Activity recognition approach based on spatial-temporal constraints for aged-care in smart home

Haibao Chen

School of Computer and Information Engineering,
Chuzhou University,
Anhui, China
Email: chb@chzu.edu.cn

Chih-Yung Chang*

Department Computer Science and Information Engineering,
Tamkang University, Taiwan
Email: cychang@mail.tku.edu.tw
*Corresponding author

Cuijuan Shang, Guilin Chen and Shenghui Zhao

School of Computer and Information Engineering,
Chuzhou University,
Anhui, China
Email: shangcuijuan@chzu.edu.cn
Email: glchen@chzu.edu.cn
Email: zsh@chzu.edu.cn

Abstract: Activity recognition plays an important role in smart homes for aged-care. In this paper, we formulate the problem of activity recognition and propose a new method based on spatial-temporal constraints to carry out activity recognition, which consists of five phases: *initialisation*, *segmentation*, *sensor data representation*, *activity exploration* as well as *activity identification*. Besides, we analyse the time complexity and space complexity of our approach in theory. To evaluate our approach, we carried out experiments on real dataset from Wireless and Mobile Network Laboratory, Tamkang University. The experimental results demonstrate an improvement of 5.6% in the accuracy on average of recognised activities in comparison to the method of support vector machine (SVM).

Keywords: activity recognition; smart home; wireless sensor network; aged-care.

Reference to this paper should be made as follows: Chen, H., Chang, C-Y., Shang, C., Chen, G. and Zhao, S. (2020) 'Activity recognition approach based on spatial-temporal constraints for aged-care in smart home', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 33, No. 3, pp.168–183.

Biographical notes: Haibao Chen is currently serving as the Director of Department of Network and Communication Engineering at the Chuzhou University in China. His research interests include WSN and IoT for aged-care.

Chih-Yung Chang received his PhD in Computer Science and Information Engineering from the National Central University, Taiwan in 1995. He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. His current research interests include internet of things, wireless sensor networks, artificial intelligence and deep learning. He has served as an Associate Guest Editor for several SCI-indexed journals, including *IJAHUC* from 2011 to 2014, *IJDSN* from 2012 to 2014, *IET Communications* in 2011, *TS* in 2010, *JISE* in 2008, and *JIT* from 2004 to 2008.

Cuijuan Shang received his BS in Biomedical Engineering from the Zhengzhou University, Zhengzhou, China in 2008, and MS in Biomedical Engineering from the Shandong University, Jinan, China in 2011. She is currently a Lecturer with the School of Computer and Information Engineering, Chuzhou University, Chuzhou, Anhui, China, and PhD student from 2018 with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. Her current research interests include wireless sensor network and healthcare.

Guilin Chen received his BS from the Anhui Normal University, China in 1985, and MS from the Hefei University of Technology in 2007. He is currently a Professor and the Scholar Head of the School of Computer and Information Engineering at Chuzhou University, Anhui, China. His current research interests include wireless networks, healthcare and internet of things.

Shenghui Zhao received her PhD from the Southeast University of China in 2013. Currently, her interests include web service, cloud computing, big data and internet of things. She is a Professor at the School of Computer and Information Engineering and Head of the research group of cloud computing and big data, Chuzhou University in China.

1 Introduction

Aging are becoming a problem around the world. For example, by 2030 there will be about 20% of total population in Australia over the age of 65 (Hugo and Sa, 2014). In such situation, the healthcare expenditure for the growing number of older people poses challenge to the governments and societies. The significant improvement in the technologies of wireless sensors, especially low-power, low-cost, high-capacity, and miniaturised sensors, and internet of things has paved the way towards realising continuous and cost-effective monitoring services in smart homes, which is a promising solution to provide smart and efficient healthcare while dealing with this challenge (Amiribesheli et al., 2015).

In smart homes, a large amount of data can be continuously collected from sensors triggered by elderly, which can be used to recognise the activities of daily living (ADL). Furthermore, the recognised activities can be exploited to assess the cognitive and physical well-being of elderly, which will assist elderly person to live longer independently, with a better quality of life, in their own homes.

Recognising ADL are now a hot topic of research. In recent years, researchers have proposed many approaches to deal with the activity recognition problem based on different kinds of devices from different perspectives, most of which are based on cameras (Donahue et al., 2015; Pal and Abhayaratne, 2015; Jalal et al., 2017), wearable sensors (Lara and Labrador, 2013; Parkka et al., 2006; Mukhopadhyay, 2015), and WiFi (Wang et al., 2015; Abdelnasser et al., 2015; Ali et al., 2015). However, camera-based approaches potentially result in the invasion of inhabitant's privacy. Wearable sensors-based approaches are inconvenient because of the sensors that users have to wear every day. Although WiFi-based approaches do not need inhabitant to wear any devices, they still have room for improvement, for example, exploring any further optimisation to the location independence.

Binary sensors, such as pressure sensor, water flow sensor, ultrasonic sensor, infrared sensors, and gas sensors, are commonly deployed in smart homes. Previous work have shown the solid potential of binary sensors for solving the problem of activities recognition in the smart home (Tapia et al., 2004), and can be used in human-centric problems, e.g., health and elderly care (Wilson and Atkeson, 2005; Cook and Schmitter-Edgecombe, 2009). For example, Van Kasteren et al. (2008) demonstrated how to use binary sensors and motion sensors, to recognise ADLs of people living on their own home, where the binary sensors are used to measure the opening or closing of doors and cupboards, the use of electric appliances. The advantages of binary sensors-based approaches (Ordónez at el., 2013) include the low cost and ease of device deployment, as well as the least privacy invasion compared to cameras and wearable sensors.

Human activity is an abstract concept, therefore not only space but also time information must be taken into consideration. The existing research work based on binary sensors (Ordónez at el., 2013) often took activity recognition as the problems of continuous 'sequential data classification', which overlooked the associations of time and space among the sensing data. To overcome these issues, in this paper, we propose an approach based on time and space constraints to recognise activities effectively and efficiently by handling data from monitoring continuously with low-cost binary sensors, without the deployment of extra specialised or costly sensing equipment. Experimental results show that our approach works well, which will largely increase the opportunity for wide deployment and in-home use.

The main contributions of this work are listed as follows.

- Based on the in-deep analysis of human activities in home, we formalise the problem of activity recognition with binary sensors deployed in smart home.
- We present the idea of about sensor data partition based on space and time constraints, to simplify the computational complexity.
- We propose a method consisting of five phases: initialisation, segmentation, sensor data representation, activity exploration as well as activity identification, to recognise the human activities.

The rest of this paper is organised as follows. Section 2 summarises related work. In Section 3, activity recognition approach based on binary sensors for home care is introduced, which is followed by the experiments in Section 4. At last, a conclusion about this work is presented in Section 5.

2 Related work

Human activities recognition has been widely explored, because it is important in pervasive environments. However, there still exist many challenges of activities recognition far from being solved well. In the following, we present an overview of the existing works and point out their shortcomings.

Recently, a large number of approaches for human activity recognition have been introduced for different settings, which can be roughly divided into categories of vision-based, sensor-based, and radio frequency-based activity recognition. The former category depends on the utilisation of visual sensing devices, e.g., video cameras, to monitor the activities of occupant and the changes of environment in smart homes. The produced sensing data are video sequences or digitised visual data. Approaches of vision-based activity recognition usually make use of the computer vision techniques for recognising activity pattern. The second category is based on the observation that different activities will occur different multi-path distortions in radio signals. The third one rests on the technologies of wireless sensor network to monitor human activity in smart homes.

2.1 Vision-based activity recognition approach

Vision-based methods (Matsuo et al., 2014; Rodriguez et al., 2016; Jalal et al., 2014) exploit video cameras to carry out the activity recognition from video sequences, which is an important research field of computer vision. Generally, activity recognition from video sequences consists of several phases, including pre-processing of images or space-time volume video data, feature extraction with respect to human activity, and activity modelling based on the extracted features.

Although vision-based methods have made positive contributions in the field of recognising human activity, they still have shortcomings. First, the privacy is a matter of concern for inhabitants, because most people are not inclined to be monitored and recorded by video cameras. Second, inhabitants need to stay within the spaces of smart homes pre-determined by fixed cameras. Third, handling images and videos require a significant amount of computing resources. Therefore, these vision-based approaches are not applicable to activity recognition in smart homes. In contrast, the advantages of sensor-based activity recognition are wide range of applications, non-intrusiveness, and the like.

2.2 Radio frequency-based methods

Radio signals-based methods have been proposed to recognise human activities (Liu et al., 2011; Wei et al., 2015; Al-Qaness et al., 2016). Kim and Ling (2009) used features from radar spectrogram to recognise human activities, e.g., running, walking, walking while holding a stick, crawling, boxing while moving forward, boxing while standing in place, and sitting still. Jokanović and Amin (2018) proposed a method to implement fall detection using deep learning in range-Doppler radars. Besides, WiFi-based approaches (Zheng et al., 2017), e.g., WiSee (Pu et al., 2013) and E-eye (Wang et al., 2014), also have been introduced based on the observation that different human activities occur different multi-path distortions in WiFi signals. However, radio frequency-based activity recognition still have some limitations, e.g., signals are susceptible to interference in smart home.

2.3 Sensor-based activity recognition approach

In this section, we roughly divide the existing work into three aspects: wearable sensors, object sensors, radio signals, and ambient sensors-based methods.

2.3.1 Wearable sensors-based methods

With wearable sensor-based methods, most of the measured attributes are in relation to the inhabitant's movement (Pansiot et al., 2007). The sensors which are used to collect data about the above attributes, are worn by inhabitants, e.g., accelerometer, physiological sensors, global positioning system (GPS) receivers, magnetometer, and gyroscope, which are usually equipped on watches (Bhattacharya and Lane, 2016) and smart phones (Sankaran et al., 2014; Kwapisz et al., 2011; Eftekhari and Ghatee, 2016; Montoya et al., 2015; Reyes-Ortiz et al., 2016).

