Device-Free Non-Privacy Invasive Classification of Elderly Travel Patterns in A Smart House Using PIR Sensors and DCNN

Munkhjargal Gochoo, Tan-Hsu Tan, *Senior Member*, *IEEE*, Vijayalakshmi Velusamy,
Shing-Hong Liu, Damdinsuren Bayanduuren, and Shih-Chia Huang, *Senior Member, IEEE*

***Abstract*— Single resident life style is increasing among the elderly due to the issues of elderly care cost and privacy invasion. However, the single life style cannot be maintained if they have dementia. Thus, the early detection of dementia is crucial. Systems with wearable devices or cameras are not preferred choice for the long-term monitoring. Main intention of this study is to propose Deep Convolutional Neural Network classifier (DCNN) for indoor travel patterns of elderly people living alone using open dataset collected by device-free non-privacy invasive binary (passive infrared) sensor data. Travel patterns are classified as direct, pacing, lapping, or random according to Martino-Saltzman (MS) model. MS travel pattern is highly related with person's cognitive state, thus can be used to detect early stage of dementia. We have utilized an open dataset that was presented by Center for Advanced Studies in Adaptive Systems (CASAS) project, Washington State University. The dataset was collected by monitoring a cognitively normal elderly person by wireless passive infrared sensors for 21 months. First, 117320 travel episodes are extracted from the dataset and classified by MS travel pattern classifier algorithm for the ground truth. Later, 12000 episodes (3000 for each pattern) were randomly selected from the total episodes to compose training and testing dataset. Finally, DCNN performance was compared with seven other classical machine-learning classifiers. The Random Forest and DCNN yielded the best classification accuracies of 94.48% and 97.84%, respectively. Thus, the proposed DCNN classifier can be used to infer dementia through travel pattern matching.**

***Index Terms*— non-privacy invasive, deep learning, device-free, assistive technology, smart house, travel pattern, elder care.**

I. Introduction

ACCORDING to statistics, the number of people who live alone at home [1]–[6] and the number of single-resident houses [6] are increasing worldwide, and the global elderly population (over 60 years) is estimated to be 1.2 billion by 2025 [7]. Moreover, elderly people prefer an independent and aging-in-place life style due to the high cost of health care services and the privacy concern of living with a caregiver [8]. However, the independent life cannot be maintained in case of the person has physical or mental issues such as dementia.

The number of people with dementia (PwD) is estimated to increase more than triple (81 million) in 2040 as compared to 2001 [9] worldwide. Dementia development can be delayed by months to years if the person can be properly treated at the early stage of dementia [7]. Thus, the early detection of dementia plays crucial role in elderly independent life.

Generally, there are three types of monitoring schemes by using: (1) wearable devices [10]–[18]; (2) stationary sensors such as cameras [19]; (3) non-privacy invasive anonymous binary sensors [7], [20]–[23] such as passive infrared (PIR) sensors, piezo sensors, magnetic switches, passive RFID tags, etc. Among these three schemes, camera is the most accurate for the location detection but the least preferred one due to its privacy invasiveness. Wearable devices are less invasive than the cameras but they are proven not practical in a long-term monitoring application. A recent study shows that one third of the people who use a wearable activity tracker stopped using the tracker in six months due to its natural flaws such as the device can be lost easily, short battery life and uncomfortableness to wear [24], [25]. Thus, device-free and non-privacy invasive systems are the most promising solution for a long-term monitoring applications.

Main objective of this work is to propose a device-free non-privacy invasive Deep Convolutional Neural Network (DCNN) classifier for MS's travel patterns of elderly living alone using an open dataset acquired by the wireless binary sensors. We employed Naïve Bayes (NB), Support Vector Classification (SVC), K-Nearest Neighbor (KNN), Decision Tree (DT), Gradient Boost (GB), Random Forest (RF) and One VS rest (OVR) machine learning classifiers to compare the

This study was partially supported by Ministry of Science and Technology of the Republic of China (Taiwan) under the Contract No. MOST 106-2218-E-027-017 and MOST 106-2221-E-027-137
. (*Corresponding author: Tan-Hsu Tan*)

M. G. is with Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan, R.O.C., and School of Information and Communication Technology, Mongolian University of Science and Technology, Ulaanbaatar, Mongolia (e-mail: g.munkhjargal@must.edu.mn).

T.-H. T. is with Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608 Taiwan, R.O.C. (e-mail: thtan@ntut.edu.tw).

V. V. is with Department of Electrical and Electronic Engineering, Manchester Metropolitan University, Manchester M1 5GD, UK (e-mail: v.velusamy@mmu.ac.uk).

S.-H. L. is with Department of Information Engineering, Chaoyang University of Technology, Taichung 413, Taiwan, R.O.C. (e-mail: shliu@cyut.edu.tw).

D. B. is with School of Information and Communication Technology, Mongolian University of Science and Technology, Ulaanbaatar, Mongolia (e-mail: damdinsurenb@must.edu.mn).

S.-C. H is with Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan, R.O.C. (e-mail: schuang@ntut.edu.tw).

performance of the proposed DCNN classifier. In this study, we utilized the open dataset offered by Center for Advanced Studies in Adaptive Systems (CASAS), Washington State University (WSU) [26] and our contributions in this study are summarized as follows:

- We propose a novel device-free non-privacy invasive MS travel pattern classification method for the elderly living alone;
- For the first time, we converted PIR sensor logs into a binary image for the machine learning purpose;
- For the first time, we extracted MS travel patterns from an open dataset that collected by wireless binary sensors during a long-term real-life experiment in a smart house.
- To our best of knowledge, we propose DCNN classifier that has the highest performance for MS travel pattern classification.

The rest of this paper is organized as follows. The related works in the literature is discussed in section II followed by (III) the description of the proposed methods in detail. In section IV, the performance evaluation on the Aruba open dataset provided by CASAS project is demonstrated. In section V, the limitation and advantages of our systems are discussed. Finally, the paper is concluded in section VI.

II. Related works and background

A. Related Works

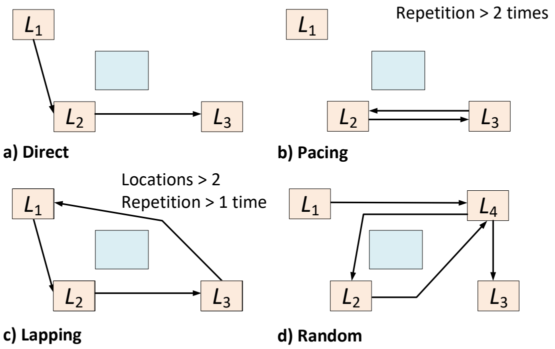
Researchers [27], [28] have found that Martino-Saltzman's (MS) travel pattern model is a useful tool to detect wandering patterns of PwD and few studies [12], [29]–[31] have employed MS model to monitor elderly people for outdoor or indoor wandering detection purpose.

