. The HMM is a special case of the DBNs, which is a probabilistic model with a particular structure that makes it easy to learn from data. Once a model is learned, the behavior identification is both easy and efficient to be implemented. However, the HMM is incapable of capturing long range or transitive dependences of the sensor data due to its very strict independence assumptions

TABLE I
COMPARISON OF THE MAIN CHARACTERISTICS OF THE PROPOSED *BIA* WITH THE UNOBTRUSIVE SENSOR-BASED STUDIES

Studies	Dataset	Sensors	Algorithms	Unsupervised
[15]	CASAS	PIR-Door sensors	NBC, HMM,CRF, SVM	No
[16]	Laboratory of University of Ulster	Switch sensors	maximum-likelihood estimation, using Expectation–Maximization	No
[17]	MavLab user data	State-Change sensors	Decision Tree, Markov model	No
[19]	CASAS	PIR-Door sensors	Markov model	No
[20]	WMNL2016	State-Change sensors	String matching, sliding window-based approach	Yes
The Proposed *BIA*	WMNL2016	State-Change sensors	DBSCAN, Hierarchical Clustering	Yes

(on the observations). Furthermore, without significant training, an HMM may not be able to identify all of the possible sensor sequences that can be consistent with a particular behavior.

Study [16] built up a probabilistic model based maximum-likelihood estimation using Expectation–Maximization. The model can learn behavioral patterns from sensor readings by making full use of all limited labeled activities. Study [17] proposed a new approach for behavioral pattern extraction using decision tree and Markov model based on the On–Off states of home appliances. Study [20] proposed a behavior identification algorithm, namely MLAR, which transforms the sensor data into strings and adopts string matching to recognize elderly behaviors. The proposed MLAR doesn't require the labeled sensor data and identify behaviors based on the spatial-temporal constraints. However, it lacks elasticity and flexibility due to the rigid string matching and satisfactory of spatial-temporal constraints.

Most of the abovementioned researches can identify some interested behaviors with certain accuracy in the considered environment. However, they require labeled training dataset to build and learn the model, which is expensive and time consuming. Different from existing work, this paper aims to propose a *Behavior Identification Algorithm* (*BIA*) using unsupervised learning based on sensor data for the elderly living alone in smart home. The sensor data collected from the unobtrusive sensors don't need to be annotated by the resident. A comparative summary of the characteristics of the proposed *BIA* and the related works based on the unobtrusive sensor is given in Table I. As shown in Table I, the data set, deployed sensors and algorithms of the main related work are presented. Compared with the related work, the proposed *BIA* employs unsupervised clustering algorithms, including DBSCAN and Hierarchical Clustering, to identify behaviors of the elderly living alone with elasticity.

III. ASSUMPTIONS AND PROBLEM STATEMENT

This section initially introduces the network environment and assumptions. Then the problem statement and related constraints are described.

A. Network Environment

Let $\mathbb{B} = \{b^1, b^2, \ldots, b^{|\mathbb{B}|}\}$ denote the set of interested behaviors which are highly correlated to the health of the elderly people. Let $b^i = (a_1^i, a_2^i, \ldots, a_{|b^i|}^i)$ denote a behavior which consists of an ordered list of actions, where a_j^i denote the *j*-th action of behavior b^i. For the ease of presentation, in the later description, we use action $a_j^i \dot{\in} b^i$ to represent that action a_j^i is in action list of behavior b^i. Assume that a smart home space can be divided into several small areas, which is denoted by $\mathbb{R} = \{r_1, r_2, \ldots r_{|\mathbb{R}|}\}$. In order to recognize actions which compose behaviors, a set of sensors $\mathbb{S} = \{s_1, s_2, \ldots, s_n\}$ is deployed in the smart home. Each sensor $s \in \mathbb{S}$ has an unique ID and a fixed location area, denoted as Id and $loc \in \mathbb{R}$, respectively. When the elderly is active at home, some actions will trigger the corresponding sensors. As soon as a sensor is triggered by an action *a*, a sensor event *e* creates. Let notation occ(*a*) represent the occurrence of an action *a*. Let $\mathcal{A}(\cdot)$ and $\mathcal{E}(\cdot)$ be two functions which take event *e* and occurrence of action *a* as their inputs and returns action *a* and event *e*, respectively. That is, $a = \mathcal{A}(e)$ and $e = \mathcal{E}(occ(a))$. Further, we assume that any event *e* can only be triggered by the occurrence of one action.

The detailed information of a sensor event *e* can be denoted by $e\ (Id, loc, t^{start}, t^{end}, value)$, where Id denotes the sensor ID, loc denotes the location area of the sensor, t^{start} denotes the time when the sensor event occurs, t^{end} denotes the time when the sensor event ends and *value* denotes the reading of the sensor. Let e_i denote the *i*-th sensor event of any sensor event set. Furthermore, let $E_j = (e_1, e_2, \ldots, e_{|E_j|})$ denote the sequence of sensor events collected in the *j*-th day. It is obvious that $\mathbb{E} = \cup_{j=1}^m E_j$ denote the set of all sensor events collected in the past *m* days.

Since a behavior $b^i = (a_1^i, a_2^i, \ldots, a_{|b^i|}^i)$ can be represented as $(\mathcal{A}(e_1), \mathcal{A}(e_2), \ldots, \mathcal{A}(e_{|b^i|}))$, the sequence $(e_1, e_2, \ldots, e_{|b^i|})$ is called the *event sequence* of behavior b^i. For the ease of presentation, in the later description, notation $b^i \to (e_1^i, e_2^i, \ldots, e_{|b^i|}^i)$ denotes that behavior b^i triggers the event sequence $(e_1^i, e_2^i, \ldots, e_{|b^i|}^i)$.

In the next subsection, the goal and its associated constraints of the investigated problem will be presented.

B. Problem Statement

To observe the health condition of an elderly at home, this paper aims to design a *Behavior Recognition* system to identify the behaviors of an elderly from spontaneous sensor events. The following represents the problem statement and related constraints.

Given a set $\mathbb{E}$ of sensor events which are collected in the past m days, this paper aims to identify the behaviors of the elderly. Let $B_j = (b_{j,1}, b_{j,2}, \ldots, b_{j,|B_j|})$ denote the sequence of behaviors which actually occur in the j-th day. Let $T_{j,k}$ denote the time period of behavior $b_{j,k}$. Let $\mathbb{T}_j = (T_{j,1}, T_{j,2}, \ldots, T_{j,|B_j|})$ denote the sequence of the time periods corresponding to behaviors in B_j. Let $\delta_{j,k,l}$ be a Boolean variable which represents the fact whether or not behavior $b_{j,k}$ occurs in time period $T_{j,l}$. This is,

$$\delta_{j,k,l} = \begin{cases} 1, & b_{j,k} \text{ occurs in } T_{j,l} \\ 0, & \text{otherwise} \end{cases}.$$

It is obvious that for any pair of $b_{j,k}$ and $T_{j,l}$, if $k = l$, we have $\delta_{j,k,l} = 1$ and $\delta_{j,k,l} = 0$ for all $k \neq l$.

Furthermore, let $\hat{B}_j^A = (\hat{b}_{j,1}, \hat{b}_{j,2}, \ldots, \hat{b}_{j,|\hat{B}_j^A|})$ denote the sequence of recognized behaviors extracted from E_j by applying behavior recognizing algorithm A. The following defines some notations which represent the recognized results by executing a given algorithm A. Let $\hat{\delta}_{j,k,l}$ be a Boolean variable representing whether or not $\hat{b}_{j,k}$ occurs in time period $T_{j,l}$. This is,

$$\hat{\delta}_{j,k,l} = \begin{cases} 1, & \hat{b}_{j,k} \text{ occurs in } T_{j,l} \\ 0, & \text{otherwise} \end{cases}$$

For a considered algorithm A, the correct identification that behavior $b_{j,k}$ occurs in $T_{j,l}$ is treated as $TP_{j,k}^A$ (True Positive) which is measured by

$$TP_{j,k}^A = \sum_{l=1}^{|B_j|} \delta_{j,k,l} * \hat{\delta}_{j,k,l} \tag{1}$$

Exp. (1) indicates that, the value of $TP_{j,k}^A$ will be added by 1 if both $b_{j,k}$ and $\hat{b}_{j,k}$ occur in time period $T_{j,l}$. Further, two types of errors, False Positive $FP_{j,k}^A$ and False Negative $FN_{j,k}^A$, can be measured by Exps. (2) and (3), respectively.