Among those research based on wearable sensors, the accelerometers is mostly adopted. In addition, gyroscope and magnetometer are also exploited together with the accelerometer. Wearable sensors need to be attached to inhabitants, which result in lots of discomfort to inhabitants and does not meet the requirements of practical applications in smart homes.

2.3.2 Object sensor-based methods

This kind of method needs to attach sensors to objects (Li et al., 2015, 2016; Buettner et al., 2009; Lien et al., 2016), to monitor the changes in position of a target object, and rests on the assumption that human activities can be inferred from object sequences manipulated by inhabitant in smart home. For example, the accelerometer attached to a cup can be used to detect the drinking water activity, and cooking-related activities often contain picking up and putting down dishes. In comparison with wearable sensors, object sensors are difficult to deploy in smart home. Besides, object sensors do not work when recognising activities, which require no contact with the object (e.g., taking shower).

2.3.3 Binary sensors-based methods

With the development of wireless sensor technology and the cost reduction of production, more and more sensors are employed in smart homes, including sound sensors, pressure sensors, and temperature sensors, and so on and so forth.

The differences between object sensors and binary sensors are the former ones are used to measure the movements of objects, while the later ones can capture the changes of the smart home environment, e.g., the opening or closing of doors and cupboards, the use of water, gas, and electric appliances, as well as the entering rooms.

Researchers can take advantage of binary sensors for capturing interaction between inhabitants and the smart home environment, to recognise human activities. These kind of methods usually extract time-space feature from sensing data involves using artificial space features as well as fixed-length time windows to recognise human daily activities (Van Kasteren et al., 2008; Ordónez et al., 2013). In fact, artificial features have the limits of generalisation, and fixed-length time windows are not suitable for activities with different durations, which is common in real world.

Different from existing work, in this paper, we introduce an approach according to time and space constraints, which can recognise activities effectively and efficiently by handling sensing data from binary sensors commonly deployed in smart homes.

3 The proposed approach

The proposed approach mainly consists of five phases: the initialisation, segmentation, sensor data representation, and activity exploration as well as activity identification phases. The initialisation phase mainly checks the correctness and completeness of the raw data and then filters all invalid data. Then, the segmentation phase pushes the computational complexity of activity exploration into smaller one through partitioning data to reduce the search space of activity recognition process. From the logic point of view, the sensor data of each day has been partitioned into several sets of data. After that, the phase of sensor data representation will transfer the presentation of data to another one, aiming to map each sensor data to a character for simplifying the operations of later phase in advance. As a result, the sensor data of each day can be represented as a set of strings. Then, the activity exploration aims to explore the potential activities from the string representation. At last, the activity identification phase compares the explored potential activities with the predefined activity patterns. The following presents the details of each phase.

3.1 Phase 1: initialisation

Given a set of data, there probably exists invalid elements. If we ignore the particularity of these invalid elements and use them directly for processing, wrong results might be obtained. This phase aims to process the raw data collected from wireless sensors by filtering out the invalid data. Several conditions should be checked in this phase. For example, some data might have null value. Furthermore, the data generating time might be out of the lower and upper bounds. In addition, the sensor id in data is not able to be correctly verified.

3.2 Phase 2: segmentation

This phase aims to reduce the computational complexity for the later phases. Herein, we notice that the activity recognition for elder takes much time since there might be thousands of sensing data collected by sensors in one day. Since the time complexity for matching activity pattern highly depends on the length of each pattern and the number of patterns, reducing the length of each activity pattern can significantly reduce the time complexity and hence simplifies the operations designed in the later phases.

The main concepts behind in the segmentation are the space and time constraints. Let the space constraint refers to the property that the set of sensing data detected for each elder activity can be occurred only in a single space while the time constraint verifies the property that the occurrence times for any two consecutive sensing data cannot differ more than a certain time period. Based on the space and time constraints, the big set of sensing data collected in one day can be further partitioned into a set of smaller subsets. The following presents the operations designed in this phase, which consists of two parts:

1 operations based on space constraints

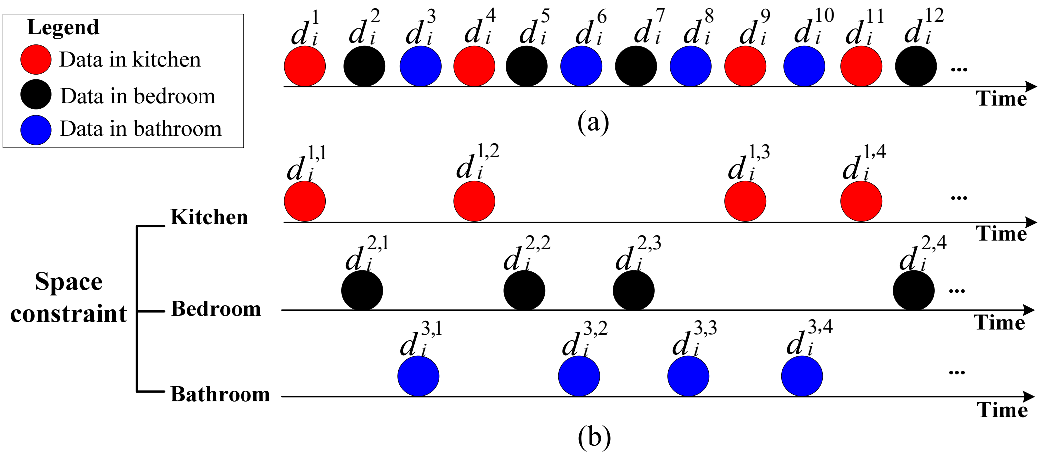
2 operations based on time constraints.

3.2.1 Step 1: space partition based on space constraints

Let D^i denotes the set of data collected in the i^{th} day, where $D^i = \{d_i^1, d_i^2, ..., d_i^{\beta_i}\}$, and β_i is the number of elements in D^i. As mentioned in Section 2, the home space L is consisted by a set of non-overlapped subspaces, denoted as $L = \{L_1, L_2, \ldots, L_a\}$, where a is the number of spaces in a house. Based on the space constraint, the data in D^i can be partitioned into a number of non-overlapped subsets according to the location information given in each $d_i^k \in D^i$. After processing according to space constraints, D^i can be handled into a collection of non-overlapped subsets, i.e., $D^i = \{D^{i,1}, D^{i,2}, D^{i,3}, \ldots, D^{i,a}\}$, where $D^{i,j}$ ($1 \leq j \leq a$) denotes the collection of data ordered by time from the j^{th} space in the i^{th} day. Furthermore, $D^{i,j} = \{d_i^{j,1}, d_i^{j,2}, ..., d_i^{j,\beta_{i,j}}\}$ and $\beta_{i,j}$ is the number of data in $D^{i,j}$ based on the place constraint. Take $d_i^{j,k}$ as an example, it means the k^{th} sensor data from the j^{th} space in the i^{th} day.

Figure 1 gives an example to illustrate the process of space partition. As shown in Figure 1(a), there are a number of sensor data collected in the i^{th} day. Each circle represents a sensor data ordered by the time when data generated. Circles with different colours represent the sensor data are collected from different spaces. For example, red circles represent the sensor data collected in kitchen while blue circles represent the sensor data collected in bathroom. These data can be represented by $D^i = \{d_i^1, d_i^2, ..., d_i^{12}\}$.

Figure 1 An example of space constraints, (a) sensor data in i^{th} day before processed (b) sensor data in i^{th} day after processed according to space constraint (see online version for colours)



Herein, we notice that the sensor data expressed by red colours might be collected from different sensors. For example, the four circles represent that the first two data are collected by water flow sensor but the last two red circles represent the data collected by gas sensor. Figure 1(b) depicts the results obtained by applying the space partition step. As shown in Figure 1(b), the circles with different colours have been partitioned into different groups. Let $D^{i,1}$, $D^{i,2}$ and $D^{i,3}$ denote three different groups, which contain data collected in kitchen, bedroom and bathroom, respectively. The partitioned results obtained by applying space partition step can be expressed by:

$$D^{i,1} = \left\{d_i^1, d_i^4, d_i^9, d_i^{11}\right\} = \left\{d_i^{1,1}, d_i^{1,2}, d_i^{1,3}, d_i^{1,4}\right\},$$
$$D^{i,2} = \left\{d_i^2, d_i^5, d_i^7, d_i^{12}\right\} = \left\{d_i^{2,1}, d_i^{2,2}, d_i^{2,3}, d_i^{2,4}\right\},$$
$$D^{i,3} = \left\{d_i^3, d_i^6, d_i^8, d_i^{10}\right\} = \left\{d_i^{3,1}, d_i^{3,2}, d_i^{3,3}, d_i^{3,4}\right\}.$$

As shown in the expressions of $D^{i,1}$, $D^{i,2}$ and $D^{i,3}$, to represent that the first data is the first element in the first group $D^{i,1}$, we rename d_i^1 as a two dimensional notation $d_i^{1,1}$. The same operation goes for the second element in the first group, as well as elements in other two groups.

Since each activity can be occurred in one space, the space partition step guarantees that the sensor data associated with one activity will not be separated into two groups. That is, the space partition will not destroy the relationship among data but can reduce the computational time complexity for later phase.