Vuong *et al*. [29] used a dataset which was suggested by Makimoto et al. [14] for MS travel pattern detection algorithm. The dataset was collected in Korea and Japan for 7 and 14 days, respectively, by RFID tags sewed into the clothes of 20 institutionalized elders with dementia. Vuong's algorithm is straightforward and accurate for detecting MS travel patterns, thus we employed this algorithm in this study to prepare the ground truth dataset. Zhao *et al*. [31] conducted a pilot study for device-free wandering detection based on MS pattern using PIR sensors installed on top of each door. According to the experimental results, average MS pattern detection accuracy was 90.03%. However, the random pattern detection accuracy was 74.11% and wandering patterns within the room cannot be detected due to few sensors. Kumar *et al*. [12] made preliminary work of a grid-based method to detect indoor and outdoor wandering events based on MS travel patterns using GPS for outdoor, and an ultra-wide band radio tag for indoor localization. Similarly, Batista *et al*. [30] used GPS to detect wandering patterns based on MS travel patterns. However, all these studies used wearable devices, and none has demonstrated the device-free and non-privacy invasive MS travel pattern detection system. This motivates us to propose a novel device-free non-privacy invasive supervised machine learning classifier for MS travel patterns using PIR sensors. PIR sensors cannot identify the person but it can remotely sense the presence of a person without raising any privacy issues.

B. Martino-Saltzman's Travel Patterns of PwD

Martino-Saltzman *et al*., [32] modeled travel patterns for PwD into four types after systematically examined the indoor travel patterns of 40 residents with dementia in a nursing home. Among the residents, 24 of them were identified as wanderers or suspected wanderers, and the rest 16 residents were identified as non-wanderers by nursing staff. Four basic travel patterns were observed during the experiment which were direct, pacing, lapping, and random as shown in Fig. 1. Travel efficiency (percentage of direct travel) is highly related to cognitive status of the residents. The findings have been used in various dementia-related studies.

The travel patterns are categorized to an efficient (direct) and inefficient (pacing, lapping, and random) travels. Inefficient travels are referred as wandering and have been used to define the wandering behavior of PwD. Dementia-related wandering is defined as "A syndrome of dementia-related locomotion behavior having a frequent, repetitive, temporally-disordered and/or spatially-disoriented nature that is manifested in lapping, random and/or pacing patterns, some of which are associated with eloping, eloping attempts, or getting lost unless accompanied" [33]. Moreover, wandering patterns can be classified into three different wandering types (classic, moderate, and subclinical) with other spatial and temporal parameters such as rate, duration, and time of the day of wandering [33]. Several studies [27], [28] suggest that the random-pattern is the most severe symptom in PwD followed by lapping and pacing.



a) Direct b) Pacing

c) Lapping d) Random

Fig. 1. Martino-Saltzman's travel patterns: (a) Direct; (b) Pacing; (c) Lapping; and (d) Random.

C. Location, Movement, and Episode

The three concepts for a wandering patterns are: location, movement, and episode [29]. A "location" can be represented as coordinates in the grid layout or places such as bed, dining table, bathroom, etc. (Fig. 5). A "movement" is an action defined as moving from the current location to the next location. Each movement must have only two locations. An "episode" consists of one or more sequential movements, and each episode has start and stop locations. We denote L_1, L_2, L_3, and L_4 as locations.

Direct pattern is a single straightforward path between two locations without diversion or crossing in between. An episode consisting of more than one sequential direct patterns to different locations is also considered as direct.

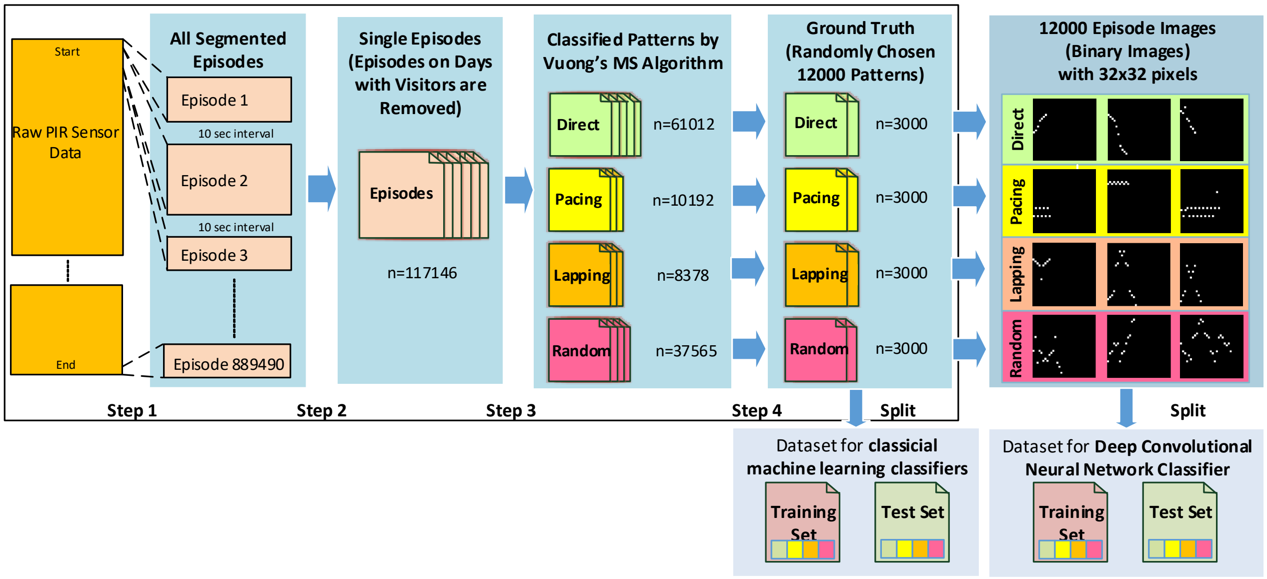


Fig. 2. Framework of the training and testing datasets for the machine learning travel pattern classifiers.

If a travel path intersects at some point, the travel is not considered as direct because it contains redundant sub-path. For example, a path $L_1L_2L_3L_4$ is a direct travel, but $L_1L_2L_3L_2L_3L_4$ is not direct since it has redundant sub-path from L_3 to L_2 and back to L_3, and considered as an inefficient travel. Thus, direct patterns must move through different locations.

Pacing is a repeated path between two locations that has more than two consecutive repetitions. For example, $L_1L_2L_1L_2$ L_1L_2 is a pacing pattern between L_1 and L_2 locations.

Lapping is a repeated circular path either in the same direction or the opposite direction. Lapping must have multiple repeated circular paths which has at least three different locations. For example, $L_1L_2L_3L_4L_1L_2L_3L_4$ (same direction) and $L_1L_2L_3L_4L_3L_2L_1$ (opposite direction) are lapping patterns.