$$FP_{j,k}^A = \sum_{l=1}^{|B_j|} \max\left(0, \hat{\delta}_{j,k,l} - \delta_{j,k,l}\right) \tag{2}$$

$$FN_{j,k}^A = \sum_{l=1}^{|B_j|} \max\left(0, \delta_{j,k,l} - \hat{\delta}_{j,k,l}\right) \tag{3}$$

Exp. (2) indicates that, the value of $FP_{j,k}^A$ will be added by 1 if $\hat{b}_{j,k}$ occurs in $T_{j,l}$ but $b_{j,k}$ doesn't occur in $T_{j,l}$ while Exp. (3) indicates that, the value of $FN_{j,k}^A$ will be added by 1 if $b_{j,k}$ occurs in $T_{j,l}$ but $\hat{b}_{j,k}$ doesn't occur in $T_{j,l}$. Then, the recall and precision of behavior $b_{j,k}$, denoted by $RC_{j,k}^A$ and $PC_{j,k}^A$ respectively, can be calculated as follows:

$$RC_{j,k}^A = \frac{TP_{j,k}^A}{TP_{j,k}^A + FN_{j,k}^A} \tag{4}$$

$$PC_{j,k}^A = \frac{TP_{j,k}^A}{TP_{j,k}^A + FP_{j,k}^A} \tag{5}$$

As shown in Exps. (4) and (5), the values of $RC_{j,k}^A$ and $PC_{j,k}^A$ will both equal to 1 if neither false positive nor false negative occurred. This also implies that the behavior recognition algorithm A expects to have high values (approaching 1) of $RC_{j,k}^A$ and $RC_{j,k}^A$ for any given behavior $b_{j,k}$ in the j-th day.

Let $\mathbb{A}$ denote the set of all possible recognition algorithms. This paper aims to propose a behavior recognition algorithm $A \in \mathbb{A}$ which maximizes the total recognition Recall and Precision of all actual behaviors $b_{j,k}$ in the past m days. Exp. (6) presents the objective function of this paper.

Objective Function:

$$\max_{A \in \mathbb{A}} \sum_{j=1}^{m} \sum_{k=1}^{|B_j|} \left(\alpha * RC_{j,k}^A + (1-\alpha) * PC_{j,k}^A\right) \tag{6}$$

The following introduces the assumptions and constraints that make up the premise of our work. Let $\beta_{j,k}$ denote a Boolean variable representing whether or not a sensor event $e_j \in \mathbb{E}$ occurs in area r_k. That is,

$$\beta_{j,k} = \begin{cases} 1, & e_j.loc = r_k \\ 0, & \text{otherwise} \end{cases}.$$

This paper assumes that a behavior can only occur in one area. This also indicates that sensor events associated with one behavior have the same location. Given a behavior $b^i \rightarrow (e_1^i, e_2^i, \ldots, e_{|b^i|}^i)$, the following space constraint guarantees that sensor events associated with behavior b^i should be occurred in the same area.

1) Space Constraint:

$$\text{If } b^i \text{ occurs in } r_k, \prod_{e_j^i \in b^i} \beta_{j,k} = 1 \tag{7}$$

Given a behavior $b^i \rightarrow (e_1^i, e_2^i, \ldots, e_{|b^i|}^i)$, the order constraint guarantees that sensor events belonging to b^i should be triggered in the fixed order. Let $e_j^i = (Id, loc, t_j^{i,start}, value)$ denote the j-th sensor event of behavior b^i. Exp. (8) presents the order constraint.

2) Order Constraint:

$$\text{if } \forall e_j^i, e_k^i \in b^i \text{ and } j < k, t_j^{i,start} < t_k^{i,start} \tag{8}$$

IV. The Properties of Behaviors

This section initially presents the observation on the elder behaviors, named Time Regularity Observation. Three features based on the observation are further introduced. Then, two properties, namely Event-Shift and Histogram Shape Similarity are deduced in turn based on the features of Time Regularity Observation.

A. Observation on the Elder Behaviors

Consider a behavior $b^i \rightarrow (e_1^i, e_2^i, \ldots, e_{|b_i|}^i) \in \mathbb{B}$ which triggers a sequence of sensor events $e_j^i, 1 < j < |b^i|$. For instance, when the elder goes to the toilet, he will 'open' the door of the washroom, 'close' the door after enter the room, 'sit' on the toilet, 'wash' the hands after using the toilet, and finally 'open' the door. In this example, the behavior b^i denotes 'goes to the toilet' while sensor events e_1^i, e_2^i, e_3^i, e_4^i, and e_5^i denote the detections of actions such as 'open door', 'close door', 'sit', 'wash hands', 'open door', respectively.

Consider the past *m* days. Since the behavior b^i of an elder has regularity, it means that b^i is likely occurred around a certain time on most days. Let $b_k^i \rightarrow (e_{1,k}^i, e_{2,k}^i, \ldots, e_{|b_i|,k}^i)$ denote behavior b^i occurred in the *k*-th day, where notation $e_{j,k}^i$ denote the *j*-th event in b_k^i. For the ease of representation, notation e_j^i is considered as a sensing event type which is associated with action a_j^i. Let $t_{j,k}^{i,start}$ denote the occurrence time point of $e_{j,k}^i$. Let $\bar{t}_{j,m}^{i,start}$ and $\sigma_{j,m}^i$ denote the mean and standard deviation values of the $t_{j,k}^{i,start}$ in the past *m* days, respectively. Assume that $t_{j,k}^{i,start}$ follows Normal distribution. It implies that the start times of sensor event $e_{j,k}^i$ in past *m* days mostly fall into $[\bar{t}_{j,m}^{i,start} \pm 3\sigma_{j,m}^i]$. Let $\vartheta_{j,k}$ denote a Boolean variable which represents whether or not a sensor events $e_{j,k}^i$ occur in $[\bar{t}_{j,m}^{i,start} \pm 3\sigma_{j,m}^i]$. This is,

$$\vartheta_{j,k} = \begin{cases} 1, & t_{j,k}^{start} \in \left[\, \bar{t}_{j,m}^{i} \pm 3\sigma_{j,m}^{i} \right] \\ 0, & \text{otherwise} \end{cases}.$$

The following presents the **Time Regularity Observation**.

Time-Regularity (T-R) Observation:

$$\forall e_{j,k}^i \in b^i \text{ and } 1 \le k \le m,\ 0.9974 \le \left(\frac{\sum_{k=1}^{m} \vartheta_{j,k}}{m} \right) \le 1$$

The time regularity of an elder's behaviors exhibits several important features of the events. According to the regularity of an elder's behaviors, the order of sensors triggered by the same behavior has regularity, even though the same behavior occurred in different days. This also implies that if events $e_{j,k}^i \in b_k^i$ is occurred earlier than event $e_{z,k}^i \in b_k^i$ in *k*-th day, we have, the event $e_{x,l}^i$ is occurred earlier than event $e_{y,l}^i$ in *l*-th day, where $\mathcal{A}(e_{j,k}^i) = \mathcal{A}(e_{x,l}^i)$ and $(e_{z,k}^i) = \mathcal{A}(e_{y,l}^i)$. The following presents the *Event-Order Feature*.

1) Event-Order Feature:

$$\forall\ e_{j,k}^i, e_{z,k}^i \in b_k^i,\ \forall e_{x,l}^i, e_{y,l}^i \in b_l^i,\ 1 \le k,l \le m\ ,\ x < y \text{ iff } \mathcal{A}\left(e_{j,k}^i\right) = \mathcal{A}\left(e_{x,l}^i\right),\ \mathcal{A}\left(e_{z,k}^i\right) = \mathcal{A}\left(e_{y,l}^i\right) \text{ and } j < z.$$

The following presents another important feature of the behavior regularity. When the same action occurred in different days, the associated events have similar time length. For example, time length of event associated with action 'open door' is similar, even though the 'GoToToilet' behavior occurred in different days. The following presents this feature.