3.2.2 Step 2: time partition based on time constraints

This phase mainly partitions the data in advance according to time constraint. Here we still take dataset $D^{i,j}$, which has been mentioned in step 1, as an example. Based on the time constraint, the data in $D^{i,j}$ can be further partitioned into a number of subsets according to the time information given in each $d_i^{j,k} \in D^{i,j}$. Let the time constraint ask the time difference of two consecutive sensor data cannot exceed the predefined threshold λ. Let $D^{i,j,k}$ denote the k^{th} partitioned group of $D^{i,j}$. The following illustrates how to partition the dataset $D^{i,j}$ into a number of groups. Starting from the first two data in set $D^{i,j}$. Initially, we include the first data in $D^{i,j}$ in the first group, denoted as $D^{i,j,1}$. To represent that the first data is the first element in the first group $D^{i,j,1}$, we rename $d_i^{j,1}$ three dimensional notation $d_i^{j,1,1}$. Then, we will check whether or not the second data in $D^{i,j}$ should be included in the first group. To achieve this, we further check if the following condition holds.

$$d_i^{j,2} \cdot time - d_i^{j,1} \cdot time \geq \lambda$$

If it is the case, based on time constraint, we will create the second group, denoted as $D^{i,j,2}$, and include the second data $d_i^{j,2}$ in the second group $D^{i,j,2}$. To represent that $d_i^{j,2}$ is the first element in the second group, we rename the $d_i^{j,2}$ as $d_i^{j,2,1}$. Otherwise, the second data should be included in the first group $D^{i,j,1}$ and rename the $d_i^{j,2}$ as a three dimensional notation $d_i^{j,1,2}$ since it is the second data in the first group $D^{i,j,1}$. This also indicates that the first two data might belong to the same activity. The above-mentioned operation will be repeatedly applied for each consecutive two data: $d_i^{j,k}$ and $d_i^{j,k+1}$ for $1 \leq k \leq \beta_{i,j}$.

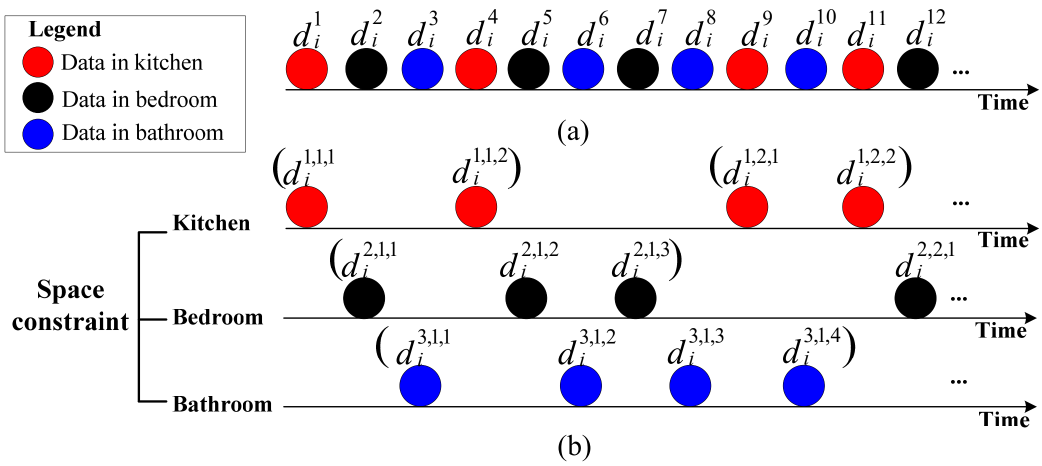
Then the time partition phase can be finished and finally we have a number of partitioned groups:

$$D^{i,j} = \left\{D^{i,j,1}, D^{i,j,2}, ..., D^{i,j,\xi_{i,j}}\right\} \text{ and}$$
$$D^{i,j,k} = \left\{d_i^{j,k,1}, d_i^{j,k,2}, ..., d_i^{j,k,\beta_{i,j,k}}\right\}$$

where $\xi_{i,j}$ denotes the number of groups in $D^{i,j}$ and $\beta_{i,j,k}$ is the number of elements in $D^{i,j,k}$.

Figure 2 continues the results of Figure 1 and further gives an example to demonstrate the process of time partition.

Figure 2 An example of time constraints and space constraints, (a) sensor data in i^{th} day before processed (b) sensor data in i^{th} day after processed according to space constraint (see online version for colours)



As shown in Figure 2, the dataset in each space (i.e., kitchen, bedroom and bathroom) is divided into different subsets according to time constraint, which are separated by parentheses. For example, the dataset in kitchen (i.e., $D^{i,1}$) is divided into two non-overlapped subsets by applying time partition step:

$$D^{i,1} = \{D^{i,1,1}, D^{i,1,2}\}$$

where

$$D^{i,1,1} = \{d_i^1, d_i^4\} = \{d_i^{1,1}, d_i^{1,2}\} = \{d_i^{1,1,1}, d_i^{1,1,2}\}$$
$$D^{i,1,2} = \{d_i^9, d_i^{11}\} = \{d_i^{1,3}, d_i^{1,4}\} = \{d_i^{1,2,1}, d_i^{1,2,2}\}$$

3.3 Phase 3: sensor data representation

For each segmentation (a set of sensor data ordered by time subjecting to space and time constraints defined in Section 2) created in phase 2, this phase aims to transfer each sensor data to a character for simplifying the operations of later phase in advance. The basic idea behind this phase is to transfer the multidimensional data to a single dimensional data, to facilitate the activity exploration.

Specifically, there are two tasks designed in this phase. First, the mapping task aims to map each sensor ID to a unique character. Second, the transferring task aims to transfer the five-dimentional data into one-dimentaional data. Let each data can be presented as five-tuple form (*id*, *name*, *loc*, *time*, *value*). The location and time information have been utilised for space and time partitions in phase 2 and the value has been utilised for checking the validation of the data in phase 1. The mapping task and transferring task can remove the information, which have been utilised in the previous phases for simplifying the data representation.

Let $C = \{c_i \mid 1 \le i \le n\}$ denote the set of n unique characters, and *transfer*(*Object o*) denote the transforming function of from sensor data to characters according to the sensor id.

Table 1 gives an example to transform each sensor data to a unique character c_i based on the sensor id of data.

Take $D^{i,j,k} = \{d_i^{j,k,w} \mid 1 \le w \le \beta_{i,j,k}\}$ as an example, which is a set of sensor data, i.e., a segmentation generated by phase 3 according to space and time constraints. After the process of this phase, all data in:

$$D^{i,j,k} = \{d_i^{j,k,w} \mid 1 \le w \le \beta_{i,j,k}\}$$

can be transferred to a string, expressed as:

$$transfer(D^{i,j,k}) = S^{i,j,k} = \{c_i^{j,k,w} \mid 1 \le w \le \beta_{i,j,k}\}$$

where $S^{i,j,k}$ consists of $\beta_{i,j,k}$ characters, and the order of sensor data in $D^{i,j,k}$ is also remained in $S^{i,j,k}$.

Table 1 A simple example of mapping table

Sensor ID	*Character*
1	c_1
2	c_2
3	c_3
4	c_4
…	…
n	c_n

Following the example of Figure 2, as shown in step 2 of phase 2, where:

$$D^{i,1,1} = \{d_i^{1,1,1}, d_i^{1,1,2}\}$$

Assume the ids of $d_i^{1,1,1}$ and $d_i^{1,1,2}$ in $D^{i,1,1}$ are 1 and 2, respectively. That is:

$$d_i^{1,1,1} \cdot \mathrm{id} = 1$$
$$d_i^{1,1,2} \cdot \mathrm{id} = 4$$

Since $d_i^{1,1,1}$ and $d_i^{1,1,2}$ will be mapped to c_1 and c_4, respectively, according to the mapping rule between sensor id and character shown in Table 1, $D^{i,1,1}$ can be represented as:

$$transform(D^{i,1,1}) = S^{i,1,1} = \{c_1, c_4\}$$

3.4 Phase 4: activity exploration

Consider any string, say:

$$S^{i,j,k} = \{c_i^{j,k,w} \mid 1 \le w \le \beta_{i,j,k}\}$$

which is obtained from the previous phase. Herein, we notice that $S^{i,j,k}$ might contain more than one activity. Therefore, each substring of $S^{i,j,k}$ might be a primitive activity.

This phase consists of three steps. The first step, called *valid substring generation*, aims to generate all valid substrings of string $S^{i,j,k}$. Then the second step, named *valid substring hashing and counting,* aims to count the times of each substring appears in the same space during consecutive x days. The last step, called *activity pattern candidates selection*, aims to select the candidates of activity pattern from all valid substring, the times of which appears in the same space exceed a threshold during consecutive x days.

3.4.1 Step 1: valid substring generation

Recall that each substring of $S^{i,j,k}$ might be a primitive activity. This motivates to consider all possible substrings in $S^{i,j,k}$ and count if each substring usually appears in the same space of different days. If it is the case, we will identify that this substring should be an activity.

This step first presents the description of valid substring. Here, a substring is valid only if it satisfies two criteria. First, it must be the substring of the string $S^{i,j,k}$ obtained from phase 3. Second, its length is not less than a threshold τ, which can be set by experimental comparisons. The following formally defines the substrings of string $S^{i,j,k}$ string $S^{i,j,k}$ be presented as:

$$S^{i,j,k} = \{c_i^{j,k,1}, c_i^{j,k,2}, ..., c_i^{j,k,t}\},$$

where $t = \beta_{i,j,k}$.

A string

$$S_{m,q}^{i,j,k} = \{c_i^{j,k,q+1}, c_i^{j,k,q+2}, ..., c_i^{j,k,q+m}\}$$

is said to be a *substring* of string $S^{i,j,k}$ if $0 \le q$ and $q + m \le t$, where m denotes the number of elements in $S_m^{i,j,k}$ and q represents the starting index in $S^{i,j,k}$.