Random is a path, which has multiple locations with no particular order. A random pattern must include at least one location that occurred more than once and it must be non-direct. Because of these two conditions, lapping and pacing patterns can be included in random patterns.

III. Methods

Fig. 2 illustrates a framework of the training and testing datasets preparation for the MS travel pattern classifiers. In dataset preparation, firstly, 889490 episodes (E_1, E_2, ... E_n) are segmented from the raw data which is collected via non-privacy invasive wireless binary sensors. Each episode consists of at least two movements (M_1, M_2, ... M_n), and each movement has two locations (L_1 and L_2).

There are basically two types of episodes which are (a) resident episodes (made by the resident when he/she is alone); (b) multiperson episodes (made by the resident, the visitor(s), or by the both when the visitor(s) are present at home). Thus, we need to exclude the multiperson episodes from all episodes to acquire correct travel patterns of the resident.

Secondly, the multiperson episodes are removed using a simple visitor detection algorithm based on the approximate walking speed (9). The algorithm assumes that there are multiple persons if the approximate speed between two PIR sensors is higher than 10m/s which could be only made by multiple persons who are moving around the house at the same time, because a single elderly person cannot achieve this high speed. For the sake of simplicity, 117146 resident episodes are left after removing all episodes that are made on days when visitors are detected.

Thirdly, the segmented episodes are classified into four patterns using Vuong's [29] MS travel pattern classification algorithm. There were 61012, 10192, 8378, and 37564 patterns classified as direct, pacing, lapping, and random. Fourthly, 3000 patterns from each travel pattern, totally 12000 patterns, were randomly selected as the ground truth for the training and testing the machine learning classification models. For DCNN classification model, 12000 classified episodes were converted into 32×32 binary episode images as shown in Fig. 2.

Then, these 12000 episodes and episode images were fragmented into 10 different sets each consists of a training set (90%) and a testing set (10%) for 10-fold cross-validation. Fig. 3 (a) represents the training process of the machine learning classifier which take the labels and extracted features as the inputs.

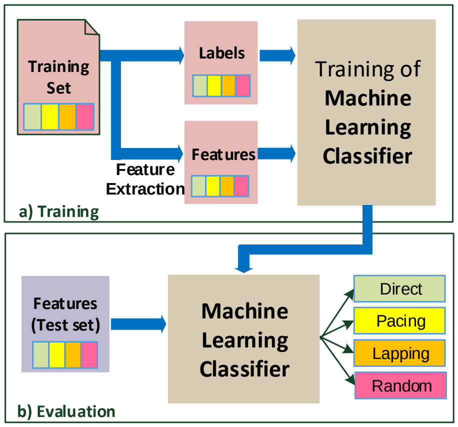


Fig. 3. (a) Training process and (b) Evaluation process of machine learning classifiers for travel patterns.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2771287, IEEE Sensors Journal

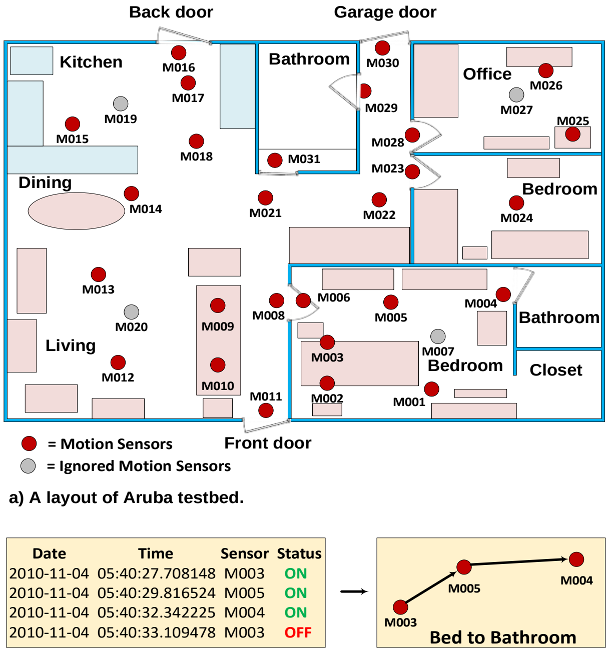
Fig. 3 (b) represents the evaluation part of the classifiers where the test set's extracted features are inputted to the trained classification model, and then the classifier outputs the predicted labels for the corresponding features.

The accuracy of the classification model is highly dependent on the number of training samples and the structure of the model. In the following subsections, all parts are explained in detail.

A. Smart Home Environment

Aruba testbed, shown in Fig. 4, is one of the testbeds of CASAS project [26] that is chosen for this study. CASAS is a long-term project, which studies about daily life events of residents in the smart home using wireless non-invasive binary sensors, and the project offers open datasets for researchers. The real life experiments are conducted according to the ethical approval of WSU review board. Fig. 4 (a) illustrates a layout of Aruba testbed. Aruba testbed has a kitchen, a living room, a dining area, two bedrooms, an office, two bathrooms, a pantry, a backyard, and a garage.

The testbed is equipped with 31 wireless PIR motion sensors, four door switch sensors, and four temperature sensors. Though, only PIR motion sensors are related in this study; therefore, the other sensors are not represented in Fig. 4 (a).



a) A layout of Aruba testbed.

Date	Time	Sensor	Status
2010-11-04	05:40:27.708148	M003	ON
2010-11-04	05:40:29.816524	M005	ON
2010-11-04	05:40:32.342225	M004	ON
2010-11-04	05:40:33.109478	M003	OFF

b) Raw samples from the dataset.

Fig. 4. (a) A layout of Aruba testbed and the locations of the PIR sensors; (b) Samples of the raw dataset and their representation.

B. Resident

According to the CASAS project [26], a single voluntary elderly woman lived in Aruba testbed, and she regularly receives her children and grandchildren during the experimental period. Resident's exact age, cognitive state, daily activity level, etc. are not available in the dataset; therefore, we consider her as a healthy person.

C. Binary Sensors

All binary sensors are equipped with a battery and a ZigBee wireless module; thus they can be installed easily on any place of the testbed and can be connected to a server via wireless mesh network. Events (any detected motion or no motion) are chronologically logged in the server. An event log contains four parts that are date, time, sensor type, and status as shown in Fig. 4 (b). In Fig. 4 (a), PIR motion sensors are labeled as M0XX, and represented by red and grey circles. The red sensors have a small coverage area which sense movements under it, and the grey sensors have wider coverage area that covers most of the room. These motion sensors send a simple "ON" message when motion is present under the coverage area, followed by an "OFF" message shortly after the motion is stopped. Information of the grey sensors are not used in this study, because their coverage area overlaps with the surrounding red ones' coverage area.