2) Time-Length-Similarity Feature:

$$\forall e_{j,k}^i \in b_k^i,\ \forall e_{j,l}^i \in b_l^i, 1 \le k,l \le m,\ (t_{j,k}^{i,end} - t_{j,k}^{i,start}) \approx \left(t_{j,l}^{i,end} - t_{j,l}^{i,start}\right) \text{ iff } \mathcal{A}\left(e_{j,k}^i\right) = \mathcal{A}\left(e_{j,l}^i\right).$$

Furthermore, the time interval between two actions also has important feature. It is obvious that an elder likely walks at an almost constant speed. Let v denote the moving speed of an elder. As we know, each sensor is deployed in a fixed location of an indoor space. Consider a certain behavior $b^i \rightarrow (e_1^i, e_2^i, \ldots, e_{|b^i|}^i)$. Let $L = (u_1, u_2, \ldots, u_{|U|})$ denote the sequence of locations corresponding to the deployed sensors which detect these events. Let $d(u_j, u_k)$ denote the distance between $u_j \in L$ and $u_k \in L$. The time that takes the elder to move from locations u_j to u_{j+1} can be calculated as $\frac{d(u_i,u_{i+1})}{v}$. Obviously, the value of $\frac{d(u_i,u_{i+1})}{v}$ is nearly a constant because that both v and (u_j, u_k) are constant. The start time of sensor event $e_{j+1,k}^i$ detected in the *k*-th day can be represented as the end time of the last sensor event $t_{j,k}^{end}$ plus the time duration required for the elder moving from u_j to u_j. That is, we have

$$t_{j+1,k}^{start} = t_{j,k}^{end} + \frac{d\left(u_i, u_{i+1}\right)}{v} \text{ and } t_{j+1,k}^{start} - t_{j,k}^{end} = \frac{d\left(u_i, u_{i+1}\right)}{v}.$$

Since the value $\frac{d(u_i,u_{i+1})}{v}$ is nearly a constant, the time interval between any adjacent events in the sequence of the same behavior is almost a constant. The following presents the *Time-Interval-Similarity Feature*.

3) Time-Interval-Similarity Feature:

$$\forall\ e_{j,k}^i \in b_k^i,\ \forall e_{j,l}^i \in b_l^i,\ 1 \le k,l \le m,\ t_{j+1,k}^{i,start} - t_{j,k}^{i,end} \approx t_{j+1,l}^{i,start} - t_{j,l}^{i,end},\ \text{iff } \mathcal{A}\left(e_{j,k}^i\right) = \mathcal{A}\left(e_{j,l}^i\right).$$

B. Histogram of Sensor Events

This subsection further analyzes the regularity of sensor events detected due to the occurrence of a certain behavior $b^i = (a_1^i, a_2^i, \ldots, a_{|b^i|}^i)$. Herein, we notice the regularity of behavior implies that behavior b^i, for example, 'GoToToilet', should be occurred every day but the occurred time might be different. Consequently, the action a_j^i will be detected by a certain type of sensor events and the detected time of every day is also different. To analyze the distribution of *m* events $e_{j,k}^i$ for $1 \le k \le m$, the histogram is used.

A histogram of events $e_{j,k}^i$ which occurred due to action a_j^i of behavior b^i occurred in different days is a graphical display of data using bars with different heights. The width of each bar is also called the *class interval*, denoted by τ, which might be an hour, or 10 minutes. Each bar, called bin, counts the number of occurrences of data in the data set as its height. It is obvious that the *x*-axis is a time line which represents the 24 hours of one day. Let T^D denote the time length of one day and is divided into *q* bins by τ. That is, $T^D = (\tau_1, \tau_2, \ldots, \tau_q)$. For instance, if $\tau = 10$ minutes, we have $q = 1440$.

Let $\boldsymbol{h_{j,x}^i}$, or $\boldsymbol{h_{j,x}}$ in short, denote the ***frequency*** of action a_j^i occurred in time period τ_x. Assume that action a_j^i occurred in the period τ_x of the *k*-th day, which triggered a sensor event $e_{j,k}^i$. It implies that $t_{j,k}^{i,start} \in \tau_x$ and consequently, $h_{j,x}^i = h_{j,x}^i + 1$. According to the *Event-Order Feature*, action a_{j+1}^i in behavior b^i would similarly occur in the *k*-th day and triggered a sensor event $e^i_{j+1,k}$. Further, we have,

$$\begin{aligned} t^{i,start}_{j+1,k} &= t^{i,end}_{j,k} + \frac{d\left(u_i, u_{i+1}\right)}{v} \\ &= t^{i,start}_{j,k} + \left(t^{i,end}_{j,k} - t^{i,start}_{j,k}\right) + \frac{d\left(u_i, u_{i+1}\right)}{v} \\ &= t^{i,start}_{j,k} + T^c_{a_i} \\ &= t^{i,start}_{j,k} + \left(\frac{T^c_{a_i}}{\tau}\right)\tau, \end{aligned}$$

where $T^c_{a_i} = (t^{i,end}_{j,k} - t^{i,start}_{j,k}) + \frac{d(u_i,u_{i+1})}{v}$.

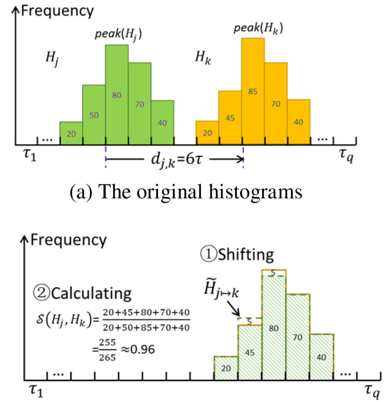
According to the *Time-Length-Similarity* and *Time-Interval-Similarity Features*, the $T^c_{a_i}$ is almost a constant. Let $C_i =$ **Round** $(\mathrm{T}^c_{a_i}/\tau)$ denote a function which rounds value $\mathrm{T}^c_{a_i}/\tau$ and returns an **Integer** C_i. Then, we have, $t^{i,start}_{j+1,k} \approx t^{i,start}_{j,k} + C_i\tau$. As a result, we have $t^{i,start}_{j+1,k} \in \tau_{x+C_i}$. This derivation also concludes the following important property.

1) Event-Shift Property: $\forall a^i_j \in b_i$, $t^{i,start}_{j+1,k} \in \tau_{x+C_i}$ *iff* $t^{i,start}_{j,k} \in \tau_x$, where C_i is a constant.

According to *Event-Shift Property*, any $h^i_{j,x} \in H^i_j$ is almost equal to $h^i_{j+1,x+C_i} \in H^i_{j+1}$. In other words, the histogram H^i_{j+1} almost can be obtained by left shifting histogram H^i_j with $C_i\tau$ length of time period. Herein, it is noticed that the shapes of H^i_j and H^i_{j+1} are similar for any action $a^i_j \dot{\in} b^i$. This also implies that the shapes of H^i_{j+1} and H^i_{j+2} are also similar since j can be substituted by $j+1$. Based on the transitive relation on 'similarity', it is concluded that the shapes of H^i_j and H^i_k are similar for any actions $a^i_j \dot{\in} b^i$ and $a^i_k \dot{\in} b^i$.

Let *peak bin*, denoted by *peak* (H^i_j), of a histograms H^i_j represent the highest bar in H^i_j. Let notation $d_{j,k}$ denote the distance between *peak* (H^i_j) and *peak* (H^i_k). It is obvious that $d_{j,k} = \alpha\tau$. The following defines ***Shift*** operation operated on two histograms H^i_j and H^i_k. Let $\tilde{H}^i_{j\mapsto k} = \boldsymbol{Shift}_{j\mapsto k}(H^i_j, H^i_k)$ denote a shift operation which shifts *peak* (H^i_j) in right direction with a distance of $d_{j,k} = \alpha\tau$ to align with *peak* (H^i_k) and returns a new histogram $\tilde{H}^i_j$. Recall that $H^i_j = (h^i_{j,1}, h^i_{j,2}, \ldots, h^i_{j,q})$ can be obtained from the occurrences of action $a^i_j \dot{\in} b^i$ in the past *m* days. We have, $\tilde{H}^i_{j\mapsto k} = \boldsymbol{Shift}_{j\mapsto k}(H^i_j, H^i_k) = (h^i_{j,q-\alpha+1}, h^i_{j,q-\alpha+2}, \ldots, h^i_{j,q}, h^i_{j,1+\alpha}, h^i_{j,2+\alpha}, \ldots, h^i_{j,q-\alpha})$.