Let $S_m^{i,j,k}$ denote a group consisting of all substring with the size of m, which can be denoted as:

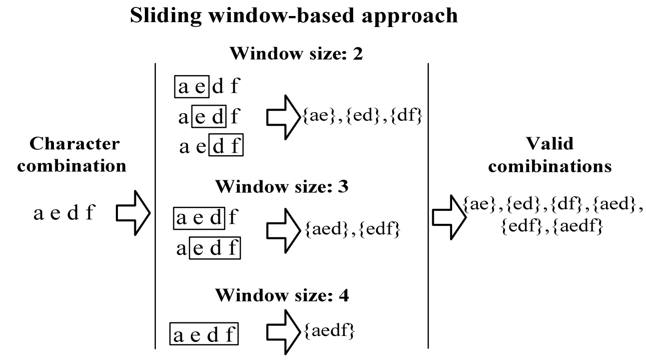
$$S_m^{i,j,k} = \{S_{m,q}^{i,j,k} \mid 0 \le q \text{ and } q + m \le t\}$$

Let $V^{i,j,k}$ denote the group of all valid substrings from string $S^{i,j,k}$, expressed by:

$$V^{i,j,k} = \{S_m^{i,j,k} \mid \tau \le m \le t\}$$

Here, we use a sliding window-based approach to extract all valid substrings of string $S^{i,j,k}$. For any string $S^{i,j,k}$, the window size m denotes the number of elements in its substring $S_{m,q}^{i,j,k}$. Figure 4 depicts an example of the process designed in step 1 of this phase.

Figure 3 An example of sliding window-based approach



As shown in Figure 3, a string $S^{i,j,k} = \{a, e, d, f\}$ is handled by sliding window-based approach, and the threshold is set to 2. Therefore, three different window sizes (i.e., 2, 3 and 4) are used according to the size of $S^{i,j,k}$. When the size of window is set to 2, the substrings are:

$$S_{2,1}^{i,j,k} = \{a, e\}, S_{2,2}^{i,j,k} = \{e, d\} \text{ and } S_{2,3}^{i,j,k} = \{d, f\}$$

As a result, the group of substring with size of 2 can be denoted as:

$$S_2^{i,j,k} = \{S_{2,1}^{i,j,k}, S_{2,2}^{i,j,k}, S_{2,3}^{i,j,k}\} = \{\{a, e\}, \{e, d\}, \{d, f\}\}$$

Similarly, when window size is set to 3, the substrings are:

$$S_{3,1}^{i,j,k} = \{a, e, d\}, S_{3,2}^{i,j,k} = \{e, d, f\}$$

Meanwhile, the group of substring with size of 3 is expressed by:

$$S_3^{i,j,k} = \{S_{3,1}^{i,j,k}, S_{3,2}^{i,j,k}\} = \{\{a, e, d\}, \{e, d, f\}\}$$

At last, there is only one substring $S_{4,1}^{i,j,k} = \{a, e, d, f\}$, when window size is 4. Therefore, the group of substring with size of 4 only has one member, denoted as:

$$S_4^{i,j,k} = \{S_{4,1}^{i,j,k}\} = \{\{a, e, d, f\}\}$$

After processed by sliding window-based approach, the group of all valid substrings from string $S = \{a, e, d, f\}$ is shown follows.

$$V^{i,j,k} = \{S_2^{i,j,k}, S_3^{i,j,k}, S_4^{i,j,k}\} = \{\{ae\}, \{ed\}, \{df\}, \{aed\}, \{edf\}, \{aedf\}\}$$

3.4.2 Step 2: hashing and counting all substrings

This step aims to improve the efficiency of exploring potential activity patterns from a large number of valid substrings. For all valid substrings collected from the same place, this step transfers them into numbers through a collision-resistant Hash method, denoted as *Hash*(*Object S*).

For any valid substring $S_{m,q}^{i,j,k}$ obtained in step 1, after performing hash operation, $S_{m,q}^{i,j,k}$ is hashed to a hash value represented by $H_{m,q}^{i,j,k}$. That is:

$$Hash\left(S_{m,q}^{i,j,k}\right) = H_{m,q}^{i,j,k}$$

Take $S_{2,1}^{i,j,k} = \{a, e\}$ as an example. The hash value of $S_{2,1}^{i,j,k}$ can be denoted as:

$$Hash\left(S_{2,1}^{i,j,k}\right) = Hash(\{a, e\}) = H_{2,1}^{i,j,k}$$

Then, for all valid substrings of all strings (obtained in step 1 of phase 4) in the j^{th} space of a house during *x*-day in a row expressed as a group:

$$V^{[i,i+x],j,*} = \left\{V^{i,j,*}, V^{i+1,j,*}, ..., V^{i+x,j,*}\right\}$$

where

$$V^{i,j,*} = \left\{V^{i,j,1}, V^{i,j,2}, ..., V^{i,j,\mu_{i,j}}\right\}$$

and $\mu_{i,j}$ denotes the number of groups in $V^{i,j,*}$.

What we need is to count the number of each valid substring appears based on their hash values denoted as a group:

$$H^{[i,i+x],j,*} = \left\{H^{i,j,*}, H^{i+1,j,*}, ..., H^{i+x,j,*}\right\}$$

where

$$H^{i,j,*} = H^{i,j} = \left\{H^{i,j,1}, H^{i,j,2}, ..., H^{i,j,k}, ..., H^{i,j,\mu_{i,j}}\right\}$$

and $\mu_{i,j}$ denotes the number of groups in $H^{i,j,*}$, which is the same as the number of the groups in $V^{i,j,*}$.

If the frequency of hash value $H_{m,q}^{i,j,k}$ in $H^{[i,i+x],j,*}$ is not less than φ_x in consecutive *x* days, the substring $S_{m,q}^{i,j,k}$ will be considered as a potential candidate of activity pattern, and will be put in a group denoted as $A^{[i,i+x],j,*}$.

Following the example of Figure 3 in step 1. It shows that:

$$\begin{aligned} V^{i,j,k} &= \left\{S_2^{i,j,k}, S_3^{i,j,k}, S_4^{i,j,k}\right\} \\ &= \left\{S_{2,1}^{i,j,k}, S_{2,2}^{i,j,k}, S_{2,3}^{i,j,k}, S_{3,1}^{i,j,k}, S_{3,2}^{i,j,k}, S_{4,1}^{i,j,k}\right\} \\ &= \{\{ae\}, \{ed\}, \{df\}, \{aed\}, \{edf\}, \{aedf\}\} \end{aligned}$$

With the operation defined in this step, the group of hash values is described as follows.

$$\begin{aligned} H^{i,j,k} &= Hash\left(V^{i,j,k}\right) \\ &= \left\{Hash\left(S_{2,1}^{i,j,k}\right), Hash\left(S_{2,2}^{i,j,k}\right), Hash\left(S_{2,3}^{i,j,k}\right),\right. \\ &\quad \left. Hash\left(S_{3,1}^{i,j,k}\right), Hash\left(S_{3,2}^{i,j,k}\right), Hash\left(S_{4,1}^{i,j,k}\right)\right\} \\ &= \{Hash(\{ae\}), Hash(\{ed\}), Hash(\{df\}), \\ &\quad Hash(\{aed\}), Hash(\{edf\}), Hash(\{aedf\})\} \end{aligned}$$

Here, we assume that:

$$\begin{aligned} &Hash(\{ae\}) = 00001, Hash(\{ed\}) = 00002, \\ &Hash(\{df\}) = 00003, Hash(\{aed\}) = 00004, \\ &Hash(\{edf\}) = 00005, Hash(\{aedf\}) = 00006. \end{aligned}$$

Therefore, $H^{i,j,k}$ ={00001, 00002, 00003, 00004, 00005, 00006}. Take $S_{2,1}^{i,j,k} = \{ae\}$ an example to demonstrate how to count the frequency of $\{ae\}$ in $V^{[i,i+x],j,*}$. Recall that $V^{[i,i+x],j,*}$ group consisting of all valid substrings of all strings (obtained in step 1 of phase 4) in the j^{th} space of a house during *x*-day in a row. Since we have compute the hash value of all valid substrings, which have been put in $H^{[i,i+x],j,*}$, we can count the frequency of $\{ae\}$ in $V^{[i,i+x],j,*}$by count the number of times that its hash value (i.e., 000001) appears in $H^{[i,i+x],j,*}$.

Finally, if the frequencies of $\{ae\}$, $\{ed\}$, $\{df\}$ and $\{aed\}$ appear in the same place are all not less than φ_x during *x*-day in a row, the $S_{2,1}^{i,j,k}$, $S_{2,2}^{i,j,k}$, $S_{2,3}^{i,j,k}$ and $S_{3,1}^{i,j,k}$ will be considered as a potential candidate of activity pattern, and will be put in a group $A^{[i,i+x],j,*}$.

3.4.3 Step 3: activity pattern candidates selection

Before presents the details of this step, the following firstly presents *substring inheriting property*.

Substring inheriting property

If a string $S_{m,q}^{i,j,k}$ exists in $A^{[i,i+x],j,*}$, which is generated in step 2, all valid substrings of $S_{m,q}^{i,j,k}$ generated in step 1 will also exist in $A^{[i,i+x],j,*}$. □

This property can be easily explained by using the following example. For instance, an activity of 'take shower' can be represented by substrings *abcd*, where *a*, *b*, *c* and *d* are the corresponding characters associated with the event data 'turn light', 'detected by infra sensor', 'detected by pleasure sensor', 'turn off the light', respectively. Assume that the frequency of 'take shower' larger than 300 per year, this also indicates that the frequency of its substrings, say, 'a' (or 'turn light') will also larger than 300.