D. Dataset

In the raw dataset, 5228655 events were logged from 31 PIR motion sensors, five temperature sensors, and four door switch sensors for 625 days during 2010-2012. Typical samples from the raw dataset is represented in Fig. 4 (b). We can realize that the resident moved from the bed to the bathroom. 27 motion sensors were used in this study and among them four motion sensors (M007, M019, M020, and M027) were ignored; because, they have wider field of view which overlaps with the coverage of surrounding PIR sensors. Supposedly, positions of the PIR motion sensors were deliberately chosen so that resident's common visiting locations are not missed.

E. Technical Specification

We have used a desktop computer that equipped with i7-7700 CPU at 3.6 GHz and GeForce GTX 1080 graphics card. GTX 1080 has a powerful graphical processor unit (GPU) which increases the training and testing speed of the DCNN models.

F. Grid layout

Fig. 5 shows a grid layout of Aruba testbed, which has been employed to find the approximate travel distance and walking speed of the resident in the testbed. Approximate real size of one grid is 0.5 m × 0.5 m. For calculation simplicity, motion sensors are placed in the center of the nearest cell. For example, coordinates of M014 and M009 are (5, 9) and (8, 5), respectively; and a distance between them is 5 which equals to 2.5 m.

G. Dataset Preparation

In the dataset preparation part, sensor data that was collected on days when the resident received visitors are removed from the raw dataset to separate raw dataset that belongs solely to the resident. Then, the resident's episodes are segmented from the raw dataset using an episode segmentation algorithm as shown in Fig. 6. Furthermore, the segmented episodes classified by Vuong's MS pattern classification algorithm were converted into four travel patterns. For DCNN, the classified episodes are converted to episode images.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2771287, IEEE Sensors Journal

1) *Episode Segmentation*

The dataset can be referred as one long list of consecutive movements. The episode segmentation is a process of separating the long consecutive movements into groups of movements that have spatial (start and stop location) and temporal (start and stop time) information. Episode starts when there is any movement is occurred in the testbed after the end of previous episode; and the episode stops if there is no motion for more than *N* seconds (*N* is set to 10 s in this study). Thus, a time period between the stop time of the previous movement and the start time of the consecutive movement must not exceed 10 s if those movements belong to the same episode.

Fig. 5 shows examples of two consecutive direct episodes from the bedroom (M002) to the office room (M026). If we assume that the resident stayed more than 10 s at location M005 without any movement, then the whole path must be divided into two separate direct episodes i.e. M002→M001→M005 and M005→M006→M008→M021→M022 M028→M026.

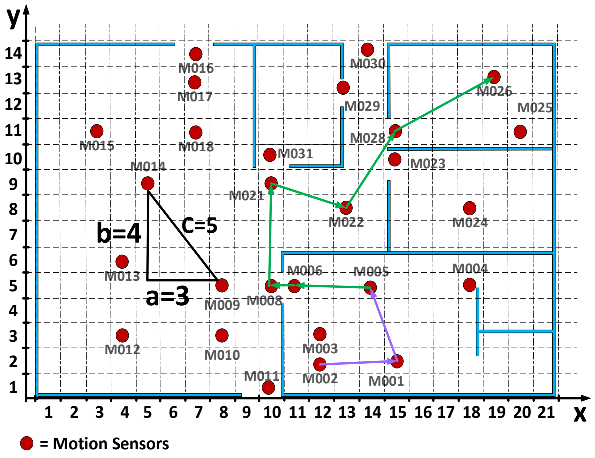


Fig. 5. Grid layout of Aruba testbed.

Fig. 6 shows a pseudocode of an episode segmentation algorithm. The algorithm simply checks the interval time between "ON" messages of PIR sensors (line 2), and once the very first "ON" message has been received or the interval time is more than 10 s (line 4), episode index *i* will be incremented by one and a new episode will be created. Label of the PIR sensor will be the first location of the episode. In case of the interval time is less than 10 s, a new label different from the previous label (line 8) will be appended to the current episode.

```
Algorithm 1: Episode Segmentation
Inputs: Raw PIR sensor signal sequence.
Outputs: Sequence of episodes $E_1, E_2, \dots, E_n$.
0:   $i = 0$                                   # episode index
1:   for all "ON" signals of PIR sensors:
2:       $interval = timestamp_{new} - timestamp_{previous}$
3:       $timestamp_{previous} = timestamp_{new}$
4:       if $interval$ > 10 s or the first "ON" signal:
5:           $i$ ++
6:           $E_i$ = []                        #start a new episode list
7:           $E_i \leftarrow label_{new}$      #append a new label to the list
8:       else if $interval$ < 10 s and $label_{previous} \neq label_{new}$:
9:           $E_i \leftarrow label_{new}$      #append a new label to the list
10:      end if
11:  end for
```

Fig. 6. A pseudocode of an episode segmentation algorithm.

2) *Ground truth*

We employed Vuong's algorithm [29] to classify the segmented episodes into MS patterns, and the classified episodes are used as the ground truth for training and testing the machine learning classification models. In this study, 45000, 24000, 11000 and 3400 episodes were classified as direct, pacing, lapping, and random, respectively. Then, 12000 episodes (3000 for each pattern) were randomly chosen to form a dataset (ground truth).

Fig. 7 shows a histogram of number of episodes in terms of number of movements. From the histogram, we can see that number of episodes are exponentially decaying as the number of movements increase, and the episodes with less than 13 movements compose most of the total 12000 episodes. For instance, more than a half (6486) of the total episodes have only two to four movements. Contrarily, there is only one episode for episodes with more than 26 movements.

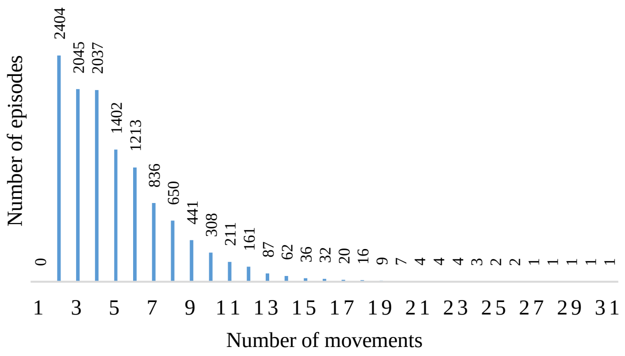


Fig. 7. A histogram of number of episodes in terms of number of movements.

3) *Vuong's MS pattern Classification Algorithm*

Vuong's algorithm determines an episode whether it belongs to the direct, pacing, lapping, or random patterns. The algorithm checks if an episode is one of the first three patterns i.e. direct, lapping and pacing. If the episode does not belong to any of these three patterns, then the episode must belong to the random pattern. First, the algorithm will check if the episode is direct, if not it will check if the episode is pacing or lapping. Finally, the episode is random if it is neither pacing nor lapping. Table 1 shows some samples of travel patterns.