After executing $\boldsymbol{Shift}_{j\mapsto k}(H^i_j, H^i_k)$ operation, the ***Similarity*** between histograms H^i_j and H^i_k, denoted by $\mathcal{S}(H^i_j, H^i_k)$, can be calculated by applying the Exp. (9). By performing the right shift operation on H^i_j, the resultant histogram might be partitioned into two parts: $h^i_{j,q-\alpha+x}$ for $1 \leq x \leq \alpha$ and $h^i_{j,x-\alpha}$ for $\alpha+1 \leq x \leq q$. The operation $\sum^{\alpha}_{x=1} \max(h^i_{j,q-\alpha+x}, h^i_{k,x})$ operation calculates the area size of the union of two histograms H^i_j and H^i_k for $1 \leq x \leq \alpha$ while operation $\sum^{q}_{x=1+\alpha} \max(h^i_{j,x-\alpha}, h^i_{k,x})$ calculates the area size of the union of two histograms H^i_j and H^i_k for $\alpha+1 \leq x \leq q$. As a result, the denominator part of Exp. (9) calculates the total area size of the union of two histograms $\tilde{H}^i_{j\mapsto k}$ and H^i_k. Similarly, the numerator part of Exp. (9) calculates the total size of the intersection of two histograms $\tilde{H}^i_{j\mapsto k}$ and H^i_k. In particular, in case that two histograms $\tilde{H}^i_{j\mapsto k}$ and H^i_k are identical, Exp. (9) has value 1. It is obvious that Exp. (9) returns a smaller value if the two histograms $\tilde{H}^i_{j\mapsto k}$ and H^i_k have a big difference.



(a) The original histograms

(b) The procedure of the *Similarity* calculation

Fig. 1. The example of *Similarity* calculation of H_j and H_k.

$$\mathcal{S}\left(H^i_j, H^i_k\right) = \frac{\sum^{\alpha}_{x=1} \min\left(h^i_{j,q-\alpha+x}, h^i_{k,x}\right) + \sum^{q}_{x=1+\alpha} \min\left(h^i_{j,x-\alpha}, h^i_{k,x}\right)}{\sum^{\alpha}_{x=1} \max\left(h^i_{j,q-\alpha+x}, h^i_{k,x}\right) + \sum^{q}_{x=1+\alpha} \max\left(h^i_{j,x-\alpha}, h^i_{k,x}\right)} \tag{9}$$

Obviously, the more similar the two histograms are, the larger the value $\mathcal{S}(H^i_j, H^i_k)$ will be. The following presents the ***Histogram Shape Similarity Property*** which will be used to evaluate whether or not two actions a^i_j and a^i_k are in the same action sequence of behavior b^i.

2) Histogram-Shape-Similarity (H-S-S) Property:

$$\forall e^i_j, \forall e^i_k \in b_i \text{ and } 1 \leq j, k \leq |b_i|, \mathcal{S}\left(H^i_j, H^i_k\right) \approx 1$$

Fig. 1 gives an example to elaborate the calculation of ***Similarity*** between any two histograms. As shown in Fig. 1(a), histograms H_j and H_k are associated to actions a_j and a_k, respectively. If we want to check whether or not the two actions belong to the same behavior, the *Similarity* between H_j and H_k can be calculated. As shown in Fig. 1(b), H_j is right shifted with distance $d_{j,k}$ to align the *peak bin* of H_k. As a result, there is a large overlap region, as marked with backslash shadow, between the shifting result $\tilde{H}_{j\mapsto k}$ and H_k. By applying Equ. (9), we have $\mathcal{S}(H_j, H_k) \approx 0.96$. According to the *H-S-S Property*, it implies that actions a_j and a_k might belong to the same behavior.

V. The Proposed Mechanism

This paper proposes a behavior identification algorithm, called *BIA*, which aims to accurately identify behaviors of the elderly based on the set of sensor events $\mathbb{E} = \bigcup^m_{j=1} E_j$. The *BIA* algorithm mainly consists of three phases, namely Sensor Event Grouping (*SEG*), Histogram Similarity Grouping (*HSG*) and Behavior Matching (*BM*) phases. The first phase takes the collected sensor data as its inputs, aiming to construct a number of disjoint groups using unsupervised clustering algorithm. Each group, called *primitive group*, will be a collection of sensor

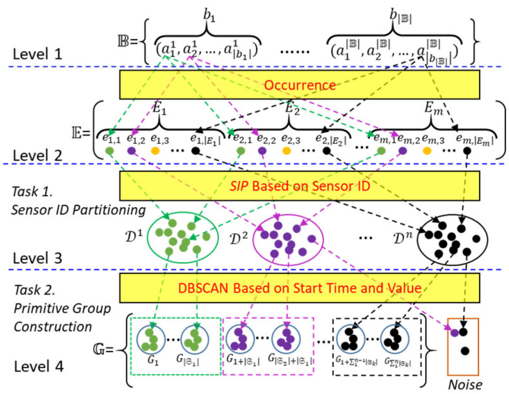


Fig. 2. The procedure of partitioning event set $\mathbb{E}$ in SEG phase.

event data which are generated from the same sensor associated with the same action in approximate time. Then the second phase treats these *primitive groups* as primitive elements, and constructs the histogram for each group. It aims to build up several *behavior clusters* by applying clustering algorithm on the histograms of *primitive groups*. Each *behavior cluster* has high correlation with a certain unknown behavior. Finally, the *BM* phase labels each *behavior cluster* with a behavior $b^i \in \mathbb{B}$ based on similarity comparison.

A. Sensor Event Grouping Phase

This phase aims to construct a number of disjoint sensor *primitive groups* using unsupervised clustering algorithm. Consider an event type $e_k = \mathcal{E}(occ(a_k))$ which is occurred due to the occurrence of action a_k. Let notation $e_{k,i}$ denote event type e_k occurred in the *i*-th day of the past *m* days. Let $G_k = \{e_{k,1}, e_{k,2}, \ldots, e_{k,m}\}$, called *primitive group*, denoting the set of same event type e_k which are occurred in the past *m* days in approximate time. Given a set of sensor events $\mathbb{E} = \bigcup_{j=1}^{m} E_j$ collected in the past *m* days, this phase aims to partition all events in $\mathbb{E}$ into a number of G_k for each event type e_k.Two tasks, namely *Sensor ID Partitioning* (*SIP*) and *Primitive Group Construction Tasks*, will be implemented to achieve the goal of this phase.

1) Sensor ID Partitioning Task: Since the sensor ID is an important feature of an event type e_k, the *Sensor ID Partitioning* (*SIP*) *Task* partitions the event set $\mathbb{E}$ into multiple small sets according to the sensor ID. The execution results of SIP step will be *n* sets of $\mathcal{D}^i$, $1 \le i \le n$, where

$$\mathcal{D}^i = \{e|e.Id = s_i.Id, e \in \mathbb{E}\} \text{ and } \mathbb{E} = \bigcup_{i=1}^{n} \mathcal{D}^i. \tag{10}$$

As shown in Fig. 2, each interested behavior consists of a list of actions in level one. The occurrence of an action will create the corresponding event. The sensor events collected in the past *m* days is depicted in the second level. In level three, *n* small primitive groups are obtained by partitioning set $\mathbb{E}$ based on the sensor ID.

2) Primitive Group Construction Task: The *Primitive Group Construction Task* constructs the *primitive groups* based on two important features, including the start time and value. Recall the *Time-Regularity (T-R) Observation* that the start times of events associated to the same action in a behavior are closed. To obtain *primitive group* G_k, the *DBSCAN* algorithm will be applied to partition the *n* sets of $\mathcal{D}^i$ into several primitive groups based on start time and value of each event. In order to check whether or not two events $e_j \in \mathcal{D}^i$ and $e_l \in \mathcal{D}^i$ belong to the same *primitive group* G_k, the distance between them, denoted by $d(e_j, e_l)$, will be defined. First, the distributions of start time $[t_{\min}^{start},\ t_{\max}^{start}]$ and reading value $[v_{\min},\ v_{\max}]$ of each sensor event should be normalized. For the ease of presentation, the event with normalized attributes is simply denoted by notation $ee(\ddot{t}, \ddot{v})$, where $\ddot{t}$ and $\ddot{v}$ denote the normalized start time and reading value, respectively. Then, the distance $d(e_j, e_l)$ of events e_j and e_l can be measured by the Equ. (11).

$$d(e_j, e_l) = \sqrt{\left(e_j.\ddot{t} - e_l.\ddot{t}\right)^2 + \left(e_j.\ddot{v} - e_l.\ddot{v}\right)^2} \tag{11}$$

The DBSCAN has two main parameters: *min_pts* and *Eps*. The parameter *min_ps* is the minimum number of points in a cluster while the *Eps* is the minimum distance between two data points for them to be considered in the same cluster. After applying DBSCAN, each sensor event can be either included in a *primitive group* or treated as a noise as shown in the last layer of Fig. 2. Given any $G_k \in \mathbb{G}$, any two events $e_{k,j} \in G_k$ and $e_{k,l} \in G_k$ have the relation $\mathcal{A}(e_{k,j}) = \mathcal{A}(e_{k,l})$. That is, they are occurred due to same action performed in approximate time in different days. It's obvious that $\mathbb{G} = \bigcup_{k=1}^{|\mathbb{G}|} G_k$ denote the set of all *primitive groups* that this phase aims to obtain. The grouping result highly depends on the parameters *min_pts* and *Eps* and the effect of different values of parameters *min_pts* and *Eps* on clustering performance will be discussed in Simulation Section.