Based on the *substring inheriting property*, the substrings of $S_{m,q}^{i,j,k}$ in the group are not necessary to be considered and they will affect the accuracy of activities recognition. Therefore, this step aims to remove all substrings for each $S_{m,q}^{i,j,k}$ from the group $A^{[i,i+x],j,*}$ obtained in step 2. After that, the remaining ones in $A^{[i,i+x],j,*}$ are the candidates of activity patterns, which will be used in phase

5, and we name the new group consisting of remaining ones as $\overline{A}^{[i,i+x],j,*}$. Each member of $\overline{A}^{[i,i+x],j,*}$ is denoted by $S_{m,q}^{y,j,*}$ where $i \le y \le i + x$.

Following the assumption of Figure 3 that the frequencies of $\{ae\}$, $\{ed\}$, $\{df\}$ and $\{aed\}$ appear in the same place are all not less than φ_x during x-day in a row. Based on the *substring inheriting property*, $\{ae\}$ and $\{ed\}$ will be removed, since they are substrings of $\{aed\}$. Finally, $\{df\}$ and $\{aed\}$ are the remaining ones that can be considered as the candidates of activity patterns.

The following presents *activity pattern candidates selection algorithm* (APCSA) which summaries the operations designed in step 3.

Activity pattern candidates selection algorithm

Input: All potential candidates of activity patterns in the j^{th} space during x-day in a row:

$$A^{[i,i+x],j,*}$$

Output: A groups containing all candidates of activity patterns in the j^{th} space during x-day in a row:

$$\overline{A}^{[i,i+x],j,*}$$

1. Sorting each member $S_{m,q}^{*,j,*}$ of $A^{[i,i+x],j,*}$ in the descending order of m
2. for ($m = \theta_l$; m++; $m < \theta_u$) {
   / *θ_l and θ_u are lower bound and upper bound of the size of member $S_{m,q}^{*,j,*}$ in $A^{[i,i+x],j,*}$ */
3. if ($S_{m,*}^{*,j,*} \in A^{[i,i+x],j,*}$)
4. for ($q = 0$; q++; $q \le n - m$) {
5. if ($S_{m,q}^{*,j,*} \subset S_{\theta_u,*}^{*,j,*}$)
6. remove $S_{m,q}^{*,j,*}$ from $A^{[i,i+x],j,*}$;
7. } /* end of for @line 6 */
8. } /* end of for @line 3 */
9. save the remaining strings in $A^{[i,i+x],j,*}$ to $\overline{A}^{[i,i+x],j,*}$;
10. Output $\overline{A}^{[i,i+x],j,*}$;

3.5 *Phase 5: activity identification*

This phase aims to identify the activity patterns from the candidates in $\overline{A}^{[i,i+x],j,*}$ generated in step 3 of phase 4. Specifically, it implements the activity identification by comparing each candidate in $\overline{A}^{[i,i+x],j,*}$ to the predefined activity patterns $R^j = \{r^{j,1}, r^{j,2}, ..., r^{j,e_j}\}$ in the j^{th} space of a house, where e_j is the number of predefined activities in the j^{th} space of a house.

For each member $S_{m,q}^{*,j,*}$ of $\overline{A}^{[i,i+x],j,*}$, the operation procedure of this phase are shown in the following algorithm.

As shown in the activity identification algorithm, computing the similarity between $S_{m,q}^{*,j,*}$ each predefined activity $r^{j,g}$ $(1 \le g \le e_j)$ in R^j, using the following function:

$$similarity\left(S_{m,q}^{*,j,*}, r^{j,g}\right)$$

which can be implemented by some classical or latest methods.

Activity identification algorithm

Input:

1. All candidates of activity patterns in the j^{th} space during x-day in a row:

$$\overline{A}^{[i,i+x],j,*}$$

2. All predefined activity patterns in the j^{th} space of a house

$$R^j = \{r^{j,1}, r^{j,2}, ..., r^{k,e_j}\}$$

/* e_j is the number of elements predefined in the j^{th} space */

Output:

1. A group of all activity patterns identified in the j^{th} space during x-day in a row:

$$\tilde{B}^{j,x}$$

2. A group of new activity patterns not defined in R^j:

$$\tilde{N}^{j,x}$$

1. for (each in $S_{m,q}^{*,j,*}$ in $\overline{A}^{[i,i+x],j,*}$){
2. $tmpValue = tempIndex = 0$
3. for ($g = 1$; g++; $g \le e_j$) {
4. if ($similarity(S_{m,q}^{*,j,*}, r^{j,g}) == 1$)
5. put $r^{j,g}$ in $\tilde{B}^{j,x}$;
6. else {
7. if ($similarity(S_{m,q}^{*,j,*}, r^{j,g}) \ge tmpValue$) { /*the initial value of *tmp* is 0*/
8. $tmpValue = similarity(S_{m,q}^{*,j,*}, r^{j,g})$;
9. $tmpIndex = g$;
10. } * end of if @line 7*/
11. } /* end of else @line 6 */
12. } /* end of for @line 3 */
13. if ($tmpValue \ge \varepsilon_j$) /* ε_j denotes the similarity threshold in the j^{th} space */
14. copy $r^{j,tmpIndex}$ to $\tilde{B}^{j,x}$;
15. $tmpValue = tmpIndex = 0$;
16. } *end of if @line 13*/
17. else
18. copy $S_{m,q}^{*,j,*}$ to $\tilde{N}^{j,x}$; /* which contains new activity pattern candidate is found from the j^{th} space, during x day. *
19. } /*end of for @line 1*/
20. output $\tilde{B}^{j,x}$ and $\tilde{N}^{j,x}$

We say $S_{m,q}^{*,j,*}$ is an activity pattern, if:

$$similarity\left(S_{m,q}^{*,j,*}, r^{j,g}\right) = 1$$

Besides, there probably exist the other situation need to be considered. That is, no similarity is 1, after comparing $S_{m,q}^{*,j,*}$ with all predefined activity patterns in R^j. For this situation, we choose the largest similarity among all similarities between $S_{m,q}^{*,j,*}$ and each predefined one $r^{j,g}$ in R^j. Here, if the largest one exceeds the threshold ε_j defined by experiments, we can also say $S_{m,q}^{*,j,*}$ is an activity pattern. Otherwise, $S_{m,q}^{*,j,*}$ is taken as a new pattern not predefined, and put in $\tilde{N}^{j,x}$, which contains new activity pattern candidate is found from the j^{th} space, during x days.

4 Complexity analysis

This section aims to theoretically analyse the time and space complexities of the proposed approach.

4.1 Assumptions and basic analysis

Before analysing our algorithm, we first introduce some assumptions. Assume the total number of sensing data from consecutive r days is N_r. Assume the number of sensing data in the i^{th} day is n_i. That is:

$$N_r = \sum_{i=1}^{r} n_i$$

Assume the data generated in each day can be further partitioned into s spaces according to the space constraint described in phase 2. Therefore, the number of sensing data in i^{th} day can be represented as:

$$n_i = \sum_{j=1}^{s} n_{i,j}$$

where $n_{i,j}$ means the number of sensing data in the j^{th} space of i^{th} day. Assume the sensing data in j^{th} space of i^{th} day are partitioned into t slices according to the time constraint described in Phase 2. Therefore, the number of sensing data in the j^{th} space of i^{th} day can be denoted as:

$$n_{i,j} = \sum_{k=1}^{t} n_{i,j,k}$$

where $n_{i,j,k}$ represents the number of sensing data in the k^{th} slice of j^{th} space in i^{th} day. Therefore, the total number of sensing data from consecutive r days can be expressed as:

$$N_r = \sum_{i=q}^{q+r} \sum_{j=1}^{s} \sum_{k=1}^{t} n_{i,j,k}$$

Take sensing data in the k^{th} slice of j^{th} space in i^{th} day as an example. Recall that these sensing data will be transformed into a string $S^{i,j,k}$ consisting of $n_{i,j,k}$ characters based on a mapping rule described in phase 3. Then, all possible substrings with a threshold size τ are generated from $S^{i,j,k}$. Let notation $\vartheta_{i,j,k}$ denote the numbers of substrings in the k^{th} slice of j^{th} space of i^{th} day. Based on the definition of substring in step 1 of phase 4, the number of substrings of $S^{i,j,k}$ is:

$$\vartheta_{i,j,k} = \frac{(\tau + n_{i,j,k}) * (n_{i,j,k} - \tau + 1)}{2}$$

Therefore, the total number of substrings in the i^{th} day, simply denoted by ϑ_i, is:

$$\vartheta_i = \sum_{j=1}^{s} \sum_{k=1}^{t} \vartheta_{i,j,k}.$$

and the total number of substrings, denoted by ϑ, in the consecutive r days is:

$$\vartheta = \sum_{i=q}^{q+r-1} \sum_{j=1}^{s} \sum_{k=1}^{t} \vartheta_{i,j,k}. \tag{1}$$

To simplify expression (1), without the loss of generality, we further assume that $s = 5$ (bedroom, bathroom, kitchen, dining room, and living room), $t = 3$ (morning, afternoon, and evening) and all n_i are equal to some certain constant n. As a result, expression (1) can be further simplified as follows.

$$\begin{aligned} \vartheta &\approx \sum_{i=q}^{q+r-1} \sum_{j=1}^{5} \sum_{k=1}^{3} \frac{\left(\theta + \frac{n}{3*5}\right) * \left(\frac{n}{3*5} - \theta + 1\right)}{2} \\ &= r * \left(\frac{n^2 + 15n}{15} - 15\theta(\theta - 1) \right) \end{aligned} \tag{2}$$

Here, the value shown in (2) can be considered as the upper bound of ϑ, because of the value of s and t we adopt close to their lower bounds.

4.2 Analysis of time and space complexities

The proposed algorithm mainly consists of five phases. In particular, there are two steps in phase 2 and three steps in phase 4. The following analyses the time and space complexities of each phase.