TABLE I.
SAMPLES OF EPISODE PATTERNS

Type	Pattern	Locations	Symbol
Single-pattern episode	**direct**	$L_1L_2L_3$ **or** $L_1L_2L_3L_5L_4$	*D*
	pacing	$L_1L_2L_1L_2L_1L_2$ **or** $L_1L_2L_3L_2L_3L_2L_3$	*P*
	lapping	$L_1L_2L_3L_1L_2L_3L_1$ **or** $L_1L_2L_3L_1L_3L_2L_1$	*L*
	random	$L_1L_2L_1$ **or** $L_1L_2L_3L_1L_2$	*R*
Multi-pattern episode	**pacing**	*P,D* **or** *P,R,P* **or** *P,L,P,R,L,P*	*P*
	lapping	*L,D* **or** *P,L,D,P,L* **or** *L,R,L,R,L*	*L*
	random	*R,D* **or** *D,P,L,R* **or** *L,L,R,R*	*R*

In case of a multi-pattern episode, the episode is considered as a series of single-pattern sub episodes. To classify multi-pattern

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2771287, IEEE Sensors Journal

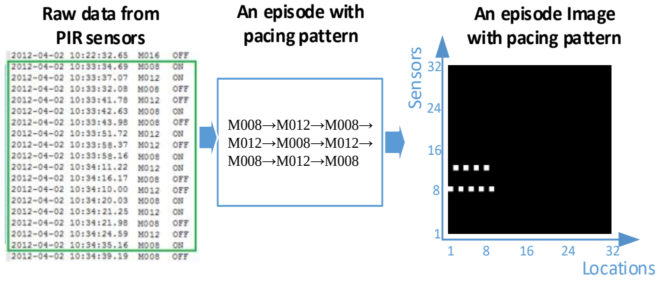
episodes, we count the number of occurrence of each type of inefficient patterns in the entire multi-pattern episode. The episode will be classified as the pattern with the highest number of occurrence. In case of multiple inefficient patterns having the highest number of occurrence, the decision is made based on the severity of the inefficient patterns which is random, followed by lapping and then pacing.

Here, we explain algorithms to check for direct, pacing, and lapping patterns. An episode is considered as direct if there is any repeated location in the episode or any shorter or more efficient path that connects the start and end locations. Checking for pacing pattern is done by looking for the repeated pacing sub-pattern, e.g. ‘L_1L_2’. For lapping patterns, we look for a pattern (e.g., $L_1L_2L_3L_1L_2L_3L_1$ or $L_1L_2L_3L_1L_3L_2L_1$) which has its first location (L_1) repeated in the middle, and has at least three different locations. Lapping can happen in the same direction and opposite direction. The detailed information of the algorithm is reported in Vuong *et al*. [29].

4) *Episode Image*

PIR motion sensors send “ON” message when they sense presence of the motion, then send “OFF” message shortly after the motion is stopped. In this study, episodes consisting of labels of 27 PIR sensors that represents the travel path of the resident. When N = 10, the longest episode has 31 movements and 32 labels. Therefore, all the episodes can be represented in a 32 × 32 binary image.

We propose a novel *episode image* based on binary signals of the PIR sensors. Fig. 8 shows the conversion of the episode image from the PIR sensor data. Suppose, a pacing episode [E = M008, M012, M008, M012, M008, M012, M008, M012, M008] with nine locations is segmented from the raw dataset, then the segmented episode can be converted to a 32×32 binary image.


Raw data from PIR sensors
An episode with pacing pattern
An episode Image with pacing pattern
Sensors
32
24
16
8
1
1
8
16
24
32
Locations

```
2012-04-02 10:22:32.65 M016 OFF
2012-04-02 10:33:34.69 M008 ON
2012-04-02 10:33:37.07 M012 ON
2012-04-02 10:33:32.08 M008 OFF
2012-04-02 10:33:41.78 M012 OFF
2012-04-02 10:33:42.63 M008 ON
2012-04-02 10:33:43.98 M008 OFF
2012-04-02 10:33:51.72 M012 ON
2012-04-02 10:33:58.37 M012 OFF
2012-04-02 10:33:58.16 M008 ON
2012-04-02 10:34:11.22 M012 ON
2012-04-02 10:34:16.17 M008 OFF
2012-04-02 10:34:10.00 M012 OFF
2012-04-02 10:34:20.03 M008 ON
2012-04-02 10:34:21.25 M012 ON
2012-04-02 10:34:21.98 M008 OFF
2012-04-02 10:34:24.59 M012 OFF
2012-04-02 10:34:35.16 M008 ON
2012-04-02 10:34:39.19 M008 OFF
```

M008→M012→M008→
M012→M008→M012→
M008→M012→M008

Fig. 8. Conversion of an episode image from the PIR sensor data.

In the binary image, x-axis represents the locations ranging from 1 to 32, and y-axis represents the number of PIR sensors, so the first location (M008) of the segmented episode is represented at coordinate (1, 8) by a white pixel. Since this episode has nine locations, there are nine white pixels on the episode image. Fig. 9 illustrates three sample episode images for each travel pattern.

H. *Feature Extraction*

Totally 8 features are extracted from each travel episode. The features are: number of movements (F1), time duration (F2), approximate distance (F3), approximate average speed (F4), entropy (F5), repeated locations (F6), repeated movements (F7), and number of pairs of opposite movements (F8). Features F5-F8 are used by Vuong et.al [29] in the machine learning classifiers for the travel pattern classification.

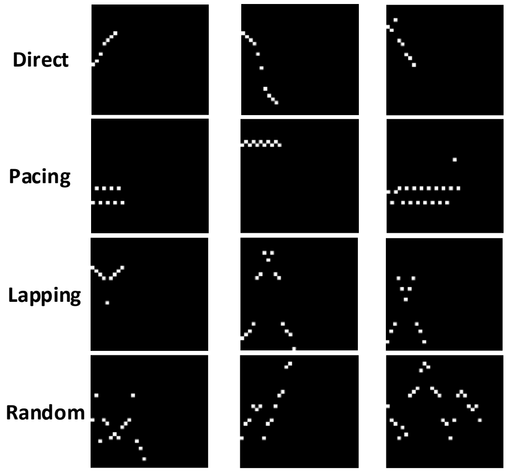


Fig. 9. Samples of episode image, three samples for each pattern.

The entropy, F5, measures the randomness of each episode because; it can be represented unpredictability in a random variable.

The repeated locations, F6, counts the occurrence of the repeated locations in an episode; thus, can be used for classifying direct patterns from the other three travel patterns (i.e. lapping, pacing and random). In case of any repeated location is present in an episode, the episode must be an inefficient travel. Pacing patterns are repetitive movements in back-and-forth; lapping patterns are repetitive in circular manner; and random patterns must have at least one repeated location.

The repetitiveness of directions, F7, counts the occurrence of repeated travel directions in each episode.