B. Histogram Similarity Grouping Phase

In the last phase, all sensor events have been partitioned into a set of several *primitive groups* $\mathbb{G} = \bigcup_{i=1}^{|\mathbb{G}|} G_k$. In this phase, these *primitive groups* will be further clustered into several *behavior clusters*. Recall the ***Histogram-Shape-Similarity (H-S-S) Property*** that if two actions a_j^i and a_k^i are occurred due to same behavior, the similarity $\mathcal{S}(H_j^i, H_k^i)$ should have a high value (approximate to 1). According to this property, this phase will treat the histogram of *primitive group* G_k, denoted by H_k, as the representation of G_k, aiming to partition histograms into several *behavior clusters.*

This phase consists of three tasks, namely Histogram Construction, Space-based Histogram Clustering and Similarity-based Hierarchical Clustering tasks. The following will present the details of three tasks.

1) Histogram Construction Task: In this task, the histogram will be constructed for each *primitive group* $G_k \in \mathbb{G}$. Recall that notation $\boldsymbol{h_{j,x}^i}$ denote the ***frequency*** of action a_j^i occurred due to the occurrence of behavior b^i in time period $\tau_x \in T^D$. Let $\eta_{x,y}$ denote a Boolean variable representing whether or not the event $e_{k,y} \in G_k$ occurs in the time period $\tau_x \in T^D$. That is

$$\eta_{x,y} = \begin{cases} 1, & e_{k,y}.t^{start}/\tau \ = \ \tau_x \\ 0, & \text{otherwise} \end{cases}$$

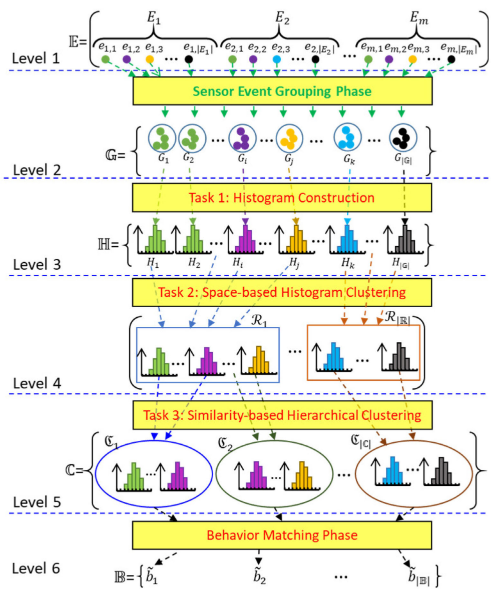


Fig. 3. The procedure of partitioning histogram set $\mathbb{H}$ in HSG phase.

Then, the histogram H_k of G_k can be calculated by:

$$H_k = \left\{ h_{k,x} | h_{k,x} = \sum_{e_{k,y} \in G_k} \eta_{x,y} \, , \forall \tau_x \in T^D \right\}. \quad (12)$$

It is obvious that the histogram H_k is another presentation of *primitive group* G_k. Till now, the set of histograms $\mathbb{H} = \{H_1, H_2, \ldots, H_{|\mathbb{H}|}\}$ has been constructed. As shown in Fig. 3, the sensor events collected in the past *m* days are depicted in level one and the *primitive groups* are obtained in level two after execution of the Sensor Event Grouping Phase. In level three, each histogram will be constructed for each *primitive group* after the execution of this task.

2) Space-Based Histograms Clustering Task: Recall that each G_k only collects those events triggered by the same sensor. Since the occurrence of one behavior consists of several actions which trigger different sensors to detect the events, the events belonging to same behavior but triggered by different sensors at the same space should be grouped in the same cluster, even though they belong to different G_k. Therefore, an important operation for behavior recognition is to group the histograms of different G_k into same cluster if they have same space. Let $H_k.loc$ denote the occurrence space of G_k. Let $\mathcal{R}_j = \{H_k | H_k.loc = r_j\}$ where $1 \leq j \leq |\mathbb{R}|$. That is, all histograms associated with same space r_j will be collected in the same cluster $\mathcal{R}_j$. It is obvious that the result of this task is $\mathbb{H} = \bigcup_{i=1}^{|\mathbb{R}|} \mathcal{R}_j$. As shown in Fig. 3, all histograms in set $\mathbb{H}$ are grouped into $|\mathbb{R}|$ clusters according to the occurrence space in level four. The next task of this phase aims to construct a number of *behavior clusters* each of which consists of several histograms which are correlated to the same behavior.

3) Similarity-Based Hierarchical Clustering Task: In this task, the Hierarchical Clustering (HC) algorithm will be applied to the histograms in each subset $\mathcal{R}_i \in \mathbb{H}$. Given a cluster $\mathcal{R}_i$, this task will perform the following two steps.

In the first step, histograms in set $\mathcal{R}_i$ will be merged round by round to construct a hierarchical tree. Each node of the hierarchical tree is a histogram cluster, denoted by U_h, which consists of one or more histograms. During the process of hierarchical tree construction, a measure of distance between clusters is required to decide the best two clusters to be merged. Consider two clusters U_x and U_y. Let $d(U_x, U_y)$ denote the distance of U_x and U_y, which is measured by the distances between histograms in clusters U_x and U_y. According to the ***Histogram-Shape-Similarity Property***, if two histograms have similar shapes, they have large value of the similarity. Recall that the notation $\mathcal{S}(H_k, H_l)$ denote the similarity between histograms H_k and H_l and notation $d_{k,l}$ denote the distance between *peak* (H_k) and *peak* (H_l). The distance $d(H_k, H_l)$ between histograms H_k and H_k can be represented by $\mathcal{S}(H_k, H_l)$ and $d_{k,l}$. The value of similarity falls in [0, 1]. The value of $d_{k,l}$ ranges from 0 to 24. For normalization, $d_{k,l}$ is divided by 24, as shown in Exp. (13). Therefore, The distance $d(H_k, H_l)$ is evaluated by the Exp. (13).

$$d(H_k, H_l) = \sqrt{(1 - \mathcal{S}(H_k, H_l))^2 + \left(\frac{d_{k,l}}{24}\right)^2} \quad (13)$$

The distance $d(U_x, U_y)$ between two histogram clusters U_x and U_y can be defined as the average distance of all possible pairs (H_k, H_l), where $H_k \in U_x$ and $H_l \in U_y$. That is,

$$d(U_x, U_y) = \frac{\sum_{\forall H_i \in U_x, \forall H_j \in U_y} d(H_i, H_j)}{|U_x| * |U_y|} \quad (14)$$

The following introduces the details of *merge procedure* of two clusters. Two clusters U_k and U_l that have minimal distance among all possible pairs will be merged into a new cluster $U_{h=(k,l)}$ in each round. That is, pair (U_k, U_l) will be merged if they satisfy the following merge criterion.

$$(U_k, U_l) = \arg \min_{\forall U_x, \, U_y} d(U_x, U_y) \quad (15)$$

Meanwhile, a tree corresponding to the merged pair, denoted by $T_{h=(k,l)}$, will be constructed. The value of the distance $d(U_k, U_l)$ will be regarded as the merging cost, denoted by $\lambda_{h=(k,l)}$. The calculation of distances and the merging procedure will be repeated executed until all clusters are merged into one. Finally, a hierarchical cluster tree can be obtained.

As shown in Fig. 4, each histogram $H_j \in \mathcal{R}_i$ is considered as a single cluster U_j which will be the leaf node. Assume that the $d(U_1, U_5)$ has the smallest value. Then, clusters U_1 and U_5 will be merged into cluster $U_{h=(1,5)}$, or $U_{(1,5)}$ in short, and the tree $T_{h=(1,5)}$ is constructed with the merging cost $\lambda_{h=(1,5)} = d(U_5, U_8)$. The merging procedure will be repeated executing until a hierarchical cluster tree is constructed.