- Phase 1

  According to the above assumption in subsection A, the total number of sensing data in r days is $r * n$. Recall that the phase 1 of our approach aims to filter out the invalid data, which can be implemented by scanning all data once. Therefore, both the time and space complexities of phase 1 are $\boldsymbol{O(r * n)}$.

- Phase 2

  In the second phase of our approach, all data are segmented based on time and space constraints, which can be implemented by scanning all data twice (one for space constraint, and the other for time constraint). Therefore, both the time and space complexities of phase 2 are $\boldsymbol{O(r * n)}$.

- Phase 3

  In the third phase, our approach is to represent the sensor data by scanning them once. Therefore, both the time and space complexities of phase 3 are $\boldsymbol{O(r * n)}$.

- Phase 4

1. Time complexity of step 1

   In the fourth phase, based on the upper bound of ϑ, we can figure out that the time complexity of step 1 (i.e., generating the valid substrings) is $\boldsymbol{O(n^2)}$.

2. Time complexity of step 2

   The time complexity of hashing all substrings in step 2 is $O(n^2)$ and the time complexity of clustering (i.e., counting the number of times each substring appears based on hashing) in step 2 is $O(n^2)$. Therefore, the total time complexity of step 2 is $\boldsymbol{O(n^2)}$.

3. Time complexity of step 3

   As shown in the algorithm of step 3, the time complexity of line 1 is $O(n^2 + n)$ with the counting sort (Cormen et al., 2001), where n is the max length of all strings, and n^2 is the number of strings to be sorted. The time complexity of outer loop at line 3 is $O(n)$, where n is the upper bound of string length in $A^{[i,i+x],j,*}$. The time complexity of inner loop at line 6 is $O(n^2)$, because the total number of substrings is $r*\left(\frac{n^2+15n}{15}-15\theta(\theta-1)\right)$.

   Therefore, the time complexity of step 3 is $\boldsymbol{O(n^3)}$.

   Finally, the time complexity of phase 4 is the sum of that of three steps. That is, $O(n^3) + O(n^2) + O(n^2)$, which can be reduced to $\boldsymbol{O(n^3)}$.

4. Space complexity of step 1

   Since the total number of substrings ϑ in consecutive r days is $r*\left(\frac{n^2+15n}{15}-15\theta(\theta-1)\right)$, the space complexity to store these substrings generated in step 1 is $\boldsymbol{O(n^2)}$.

5. Space complexity of step 2

   The hash value computation as well as clustering (i.e., counting the number of times each substring appears based on hashing) for all substrings in step 2 have the same space complexity, i.e., $\boldsymbol{O(n^2)}$.

6. Space complexity of step 3

   In step 3, the space complexity of line 1 in APCSA has the same value with that of time complexity, i.e., $\boldsymbol{O(n^2 + n)}$, based on the counting sort, where n is the max length of all strings, and n^2 is the number of strings to be sorted. The time complexity of outer loop at line 3 is $O(1)$, because no additional space is needed.

   Finally, the space complexity of phase 4 is the sum of those of three steps, i.e., $O(n^2) + O(n^2) + O(n^2 + n)$, which can be reduced to $\boldsymbol{O(n^2)}$.

- Phase 5
- Without the loss of generality, the number of sensing data in each day is assumed to be n, and the numbers of spaces s of a house is set to a lower bound value 5. Furthermore, the upper bound of all candidates of activity patterns in the j^{th} space during x-day in a row (denoted as $\overline{A}^{[i,i+x],j,*}$) is $\frac{n^2}{25}$ based on the analysis in phase 4. Therefore, the time complexity of the outer loop in line 2 of *activity identification algorithm* is $O\left(\frac{n^2}{25}\right)$.
- The time and space complexities of computing in line 9 are similar. In line 9, the dynamic programming and divide-and-conquer algorithms 0 have been applied, which lead to the time and space complexities $O\left(\frac{n}{5}*\frac{n}{5}\right)$. Specifically, the first $\frac{n}{5}$ means the upper bound of string size in $\overline{A}^{[i,i+x],j,*}$, and the later one means the upper bound of predefined strings. Therefore, the time complexity of inner loop in line 4 is $O\left(e_j*\frac{n}{5}*\frac{n}{5}\right)$, where e_j is the number of elements predefined in the j^{th} space.
- At last, the total time complexity is:

$$O\left(\frac{n^2}{25}*e_j*\frac{n}{5}*\frac{n}{5}\right)=\boldsymbol{O\left(e_j*\frac{n^4}{625}\right)}.$$

- Besides, the space complexity of phase 5 is $\boldsymbol{O\left(\frac{2n}{5}\right)}$.
- Total complexities of time and space
- According to the analysis of time and space complexities of phases 1 to 5, the time complexity of our approach is $\boldsymbol{O(n^4)}$ while the space complexity is $\boldsymbol{O(n)}$ in the worst case.

5 Experiments

In this section, we first introduce the dataset of one smart home in Section 5.1, which is from Wireless & Mobile Network Laboratory, Tamkang University. Then the overview about the metrics and baseline are introduced in Section 5.2. In Section 5.3, the experimental settings are descried, which are followed by the experimental results and analysis.

5.1 Experimental data

The experiments are carried out based on the WMNL2016 datasets, which is collected in Wireless & Mobile Network Laboratory, Tamkang University. In the case of WMNL2016, 40 state-change sensors were deployed at an apartment with six areas (i.e., bedroom, kitchen, bathroom, dining room, living room, and foyer) and activities were performed by one volunteer for 12 months.

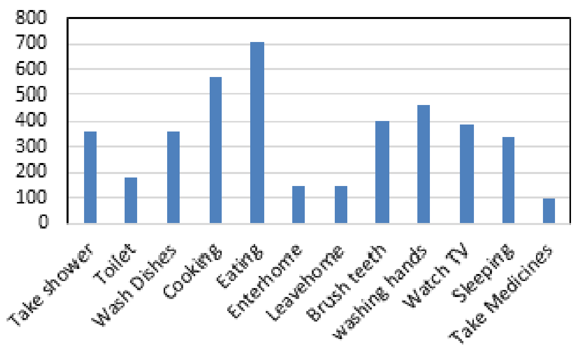
Table 2 Example of data format in WMNL2016

Apartment ID	*Area ID*	*Sensor type*	*Sensor ID*	*Info.*	*Time*
0001	04	PR01	06	0000000001	201601010728
0001	04	PR01	06	0000000001	201601010749
0001	04	PD02	01	0000000001	201601010753
0001	01	MG01	07	0000000001	201601010800
0001	01	PR01	01	0000000001	201601010801

The data format of *WMNL2016* is described in Table 2. Specifically, the 'Apartment ID' refers to the id of apartment, where the sensor events are recorded. The meaning of 'area ID' is the id the different areas (such as bedroom, kitchen, bathroom, dining room) in the apartment. 'Sensor type' means the type of different sensors (e.g., pressure sensor, water flow sensor, ultrasonic sensor, infrared sensor, and gas sensor) deployed in the apartment. 'Sensor ID' refers to the id of sensor. 'Information' means the sensor data and the 'Time' refers to the time when sensor event occurs.

Those activities of *WMNL2016* consist of sleeping, leave home, watch TV, take shower, toilet, take medicines, cooking, eating, wash dishes, enter home, brush teeth, and washing hands. The dataset includes 64,944 sensor events and 4,172 annotated activity instances. Specifically, the numbers of recorded activities over time are shown in Figure 4.

Figure 4 The numbers of recorded activities (see online version for colours)



5.2 Metrics and baseline

5.2.1 Metrics

Let $\tilde{B}^i = \{\tilde{b}_1^i, \tilde{b}_2^i, ..., \tilde{b}_{\tilde{q}_i}^i\}$ denote the set of potential activities of an elder, which are recognised from the sensor dataset D^i. A recognised activity $\tilde{b}_j^i (1 \le j \le \tilde{q}_i)$ is said to be *true positive* if $\tilde{b}_j^i$ is belong to the ADLs set B^i. Let γ_j^i denote the true positive for activity $\tilde{b}_j^i$, then notation γ_j^i can be represented by expression (3).

$$\gamma_j^i = \begin{cases} 1 & \text{if } \tilde{b}_j^i \in B^i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

To evaluate our approach, we adopt the accuracy of activity recognition as a metric in this section. Specifically, the metrics is computed as follows.

$$Accuracy = \frac{\sum_{k=1}^{N} T_k}{Total} \tag{4}$$

where N is the number of activities and T_k is the number of true positives of the k^{th} activity.

5.2.2 Approaches and baseline

The naive Bayes (NB) method exploits the relative frequencies of feature values and the frequency of activities from the training data to construct a mapping from activity features to an activity.

The hidden Markov model (HMM) is a statistical method. The underlying model of HMM is a stochastic Markovian process, which is not observable but can be observed by other processes that produce the sequence of observed features.

The conditional random field (CRF) *approach* derives a label for the current data point, not only based on the transition likelihoods between states, but also the emission likelihoods between the states of activity and observable states. The CRF approach obtains a label sequence, which is corresponding to the observed sequence of features.

Support vector machine (SVM) is supervised learning model with associated learning algorithms that analyses data used for classification and regression analysis.

Baseline. In this paper, we take SVM as the baseline to evaluate our approach, because SVM model yields the most consistent performance for modelling and recognising activities in comparison to NB classifier, HMM, according to the experimental results in 0.

5.3 Experimental settings and analysis

In this section, we conduct three experiments to evaluate our approach. In the first experiment, we compare different thresholds, which are used in time partition based on time constraints in phase 2, for different activities to choose the predefined values for the following experiments. In the second experiment, we design a test to demonstrate how the number of training days in dataset affects the performance of our approach. In the third experiment, we test our approach in comparison to the baseline with the predefined threshold obtained in the first experiment.