The opposite directions, F8, counts the occurrence of pairs of opposite travel directions. For instance, travel directions of two movements of L_1L_2 and L_2L_1 are considered as a pair of opposite travel direction. Feature F8 is needed, because a person can pace and lap in opposite directions.

To explain the mathematical derivation of the features, we assume that an episode with n locations in a chronological order is represented as a vector [29]:

$$E = (L_1, L_2, \dots, L_n) \quad (1)$$

where $L_i \neq L_{i+1}, i = \overline{1, n-1}$ and $L_i \in L$ is a label of the locations, L is a set of all locations.
From the vector E, we find:
The movements:

$$M = ((L_1, L_2), (L_2, L_3), \dots, (L_{n-1}, L_n)) \quad (2)$$

The set of distinct elements in vector E:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2771287, IEEE Sensors Journal

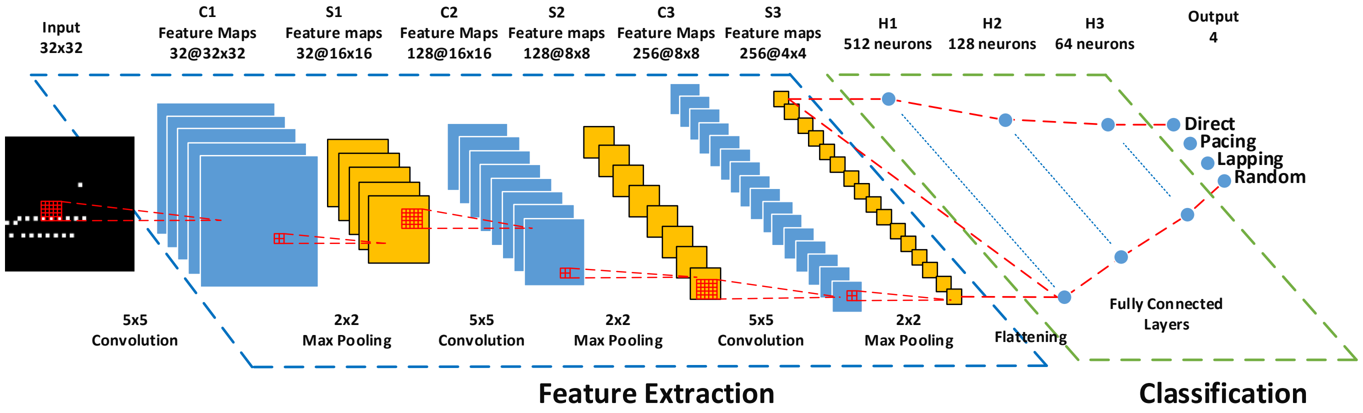


Fig. 10. DCNN architecture.

$$S_E = \{L_i, 1 \ll i \ll n \,|L_i \in L\} \tag{3}$$

The set of distinct elements in vector M:

$$S_M = \{(L_i, L_{i+1}), 1 \ll i \ll n - \ 1\,|(L_i, L_{i+1}) \subseteq M\} \tag{4}$$

The frequency of occurrence of each element in S_E:

$$f_i = (\text{number of occurrences of } L_i \text{ in } E)/n,\ 1 \le i \le n \tag{5}$$

Then, the eight features are calculated as follows:

$$F1 = n - 1 \tag{6}$$

$$F2 = timestamp_{stop} - timestamp_{start} \tag{7}$$

$$F3 = \sum_{i=1}^{n-1} \sqrt{(x_{i,2} - x_{i,1})^2 + (y_{i,2} - y_{i,1})^2} \tag{8}$$

where $x_{i,2}, x_{i,1}$ are x coordinates of two locations in i-th movement; similarly, $y_{i,2}, y_{i,1}$ are y coordinates of two locations in i-th movement.

$$F4 = \frac{F3}{F2} \tag{9}$$

$$F5 = -\sum_{i=1}^{n} f_i log f_i \tag{10}$$

$$F6 = n - \|S_E\| \tag{11}$$

$$F7 = n - 1 - \|S_M\| \tag{12}$$

$$F8 = \left\|\{1 \le i \le n-1 \,|\, \exists\, j, 1 \ll i < j \ll n-1 \ \wedge\ L_i = L_{j+1} \wedge\ L_{i+1} = L_j\}\right\| \tag{13}$$

I. *DCNN Architecture*

The architecture of our proposed DCNN classifier is summarized in Fig. 10. The classifier architecture has three convolutional layers and three fully connected layers. All convolutional layers are followed by max-pooling layers. Each convolutional layer has feature filters of size 5×5. Each max-pooling layer has a pooling window of size 2×2. In the first convolutional layer, an episode image of size 32×32 is convoluted with each one of 32 feature filter, thus creates 32 feature maps of size 32×32. Zero padding is employed in the convolutional operation; therefore, the input image and the feature maps can have the same size.

After the max-pooling operation, output images become two times smaller than the input images since the pooling window is 2×2.

The second convolutional layer receives the output of the first max-pooling layer as inputs and convolute them with 128 feature filters. The third convolutional layer receives the output of the second max-pooling layer as inputs and convolute them with 256 feature filters. Next, the first fully connected layer flattens the output of the third max-pooling layer into a feature vector. The second and the third fully connected layers have 128 and 64 neurons that connected with each neurons of the previous and latter layers. Finally, neurons of the last fully connected layer are connected to all four outputs i.e. direct, pacing, lapping, and random.

Feature maps that generated by 2D convolution are defined as follows:

$$y_j = \frac{1}{1+\exp(b_j + \sum_i k_{ij} * x_i)} \tag{14}$$

where $*$ denotes the convolutional operator, k_{ij} denotes the convolutional filter, x_i denotes i-th input map, y_j is the j-th output feature map, and b_j is a bias.

Max-pooling creates smaller version of input maps while keeping their features; thus, the max-pooling leads to faster convergence. The output map y_i is a result of finding maximum values by overlapping the pooling regions on the input map x_i with a m × m square filter:

$$a_j = \max_{i \in R_j} a_i \tag{15}$$

where a_j denotes the pixel value on the output map, R_j denotes the pooling region on the input map, and a_i denotes an activation in a set $\{a_1, \ldots, a_{|R_j|}\}$.

Then, the flattened outputs of the third max-pooling layer are connected to the fully connected layers. The softmax function is employed to find a probability distribution of the four travel pattern classes:

TABLE II.
10-FOLD CROSS-VALIDATION RESULTS FOR MACHINE LEARNING CLASSIFIERS.