In the second step, the *tree partition procedure* will be executed for partitioning the tree T_h into several clusters. Recall that each tree T_h has a merging cost λ_h. An cutoff threshold, denoted by Ω_{trh}, will be determined as a condition for terminating the tree partition procedure which is a repeated operation for partitioning each subtree obtained by the last round. Assume

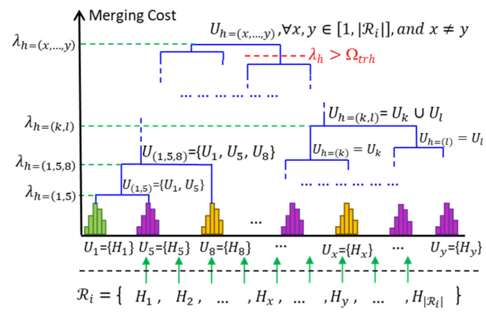


Fig. 4. The merging of clusters and the construction of a hierarchical tree.

that tree T_h is obtained by merging two subtrees T_k and T_l. In the first round, tree T_h will be partitioned into two subtrees T_k and T_l if the following partition criterion holds.

Partition Criterion of Tree T_h:

$$\lambda_h > \Omega_{trh}. \tag{16}$$

In the second round, each subtree of T_h, say T_k or T_l, will be further checked if the Partition Criterion is satisfied. If it is the case, the subtree will be further partitioned into two subtrees. The partition operation will be repeated executed round by round until the partition criterion failure.

After the execution of this task, we have obtained several partitioned trees. From the conceptual point of view, each partitioned tree, say T_k, represents a behavior b^i. This also indicates that all the leaf nodes in T_k are the sensor events which should be recognized as behavior b^i. The following explains this argument. Recall the construction of the hierarchical tree. Let $\mathfrak{C}_k = \{H_1, \ldots, H_{|\mathfrak{C}_k|}\}$ denote the set of all histograms of T_k. It means that these histograms have similar shapes. This also implies that cluster $\mathfrak{C}_k$ is a *behavior cluster*.

Herein, it is noticed that the two steps mentioned in this task should be applied to each $\mathcal{R}_i$ since each $\mathcal{R}_i$ represents the set of histograms in space $r_i \in \mathbb{R}$. Let $\mathbb{C} = \{\mathfrak{C}_1, \mathfrak{C}_2, \ldots, \mathfrak{C}_{|\mathbb{C}|}\}$ denote all possible *behavior clusters* obtained in this task for all possible space $r_i \in \mathbb{R}$. As shown in level 5 of Fig. 3, this task is implemented on each $\mathcal{R}_i \in \mathbb{H}$ and obtains the set $\mathbb{C}$ of *behavior clusters*. The remaining work for behavior recognition is to label each behavior cluster with behavior name, which will be presented in the next phase.

C. Behavior Matching Phase

This phase aims to label each behavior cluster with a behavior name. Assume that each behavior pattern $p_i = (s_{i,1}, s_{i,2}, \ldots s_{i,|p_i|})$ for each interested behavior has been predefined by engineers. Let $P = \{p_1, p_2, \ldots, p_{|\mathbb{B}|}\}$ denote the set of all predefined behavior patterns. Herein, we notice that the actual sensor pattern for each behavior of the elder might be different from the pattern predefined by the engineers. To be able to compare with the predefined pattern, an actual behavior pattern $\tilde{p}_j$ will be extracted from each *behavior cluster* $\mathfrak{C}_j \in \mathbb{C}$ obtained in the last phase. The histograms in each *behavior cluster* $\mathfrak{C}_j \in \mathbb{C}$ can be transferred to a sequence of sensor which are triggered by the behavior b_x. The sequence of sensors will be treated as an actual sensor pattern, denoted by notation $\tilde{p}_j = (s_{j,1}, s_{j,2}, \ldots s_{j,|\mathfrak{C}_j|})$.

Let each sensor $s_j \in \mathbb{S}$ be denoted by a unique character. Each pattern p_i or $\tilde{p}_j$ can be represented by a string. Let $LCS(p_i, \tilde{p}_j)$ denote the length of the longest common substring between two strings p_i and $\tilde{p}_j$. The distance $d(p_i, \tilde{p}_j)$ of strings p_i and $\tilde{p}_j$ can be measured by Exp. (17).

$$d(p_i, \tilde{p}_j) = 1 - \frac{LCS(p_i, \tilde{p}_j)}{\text{Min}(|p_i|, |\tilde{p}_j|)} \tag{17}$$

Finally, the behavior identification will be implemented based on the labeling criterion which will be presented in the following.

Labeling Criterion: The *behavior cluster* $\mathfrak{C}_j$, will be labeled with the behavior b^x of the predefined pattern p_x, only if the following two criteria are satisfied.

1) $p_x = \arg\min_{p_i \in P} d(p_i, \tilde{p}_j)$
2) $d(p_x, \tilde{p}_j) < \theta_{trh}$

where θ_{trh} denotes the threshold of minimal distance between p_x and $\tilde{p}_j$. If the *Labeling Criteria* are satisfied, *behavior cluster* $\mathfrak{C}_j$ will be labeled with behavior b^x of the predefined pattern p_x. Otherwise, *behavior cluster* $\mathfrak{C}_j$ will be not considered since it is not an interested one of the engineer's expectations in the designing phase.

TABLE II
SENSOR DATA FORMAT

SensorId	RegionId	StartTime	EndTime	Value
MG0104	05	20160114074512693	20160114074512693	0000000001
MG0104	05	20160114074624114	20160114074624114	0000000000
PR0103	02	20160114101624575	20160114101624575	0000000001

VI. SIMULATION

This section investigates the performance of the proposed *BIA* algorithm with different parameters and compares it against the MLAR algorithm in terms of the accuracy of behavior identification, which is measured by the number of False Positive (FP) and False Negative (FN). The following presents the data description and experimental results.

A. Data Description

This subsection gives the description of the sensor data set, named *WMNL2016*, which was collected in Wireless & Mobile Network Laboratory, Tamkang University. There are more than 20 state-change sensors and smart devices which were deployed at an apartment with 5 regions, including bathroom, foyer, living room, bedroom, and kitchen & dining room. Each sensor has a unique sensor ID and is located in a certain region with a unique region ID. Table II gives the sensor data format of *WMNL2016*. The 'SensorId' refers to the ID of sensor while the 'RegionId' refers to the ID of the different regions in the apartment. The 'StartTime' and 'EndTime' refer to the start time and end time of the sensor event. The 'Value' records the reading value of the sensor when the event occurs.

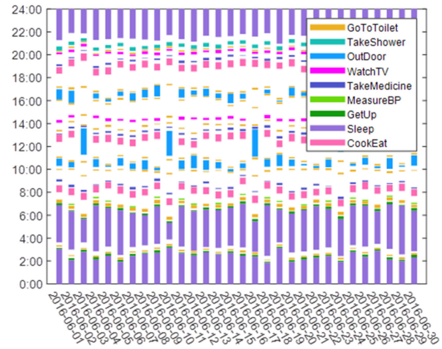


Fig. 5. The information of actual behaviors of a month.

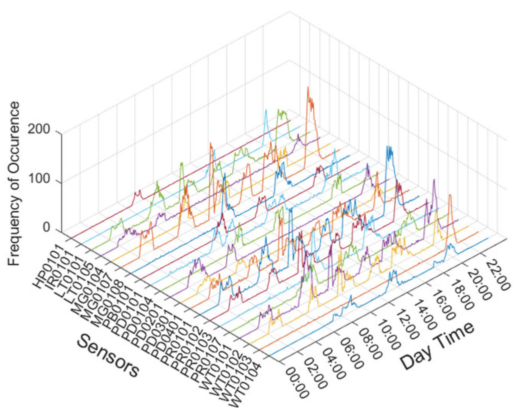


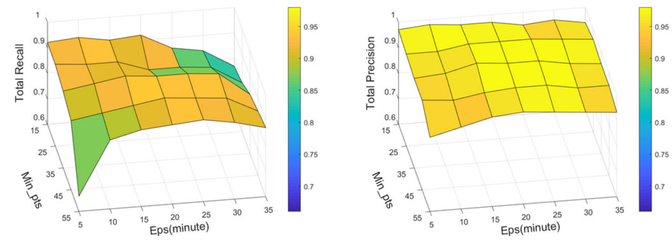
Fig. 6. The occurrence frequency of sensor events.