5.3.1 Experiment 1

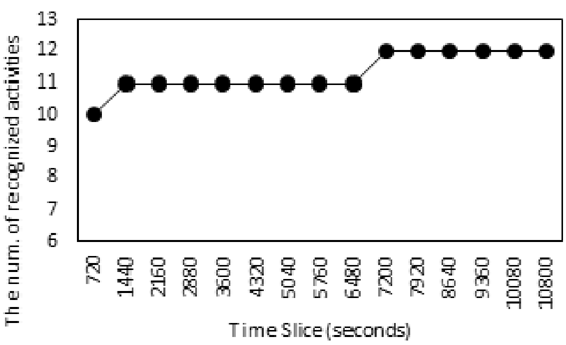
In this part, we first take three activities (i.e., 'sleep', 'take shower', and 'leave home') as examples to demonstrate experimental process of how to get their thresholds based on the dataset of *WMNL2016*, while the experimental process for other activities are omitted here due to the limitation of space. Then, we show how to determine the length of substring (i.e., τ) used in step 1 and threshold of substring frequency (i.e., φ_x) during x-day in a row.

1 Threshold for 'sleep' activity

We run the experiment 11 rounds for 'sleep' activity. The initial value for the threshold of 'sleep' activity in the first round is set to 720 s. Then, the threshold is set to the value with the step-size of 720 in the rest rounds.

As shown in Figure 5, the number of recognised activities is ten, when threshold is set to 720 s, while the number is 11 when it is set to the value from 1,440 to 6,480. The number is 12 when the threshold is set to the value from 7,200 to 10,800, which is the number of activities of daily life in *WMNL2016*. Based on these experimental results, we take 7,200 s (i.e., 2 hours) as the empirical value for the threshold of 'sleep' activity.

Figure 5 The num. of recognised activities with different threshold for 'sleep' activity



2 Threshold for 'take shower' activity

For 'take shower' activity, we run the experiment ten rounds. The initial value for the threshold of take shower' activity in the first round is set to 60 s. Then, the threshold is set to the value with the step-size of 60 in the rest rounds. It can be seen from Figure 6 that the number of recognised activities is 12 when the threshold is set to 120, 180, and 240, respectively. Therefore, we take their mean (i.e., 180) as the empirical value for the threshold of 'take shower' activity.

3 Threshold for 'leave home' activity

We run the experiment ten rounds for 'leave home' activity. The initial value for the threshold of 'leave home' activity in the first round is set to 1,800 s. Then, the threshold is set to the value with the step-size of 1,800 in the rest rounds. As shown in Figure 7, the number of recognised activities is 11, when threshold is set to the value from 1,800 to 10,800 s, respectively. The number is 12 when it is set to the value from 12,600 to 18,000. According to these experimental results, we take 12,600 s as the empirical value for the threshold of 'leave home' activity.

Figure 6 The num. of recognised activities with different threshold for 'take shower' (see online version for colours)

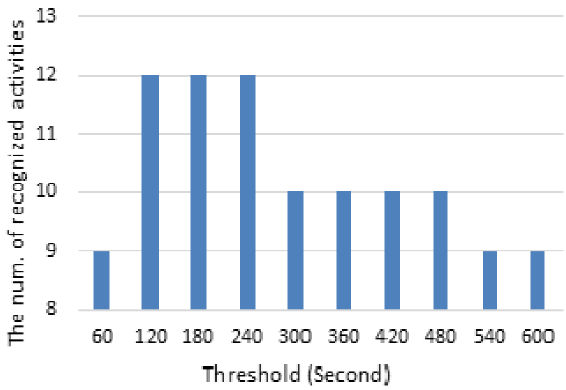


Figure 7 The num. of recognised activities with different threshold for 'leave home' activity

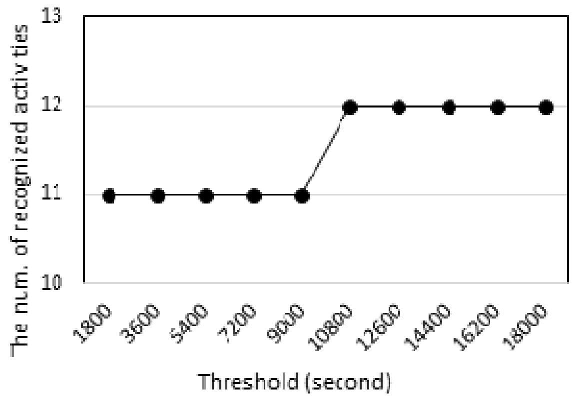
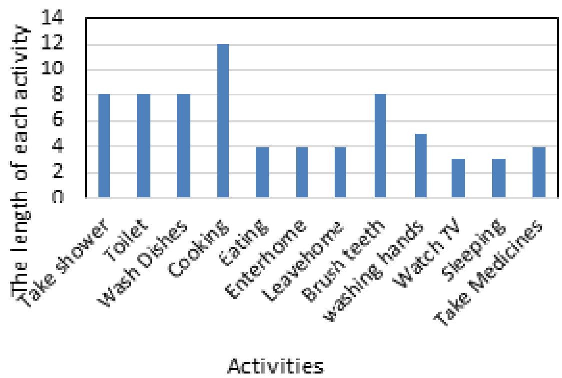


Figure 8 The length of each activity in *WMNL2016* (see online version for colours)



4 The value of τ

In order to determine the length of substring (i.e., τ) used in step 1, we count the length of each activity in *WMNL2016*. The statistics are shown in Figure 8.

As shown in Figure 8, the shortest length of activity is 3. Therefore, we take 2 as the value of τ, which is used in step 1, the valid substring generation of phase 4.

5 The value of φ_x

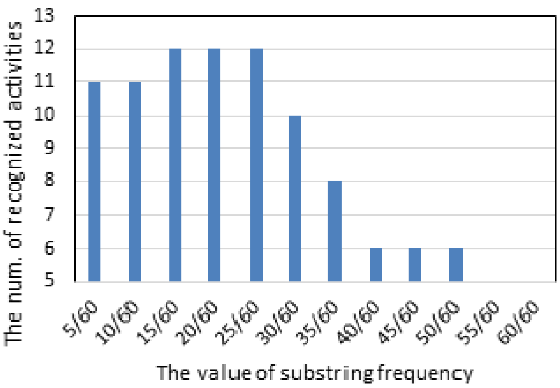
In this part, we take the data of 60 days in *WMNL2016* to determine the threshold of substring frequency (i.e., φ_{60}) during 60-day in a row.

We run the experiment ten rounds with different substring frequencies. The initial value for the substring frequency in the first round is set to 5/60. Then, the threshold is set to the value with the step-size of 5/60 in the rest rounds.

As shown in Figure 9, the number of recognised activities is 12, when the number of substring frequency is set to 15/60, 20/60 and 25/60, respectively. Therefore, we the average number (i.e., 1/3) as the default value of substring frequency.

Besides, the empiric values of rest parameters used in our approach are set to 120 s, which are obtained by conducting many experiments while the experimental process are omitted here due to the limitation of space.

Figure 9 The number of recognised activities with different substring frequencies (see online version for colours)

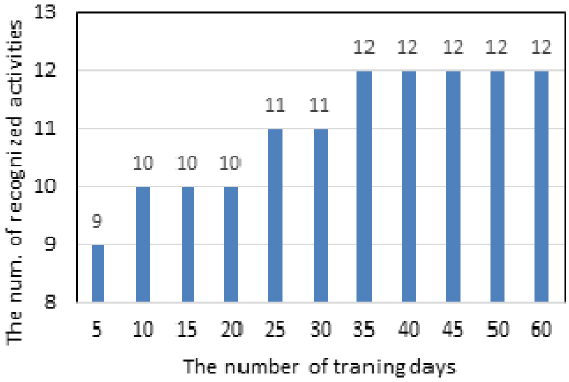


5.3.2 *Experiment 2*

In this part, we evaluate how the number of training days in dataset affects the performance of our approach. Specifically, we run this experiment 11 rounds. The initial value in the first round is set to five days. Then, the number is set to the value with the step-size of five for the rest rounds.

As shown in Figure 10, the number of recognised activities is nine, when the number of training days is set to five, while the number is ten when it is set to the value from 10 to 20. The number of recognised activities is 11 when the number of training days is set to 25 and 30, respectively. Furthermore, the number of recognised activities is 12, when the number of training days is set to 35 or higher. These experimental results show that the number of training days in the following experiment cannot be less than 35.

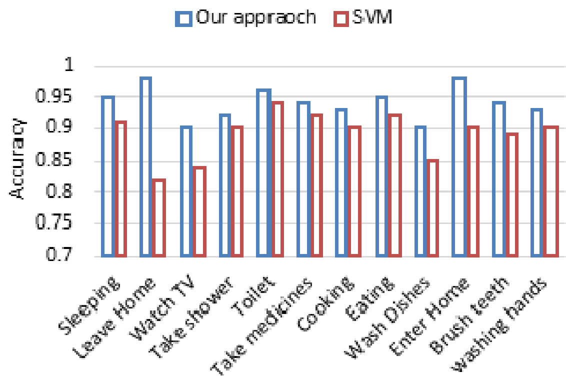
Figure 10 The number of recognised activities with different number of training days (see online version for colours)



5.3.3 *Experiment 3*

In this part, we evaluate our approach in comparison to the baseline (i.e., SVM). The experimental results are demonstrated in Figure 11, and it can be seen that our approach outperforms the baseline for the recognition of all different activities.

Figure 11 The accuracy of different approaches (see online version for colours)



For example, the accuracies of our approach to recognise ‘sleeping’ and ‘leave home’ are 0.95 and 0.98, while that of SVM are 0.91 and 0.82, respectively. The average accuracy of our approach and the baseline are shown in Table 3. In comparison to the baseline, the improvement of average accuracy by our approach is about 5.6%, which is calculated from the following formula: $(0.94-0.89)/0.89$.