Classifier		**precision**	**recall**	**specificity**	**f1-score**	**accuracy [%]**	**error [%]**	**latency [ms]**
Naïve Bayes	**μ**	0.831	0.825	0.942	0.824	82.51	17.49	< 0.02
	σ	0.01	0.011	0.004	0.011	1.14	1.14	
One Vs Rest	**μ**	0.908	0.905	0.968	0.905	90.46	9.54	< 0.02
	σ	0.005	0.005	0.002	0.005	0.47	0.47	
KNN	**μ**	0.936	0.933	0.978	0.933	93.25	6.75	< 0.02
	σ	0.007	0.007	0.002	0.007	0.65	0.65	
Decision Tree	**μ**	0.936	0.936	0.979	0.936	93.58	6.42	< 0.02
	σ	0.007	0.007	0.002	0.007	0.7	0.7	
SVC	**μ**	0.942	0.938	0.979	0.939	93.81	6.19	< 0.02
	σ	0.004	0.004	0.001	0.004	0.45	0.45	
Gradient Boost	**μ**	0.943	0.941	0.98	0.941	94.06	5.94	< 0.02
	σ	0.007	0.008	0.003	0.008	0.8	0.8	
Random Forest	**μ**	0.947	0.945	0.982	0.945	94.48	5.52	< 0.02
	σ	0.009	0.009	0.003	0.009	0.92	0.92	
DCNN	**μ**	0.979	0.978	0.993	0.978	97.84	2.14	< 20
	σ	0.004	0.004	0.001	0.004	0.379	0.379	

$$p(k) = \frac{g_s}{\sum_{j=1}^{N_a} g_j}, \text{ where } g_i = \max(0, \textstyle\sum_i f_i \cdot w_{ij} + h_j) \qquad (16)$$

where $p(k)$ is the probability of an episode belongs to the k-th class, f_i is a value of i-th neuron in the third fully connected layer, w_{ij} and h_j are coefficients in the softmax function.

J. Performance Validation

In this study, k-fold cross-validation technique is used for validation of the trained classifiers. The cross-validation basically generates k models (k = 10, in this study) where each model is trained on (k-1)/k-th of the dataset and tested on 1/k-th data that is held out. The model accuracy is equal to averaged result of all k results from k different models. In addition, precision, recall (sensitivity), F_1-score, specificity, accuracy, error, and latency are employed as validation metrics.

All episodes are categorized as 'positive' or 'negative' depending on which class they are belonging to. For example, direct episodes are 'positives' for the direct class (pattern), but they are 'negatives' for the other classes (pacing, lapping, and random).

Then, 'positive' episodes that are correctly labelled by the machine learning classifier are counted as true positive (TP), and incorrectly labelled 'positive' episodes are counted as false negative (FN). While 'negative' episodes that are correctly labelled by the machine learning classifier are counted as true negative (TN), and incorrectly labelled 'negative' episodes are counted as false positive (FP). We measure: (i) precision (or Positive predictive value):

$$Precision = \frac{TP}{TP + FP} \qquad (17)$$

which represents the proportion of how many positively classified episodes are TP; (ii) the recall (sensitivity or True Positive Rate):

$$Recall = \frac{TP}{TP + FN} \qquad (18)$$

which evaluates the accuracy of detecting 'positive' episodes; (iii) the specificity (or True Negative Rate):

$$Specificity = \frac{TN}{TN + FP} \qquad (19)$$

which measures how good the classifier is at avoiding false alarms; (iv) F_1-score:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precesion + Recall} \qquad (20)$$

which can interpret a weighted average of the precision and recall, where 1 means the best and 0 means the worst; (v) accuracy:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \cdot 100 \qquad (21)$$

which measures how good the classifier is at detecting both 'positive' and 'negative' episodes; and (vi) error:

$$Error = 100 - Accuracy \qquad (22)$$

which tells how bad the classifier is at detecting both 'positive' and 'negative' episodes.

In addition, we evaluate the latency (computation cost) of the classifiers which is the spend time during the classification of an episode. To report the overall performance metrics of the classifiers, we take the weighted average (or the arithmetic mean) for these seven measures.

IV. EXPERIMENTAL RESULTS

The experimental results of the machine learning classifiers are presented in this section and the performances of the classifiers are explained in terms of seven measures and confusion matrices.

A. Validation Measures

10-fold cross-validation results are shown in Table 2. Weighted average and standard deviation of seven measures (precision, recall, F_1-score, specificity, accuracy, error, and latency) that are averaged performances of four travel patterns are calculated for each classifier.

The classifiers are ordered by the performances in terms of precision, recall, specificity, f1-score, and accuracy. NB has the poorest performance and DCNN has the highest performance. Among the four (precision, recall, specificity, and accuracy) measures, specificity is the highest for all classifiers, which reveals that all classifiers are good at avoiding false alarms. Precision is the second highest measure which is slightly higher or equal to the recall and the accuracy.

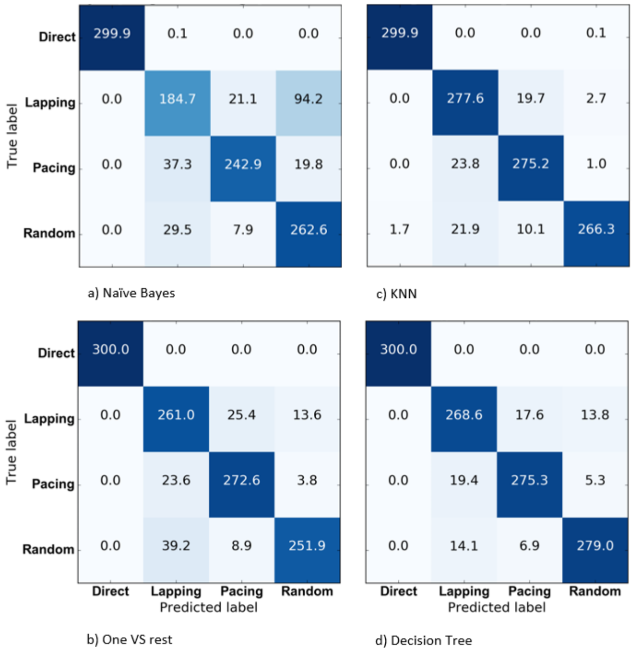
Standard deviation reveals the consistency of classifier's performance throughout 10 different folds; because, each fold has different test set that is not used for testing trained classifiers at the other folds. In terms of accuracy, NB, RF, and GB are the classifiers that have the highest standard deviations of 1.14%, 0.92% and 0.8% respectively, among the others. Thus, these three classifiers have the most inconsistent performances on different folds. However, RF and GB are the second and the third highest after DCNN in terms of overall performance.

DCNN has the lowest standard deviation of 0.379%, which makes DCNN to be the best classifiers compared to the others that yields the most consistent and highest performance on all folds.

In terms of the latency, all classifiers except DCNN have a latency of less than 20 µs; and DCNN has a latency of less than 20 ms. Thus, all classifiers are suitable for the real-time monitoring application.

B. *Confusion Matrices*

Fig. 11 and 12 illustrate confusion matrices of the classifiers. True labels are shown in the y-axis and predicted labels are shown in the x-axis. From the confusion matrices, the performance of the classifiers on each pattern can be observed. Values in the confusion matrices are weighted average (mean) of 10 fold results. Each fold has 1200 episodes for the test set, and 300 episodes for each pattern. Thus, 300 is the highest value in the confusion matrix which means the accuracy of 100%.