A volunteer lived in the apartment for more than 6 months was required to label the 9 interested behaviors by using microphone of the cellphone. The interested behaviors include 'GoToToilet', 'TakeShower', 'OutDoor', 'Watch TV', 'TakeMedicine', 'MeasureBloodPressue (MeasureBP, in short) ', 'GetUp', 'Sleep', and 'CookEat'. Fig. 5 depicts the interested behaviors labeled by the volunteer in June, 2016. As shown in Fig. 5, the volunteer behaved regularly. For instance, the volunteer got up between 6 and 7 o'clock and went out twice a day in most days. On each Saturday, the volunteer went out to have lunch with his children.

Fig. 6 shows the occurrence frequency of sensor events generated by different sensors in six months. The events of each sensor form a curve which likely indicates the regular occurrence of sensor events generated by this sensor during six months. For instance, the sensor with ID 'WT0104', deployed in the kitchen, was often triggered around 8, 13, and 19 o'clock. This also implies that the volunteer cooked and ate at the three time periods.

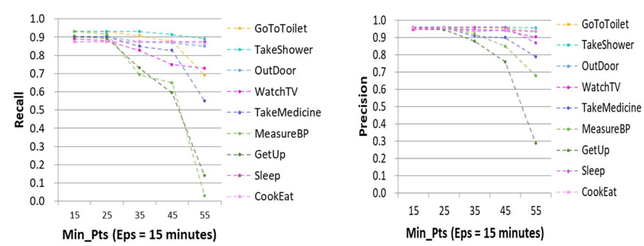
B. Experimental Results

Recall that the proposed BIA mainly consists of three phases. The *SEG* phase employs DBSCAN to cluster sensor events into *primitive groups*. Each *primitive group* will be a collection of sensor event data which are generated from the same sensor associated with the same action in approximate time. The different



(a) Total recall (b) Total precison

Fig. 7. The performance of the proposed BIA by varying *Eps* and min_pts.



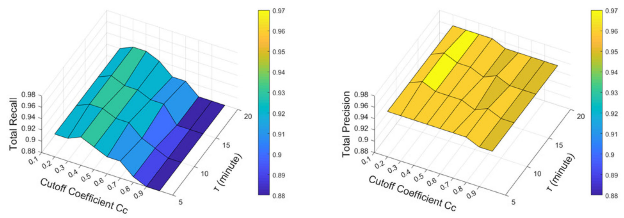
(a) Recall of individual behaviors (b). Precision of individual behaviors

Fig. 8. The performances of some interested behaviors, in terms of recall and precision.

values of parameters *min_pts* and *Eps* of DBSCAN on clustering performance will be discussed in the following.

Fig. 7 presents the impact of parameters *min_pts* and *Eps* of DBDACN in *SEG* phase on the performance in terms of total recall and total precision while the parameters in other phases are fixed. In general, there is a common trend that the experimental results, in terms of total recall and total precision, will decrease in two cases. The first case is that *min_pts* is set at a large value while the *Eps* is set at a small value. This occurs because that sensor events corresponding to the same behavior cannot be clustered in the same group. Hence the corresponding behavior cannot be identified. This result also implies that the number of FN increases and the values of recall and precision decrease. The second case is that the *min_pts* parameter is set at a small value while the *Eps* parameter is set at a large value. In this case, a large amount of sensor events will be clustered into a group, leading to a small number of event groups. This raises the problem that each event group consists of many sensor events which might be corresponding to different behaviors. Similar to the first case, the number of FN increases and the values of recall and precision decrease.

Fig. 8 further depicts the identification performance of individual behaviors in terms of recall and precision. The related parameter *Eps* is set at 15 minutes. In general, the values of recall and precision decrease with the value of *Min_pts*. This occurs because the sensor events which are corresponding to the same behavior cannot be clustered in one group and hence they are considered as noises when the value of *Min_pts* increases, leading to poor identification performances. In comparison, the performances of behaviors which might occur frequently, such as 'GoToToilet', or regularly, such as 'TakeShower', are better



(a) Total recall (b) Total precision

Fig. 9. The performance of the proposed BIA, in terms of total recall and total precision by varying cutoff coefficient C_c and class interval τ.

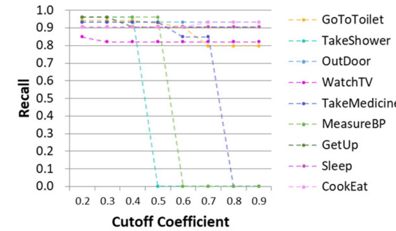


Fig. 10. The performance, in terms of recall, of some interested behaviors.

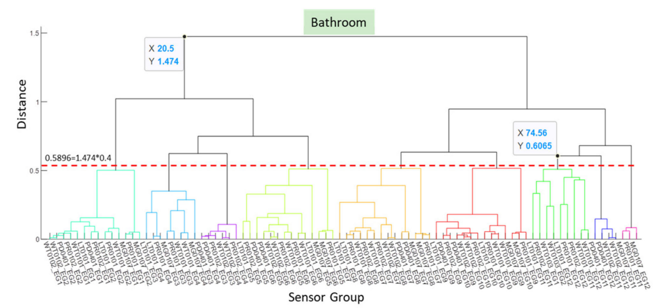


Fig. 11. The cluster tree of the region bathroom.

than the performances of behaviors which occur infrequently or unregularly, such as 'MeasureBP'. In addition, the identification performances of behavior 'GetUp', in terms of recall and precision, are worse than other behaviors. It occurs because that the sensors triggered by behavior 'GetUp' are almost identical to those triggered by behavior 'Sleep'. It is difficult to identify behavior 'GetUp', especially when the time interval between the occurrences of 'GetUp' and 'Sleep' is small.

Recall that the second phase, namely *HSG*, constructs the histogram for each *primitive group* and builds up several *behavior clusters* by applying Hierarchical Clustering algorithm on the histograms of *primitive groups*. The construction of the histogram highly depends on the value of class interval τ which further impacts the calculation of distance between two histograms. In addition, the parameter cutoff threshold Ω_{trh} of Hierachical Clustering scheme directly influences the results of the second phase.

The following investigates the impacts of the class interval τ and cutoff threshold Ω_{trh} in *HSG* phase on the performance of the proposed BIA. Recall that a hierarchical tree will be constructed and then partitioned by the cutoff threshold Ω_{trh} for each region. Considering that the behaviors in different areas might have differences in regularity, the cutoff threshold Ω_{trh} will be set at different values in different regions. Recall that notation $\lambda_{h=(k,l)}$ denotes the merging cost of two clusters U_k and U_l, and is measured by the distance $d(U_k, U_l)$. Let λ_H^j denote the distance between the last two clusters to be merged in area r_j. The parameter cutoff coefficient, denoted by C_c, is introduced to discuss the impacts of cutoff threshold Ω_{trh} which is set according to the equation $\Omega_{trh} = C_c * \lambda_H^j$.

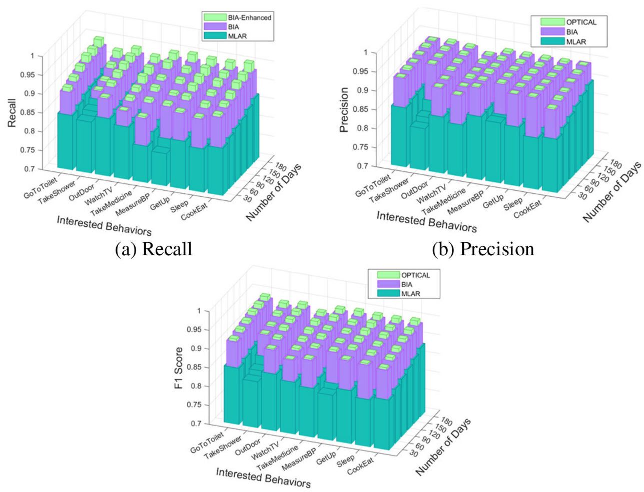
Fig. 9 presents the impacts of class interval τ and cutoff coefficient C_c in *HSG* phase, on the performance in terms of total recall and total precision while the parameters in other phases are fixed. With fixed value of class interval τ, there is a common trend that the total recall increases first and then decreases with the value of cutoff coefficient, as shown in Fig. 9(a). This occurs because *primitive groups* will not be grouped together when the cutoff coefficient is set at a small value while a large number of *primitive groups* will be clustered together when the cutoff coefficient is set at a large value. Both cases will increase the number of FN and hence decrease the performance in terms of total accuracy and total recall. In compassion, the performance of total precision drops slowly when the cutoff coefficient is set at a large value, as shown in Fig. 9(b). It occurs because that the number of FP is small in most cases. According to the experiments, the class interval τ and cutoff coefficient C_c are suggested to be set at the values ranging from 5 to 15 minutes and 0.3 to 0.5, respectively.