Table 3 The average accuracy of different approaches

Methods	*Average accuracy*
Our approach	0.94
SVM	0.89

The reason why our proposed approach outperforms baseline is that it takes both time and space constraints into consideration when recognise activities. That is, the spatial and temporal characteristics of different activities make the sensor data segmentation more efficient and effective.

6 Conclusions and future work

In this paper, we proposed an activity recognition approach based on spatial-temporal constraints. The overall accuracy of our approach is about 0.94 when it runs on the dataset from Wireless & Mobile Network Laboratory, Tamkang University.

Activity recognition is the first step in the aged-care in smart home. After that, we need to predict the activities performed by elderly and recognise the abnormal activities in elderly's daily life. Besides, we can infer the health and status of elderly through the recognised activities, which can support service providers to customise the service for elderly. In conclusion, a considerable amount of research work needs to be carried out in the area of aged-care in smart home. In the future, we will first focus on the abnormal activity recognition of elderly in daily life with deep learning and some other machine learning technologies.

Acknowledgements

We thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by Doctoral Science Foundation of Chuzhou University (No. 2017qd10), Chuzhou S&T Program (No. 201712), and Special S&T Project of Anhui Province (No. 16030901057). It was also supported by Program for Science and Technology Innovative Research Team of Chuzhou University (Key technologies and applications of IoT), the International S&T Cooperation Program of Anhui Province (No. 1704e1002217), the Outstanding Youth Support Project of Anhui Province (No. gxyq2017087), and the Natural Science Research Project of Higher Education in Anhui Province (No. KJ2017A420).

References

Abdelnasser, H., Youssef, M. and Harras, K.A. (2015) 'Wigest: a ubiquitous wifi-based gesture recognition system', in *Computer Communications* (*INFOCOM*), *2015 IEEE Conference*, IEEE, April, pp.1472–1480.

Ali, K., Liu, A.X., Wang, W. and Shahzad, M. (2015) 'Keystroke recognition using wifi signals', in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ACM, September, pp.90–102.

Al-Qaness, M.A.A., Li, F., Ma, X., Zhang, Y. and Liu, G. (2016) 'Device-free indoor activity recognition system', *Applied Sciences*, Vol. 6, No. 11, p.329.

Amiribesheli, M., Benmansour, A. and Bouchachia, A. (2015) 'A review of smart homes in healthcare', *Journal of Ambient Intelligence and Humanized Computing*, Vol. 6, No. 4, pp.495–517.

Bhattacharya, S. and Lane, N.D. (2016) 'From smart to deep: robust activity recognition on smartwatches using deep learning', in *Pervasive Computing and Communication Workshops* (*PerCom Workshops*), *2016 IEEE International Conference*, IEEE, March, pp.1–6.

Buettner, M., Prasad, R., Philipose, M. and Wetherall, D. (2009) 'Recognizing daily activities with RFID-based sensors', in *Proceedings of the 11th International Conference on Ubiquitous Computing*, ACM, September, pp.51–60.

Cook, D.J. and Schmitter-Edgecombe, M. (2009) 'Assessing the quality of activities in a smart environment', *Methods of Information in Medicine*, Vol. 48, No. 5, p.480.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2001) '8.2: counting sort', *Introduction to Algorithms*, pp.168–170, MIT Press, Cambridge, Massachusetts.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T. (2015) 'Long-term recurrent convolutional networks for visual recognition and description', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2625–2634.

Eftekhari, H.R. and Ghatee, M. (2016) 'An inference engine for smartphones to preprocess data and detect stationary and transportation modes', *Transportation Research Part C: Emerging Technologies*, Vol. 69, No. 8, pp.313–327.

Hugo, G. and Sa, A. (2014) *The Demographic Facts of Ageing in Australia* [online] http://www.dss.gov.au (accessed 23 February 2019).

Jalal, A., Kamal, S. and Kim, D. (2014) 'A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments', *Sensors*, Vol. 14, No. 7, pp.11735–11759.

Jalal, A., Kim, Y.H., Kim, Y.J., Kamal, S. and Kim, D. (2017) 'Robust human activity recognition from depth video using spatiotemporal multi-fused features', *Pattern Recognition*, Vol. 61, No. 1, pp.295–308.

Jokanović, B. and Amin, M. (2018) 'Fall detection using deep learning in range-Doppler radars', *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 1, pp.180–189.

Kim, Y. and Ling, H. (2009) 'Human activity classification based on micro-Doppler signatures using a support vector machine', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 47, No. 5, pp.1328–1337.

Kwapisz, J.R., Weiss, G.M. and Moore, S.A. (2011) 'Activity recognition using cell phone accelerometers', *ACM SigKDD Explorations Newsletter*, Vol. 12, No. 2, pp.74–82.

Lara, O.D. and Labrador, M.A. (2013) 'A survey on human activity recognition using wearable sensors', *IEEE Communications Surveys and Tutorials*, Vol. 15, No. 3, pp.1192–1209.

Li, H., Ye, C. and Sample, A.P. (2015) 'IDSense: a human object interaction detection system based on passive UHF RFID', in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, April, pp.2555–2564.

Li, X., Zhang, Y., Marsic, I., Sarcevic, A. and Burd, R.S. (2016) 'Deep learning for RFID-based activity recognition', in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, ACM, November, pp.164–175.

Lien, J., Olson, E.M., Amihood, P.M. and Poupyrev, I. (2016) *RF-Based Micro-Motion Tracking for Gesture Tracking and Recognition*, Leland Stanford Junior University and Google Inc, U.S. Patent Application 15/142,689.

Liu, L., Popescu, M., Skubic, M., Rantz, M., Yardibi, T. and Cuddihy, P. (2011) 'Automatic fall detection based on Doppler radar motion signature', in *Pervasive Computing Technologies for Healthcare* (*PervasiveHealth*), *2011 5th International Conference*, IEEE, May, pp.222–225.

Matsuo, K., Yamada, K., Ueno, S. and Naito, S. (2014) 'An attention-based activity recognition for egocentric video', in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (*CVPRW*), IEEE, June, pp.565–570.

Montoya, D., Abiteboul, S. and Senellart, P. (2015) 'Hup-me: inferring and reconciling a timeline of user activity from rich smartphone data', in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, November, p.62.

Mukhopadhyay, S.C. (2015) 'Wearable sensors for human activity monitoring: a review', *IEEE Sensors Journal*, Vol. 15, No. 3, pp.1321–1330.

Ordónez, F.J., de Toledo, P. and Sanchis, A. (2013) 'Activity recognition using hybrid generative/discriminative models on home environments using binary sensors', *Sensors*, Vol. 13, No. 5, pp.5460–5477.

Pal, S. and Abhayaratne, C. (2015) 'Video-based activity level recognition for assisted living using motion features', in *Proceedings of the 9th International Conference on Distributed Smart Cameras*, ACM, September, pp.62–67.

Pansiot, J., Stoyanov, D., McIlwraith, D., Lo, B.P. and Yang, G.Z. (2007) 'Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems', in *4th International Workshop on Wearable and Implantable Body Sensor Networks* (BSN 2007), Springer, Berlin, Heidelberg, pp.208–212.

Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J. and Korhonen, I. (2006) 'Activity classification using realistic data from wearable sensors', *IEEE Transactions on Information Technology in Biomedicine*, Vol. 10, No. 1, pp.119–128.

Pu, Q., Gupta, S., Gollakota, S. and Patel, S. (2013) 'Whole-home gesture recognition using wireless signals', in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, ACM, September, pp.27–38.

Reyes-Ortiz, J.L., Oneto, L., Samà, A., Parra, X. and Anguita, D. (2016) 'Transition-aware human activity recognition using smartphones', *Neurocomputing*, Vol. 171, No. 1, pp.754–767.

Rodriguez, M., Orrite, C., Medrano, C. and Makris, D. (2016) 'A time flexible kernel framework for video-based activity recognition', *Image and Vision Computing*, Vol. 48, No. 4, pp.26–36.

Sankaran, K., Zhu, M., Guo, X.F., Ananda, A.L., Chan, M.C. and Peh, L.S. (2014) 'Using mobile phone barometer for low-power transportation context detection', in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, ACM, November, pp.191–205.

Tapia, E.M., Intille, S.S. and Larson, K. (2004) 'Activity recognition in the home using simple and ubiquitous sensors', in *International Conference on Pervasive Computing*, Springer, Berlin, Heidelberg, April, pp.158–175.

Van Kasteren, T., Noulas, A., Englebienne, G. and Kröse, B. (2008) 'Accurate activity recognition in a home setting', in *Proceedings of the 10th International Conference on Ubiquitous Computing*, ACM, September, pp.1–9.

Wang, W., Liu, A.X., Shahzad, M., Ling, K. and Lu, S. (2015) 'Understanding and modeling of wifi signal based human activity recognition', in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ACM, September, pp.65–76.

Wang, Y., Liu, J., Chen, Y., Gruteser, M., Yang, J. and Liu, H. (2014) 'E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures', in *Proceedings of the 20th annual international conference on Mobile computing and networking*, ACM, pp.617–628.

Wei, B., Hu, W., Yang, M. and Chou, C.T. (2015) 'Radio-based device-free activity recognition with radio frequency interference', in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ACM, April, pp.154–165.

Wilson, D.H. and Atkeson, C. (2005) 'Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors', in *International Conference on Pervasive Computing*, Springer, Berlin, Heidelberg, May, pp.62–79.

Zheng, Y., Wu, C., Qian, K., Yang, Z. and Liu, Y. (2017) 'Detecting radio frequency interference for CSI measurements on COTS WiFi devices', in *Communications* (*ICC*), *2017 IEEE International Conference*, IEEE, May, pp.1–6.