Fig. 10 further observes the recall of each interested behavior by varying the cutoff coefficient. The class interval τ is set at 10 minutes. In general, the performance of recall decreases with the values of cutoff coefficient. In particular, the performances, in terms of recall, of some behaviors, such as 'TakeShower' and 'MeasureBP', drop quickly when the cutoff coefficient is set at a large value. It occurs because these behaviors don't occur frequently in each day.

Fig. 11 presents the cluster tree of the bathroom region, which is obtained after the execution of the *HSG* phase. The class interval τ is set at 10 minutes and the cutoff coefficient is set at 0.4. As shown in Fig. 11, the dimension 'Distance' refers to the distance between two trees which are merged. Take the point (X 20.5, Y 1.474) for example, the value of Y is the distance between the trees T_{h1} and T_{h2}. The other dimension 'Sensor Group' refers to the event groups which are obtained after the execution of *SEG* phase. Take 'WT0102_EG1' for example, events in 'WT0102_EG1' are generated by sensor 'WT0102', where EG1' is a unique serial number.

Recall that the proposed BIA adopts DBSCAN to group sensor events in *SEG* Phase. As shown in Fig. 6, it can be observed that the densities of events are not balanced. In such a situation, DBSCAN has difficulty to identify all clusters [21]. Therefore, the BIA-Enhanced is proposed, which replaces the DBSCAN with the OPTICS, aiming to improve the event grouping result in *SEG* phase, as compared with the BIA.

Fig. 12 further compares the performances of BIA-Enhanced, BIA, and MLAR in terms of recall, precision, and F1 Score. The



(a) Recall (b) Precision

(c) F1 Score

Fig. 12. Performance of three compared algorithms in terms of recall, precision, and F1 Score.

F1 Score is defined by the following equation.

$$\text{F1 Score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

Herein, we notice that the setting of *Min_pts* should refer to the number of data. More specifically, the value of *Min_pts* is set at the value of one sixth of the number of data. For instance, the *Min_pts* is set at value $30/6 = 5$ if the sensor data are collected from only one month. The other parameters settings are Eps $=$ 15 minutes, $\tau = 10$ minutes, and $C_c = 0.4$. In comparison, the proposed BIA-Enhanced and BIA outperform MLAR in term of recall, precision and F1 Score. This occurs because that the proposed algorithms take multiple properties of behaviors into account. Besides, the MLAR adopts string alignment and rule-based examination to identify behaviors which cannot employ fuzziness, leading to a large number of FN.

VII. Conclusion

This paper proposes a behavior identification algorithm, called *BIA,* which aims to identify daily behaviors to support the homecare of the elderly living alone. The properties of behaviors, including *Event Shift* and *Histogram Shape Similarity Properties* are presented based on the *Time Regularity Observation* of elder behaviors. The proposed *BIA* consists of three phases, named *SEG, HSG* and *BM*. The *SEG* phase mainly constructs a number of disjoint sensor primitive groups using DBSCAN, based on the *Time Regularity Observation*. Then the *HSG* phase clusters these *primitive groups* into several *behavior clusters*, based on the *Event Shift* and *Histogram Shape Similarity Properties*. Finally, the *BM* phase labels each behavior cluster with a behavior name. Based on the relations between histograms, the proposed *BIA* takes into account the statistical property of sensor event data and provides elasticity and flexibility in behavior recognition by utilizing unsupervised clustering algorithms. Extensive experiments show that *BIA* outperforms the compared algorithm in terms of the identification accuracy. Our future work will further analyze and predict the abnormal behaviors based on the elderly behavior model.

References

[1] F. Zhou, J. Jiao, S. Chen, and D. Zhang, “A case-driven ambient intelligence system for elderly in-home assistance applications,” *IEEE Trans. Syst., Man, Cybern.*, vol. 41, no. 2, pp. 179–189, Mar. 2011.

[2] J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, “From activity recognition to intention recognition for assisted living within smart homes,” *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 368–379, Jun. 2017.

[3] D. Naranjo-Hernández, L. M. Roa, J. Reina-Tosina, and M. Á. Estudillo-Valderrama, “SoM: A smart sensor for human activity monitoring and assisted healthy ageing,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 11, pp. 3177–3184, Nov. 2012.

[4] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-based activity recognition,” *IEEE Trans. Syst., Man, Cybern.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[5] C. Zhu and W. Sheng, “Realtime recognition of complex human daily activities using human motion and location data,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 9, pp. 2422–2430, Sep. 2012.

[6] Y. Chen, L. Yu, K. Ota, and M. Dong, “Robust activity recognition for aging society,” *IEEE J. Biomed. Health Inform.*, vol. 22, no. 6, pp. 1754–1764, Nov. 2018.

[7] F. Osayamwen and J.-R. Tapamo, “Deep learning class discrimination based on prior probability for human activity recognition,” *IEEE Access*, vol. 7, pp. 14747–14746, Jan. 2019.

[8] Y. Yan, E. Ricci, G. Liu, and N. Sebe, “Egocentric daily activity recognition via multitask clustering,” *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 2984–2995, Oct. 2015.

[9] M. Wang, C. Luo, B. Ni, J. Yuan, J. Wang, and S. Yan, “First-person daily activity recognition with manipulated object proposals and non-linear feature fusion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2946–2955, Oct. 2018.

[10] W. Xu, Z. Miao, X.-P. Zhang, and Y. Tian, “A hierarchical spatio-temporal model for human activity recognition,” *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1494–1509, Jul. 2017.

[11] G. Valenza *et al.*, “Wearable monitoring for mood recognition in bipolar disorder based on history-dependent long-term heart rate variability analysis,” *IEEE J. Biomed. Health Inform.*, vol. 18, no. 5, pp. 1625–1635, Sep. 2014.

[12] Nazábal, P. García-Moreno, A. Artés-Rodríguez, and Z. Ghahramani, “Human activity recognition by combining a small number of classifiers,” *IEEE J. Biomed. Health Inform.*, vol. 20, no. 5, pp. 1342–1351, Sep. 2016.

[13] D. Tao, L. Jin, Y. Yuan, and Y. Xue, “Ensemble manifold rank preserving for acceleration-based human activity recognition,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1392–1404, Jun. 2016.

[14] S. Zhang, S. I. McClean, and B. W. Scotney, “Probabilistic learning from incomplete data for recognition of activities of daily living in smart homes,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 3, pp. 454–462, May 2012.

[15] D. J. Cook, N. C. Krishnan, and P. Rashidi, “Activity discovery and activity recognition: A new partnership,” *IEEE Trans. Cybern*, vol. 43, no. 3, pp. 820–828, Jun. 2013.

[16] L. Peng, L. Chen, X. Wu, H. Guo, and G. Chen, “Hierarchical complex activity representation and recognition using topic model and classifier-level fusion,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 6, pp. 1369–1379, Jun. 2017.

[17] P. Gupta and T. Dallas, “Feature selection and activity recognition system using a single triaxial accelerometer,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, Jun. 2014.

[18] T. Magherini, A. Fantechi, C. D. Nugent, and E. Vicario, “Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living,” *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 6, pp. 509–521, Nov. 2013.

[19] D. Cook and M. Schmitter-Edgecombe, “Assessing the quality of activities in a smart environment,” *Methods Inf. Med.*, vol. 48, no. 5, pp. 480–485, May 2009.

[20] H. Chen, S. Zhao, C. Shang, G. Chen, and C. Y. Chang, “Activity recognition approach based on spatial-temporal constraints for aged-care in smart home, “ *Int. J. Ad Hoc Ubiquitous Comput.*, to be published.

[21] Y. Zhu, K. Ming Ting, and M. Carman, “Density-ratio based clustering for discovering clusters with varying densities,” *Pattern Recognit.*, vol. 60, pp. 983–997, Dec. 2016.