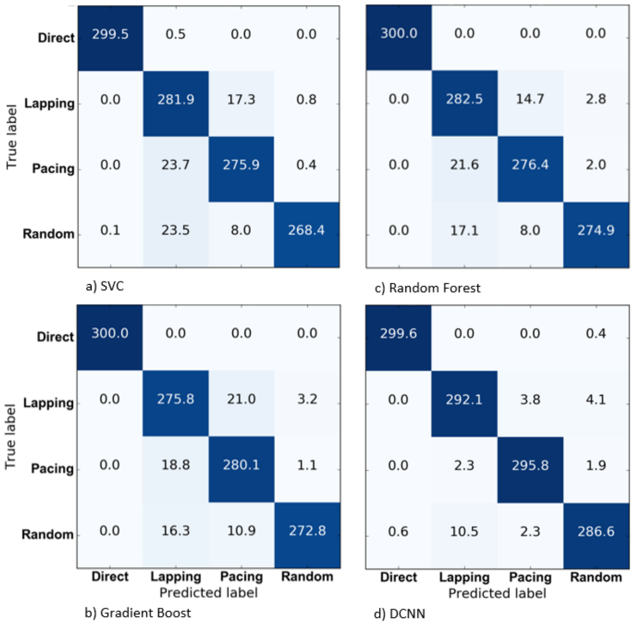
a) Naïve Bayes

c) KNN

b) One VS rest

d) Decision Tree

Fig. 11. Confusion matrices of the machine learning classifiers; a) Naïve Bayes; b) One VS Rest; c) KNN; and d) Decision Tree.



a) SVC

c) Random Forest

b) Gradient Boost

d) DCNN

Fig. 12. Confusion matrices of the machine learning classifiers: a) SVC; b) Gradient Boost; c) Random Forest; and d) DCNN.

For the direct pattern, OVR, DT, RF, and GB are the classifiers that can detect direct pattern with the accuracy of 100%. In addition, DCNN has very high accuracy of 99.86% as well. Moreover, NB, OVR, DC, RF, and GB are perfect for avoiding from FP on direct pattern.

For the lapping pattern, the highest three values are 292.1, 282.5, and 281.9 for DCNN, RF, and SVC, respectively.

For the pacing pattern, the highest three values are 295.8, 280.1 and 276.4 for DCNN, GB, and RF, respectively. Finally, for the random pattern, the highest three values are 286.6, 279 and 274.9 for DCNN, DT, and RF, respectively.
In general, all classifiers are very good at detecting direct pattern, but for pacing, lapping and random patterns, DCNN has the highest accuracy among all classifiers.

C. *DCNN Accuracy vs. Architecture*

We represent seven DCNN designs and their accuracies in Table III. All architectures have the same fully connected networks as shown in Fig. 10. For the episode image, the architecture No. 3, shown in Fig. 10, has the highest accuracy among the other architectures which is three convolutional layers combined with three subsampling.

TABLE III.
DCNN ACCURACY VS. ARCHITECTURE*

No.	DCNN Architecture	Accuracy [%]
1	32C3×3–S2×2–128C3×3–S2×2-256C3×3–S2×2	96.57
2	32C4×4–S2×2–128C4×4–S2×2-256C4×4–S2×2	97.30
3	32C5×5–S2×2–128C5×5–S2×2-256C5×5–S2×2 (Fig. 10)	97.84
4	32C7×7–S2×2–128C7×7–S2×2-256C7×7–S2×2	97.76
5	32C5×5–S2×2–128C5×5–S2×2	97.75
6	32C5×5–S2×2	96.84
7	4C5×5–S2×2-32C5×5–S2×2-64C5×5–S2×2	94.28

*"C" and "S" denote convolutional layer and maxpooling layer, respectively. The architecture is described as "{the number of output maps}C{map size}-S{pooling size}.

Architectures No. 1, 2, 3, and 4 have different feature map sizes increasing from 3×3 to 7×7, and the 3×3 size gave the lowest accuracy of 96.57%. Architecture No. 5, 6, and 7 have fewer layers or fewer feature maps than the architecture No. 3, and their performances are 97.75%, 96.84%, and 94.28%.

V. DISCUSSIONS

In this study, episodes are segmented with 10 s interval. This makes the episodes to become shorter. For example, there is only one episode with 31 movements and there were very few episodes with high number of movements. These few episodes with high number of movements may reduce the accuracy of the classifier since the number of training samples is a vital factor for the performance of the classifier. Perhaps, 20-30 s interval time may be appropriate for increasing the movements of the episodes, thus increasing the training samples for the episodes with higher number of movements.

DCNN yields considerably high performance on MS travel pattern. However, the resident in Aruba testbed was mentally healthy person and there is no annotation of travel pattern or wandering event in the dataset. Thus, we cannot detect any wandering event even that was occurred during the experimental period. However, our proposed classifier can be used for wandering detection in a real-time application.

In addition, the wandering detection is not possible when there is a visitor in the house; because, there is no multiperson tracking algorithm is implemented in the current status.

VI. CONCLUSIONS AND FUTURE WORKS

We have proposed a novel DCNN classifier for device-free non-privacy invasive MS travel pattern detection of elderly people living alone in smart houses. We employed an open dataset collected by PIR motion sensors for two years for training the DCNN classifiers, and 10-fold cross-validation method is employed for the evaluation of the classifiers.

DCNN classifier outperformed the other classical machine learning classifiers with the accuracy of 97.84%. Thus, we think the proposed classifier can be a useful tool for MS travel pattern detection and the proposed method can be used for wandering detection of PwD.

With our best of knowledge, this is the first work proposing an episode binary image converted from PIR sensor logs for DCNN classifier.

In the future, we will apply the proposed classifier to a real-life long-term experiment with PwD for wandering detection purpose. In addition, we will develop an indoor multiperson tracking algorithm which can separate the elder's trajectory from the visitor's trajectory, as a result, wandering detection can be possible even with the visitors.

ACKNOWLEDGMENT

The authors would like to thank CASAS project for their valuable open datasets.

REFERENCES

[1] N. S. Lawand, P. J. French, J. Van Driel, J. J. Briaire, and J. H. M. Frijns, *Sensing Technology: Current Status and Future Trends II*, vol. 12. Springer, 2014.

[2] S. C. Mukhopadhyay, *Next Generation Sensors and Systems*, vol. 16. Springer, 2015.

[3] S. Häfner, J. Baumert, R. T. Emeny, M. E. Lacruz, M. Bidlingmaier, M. Reincke, H. Kuenzel, R. Holle, R. Rupprecht, and K. H. Ladwig, "To live alone and to be depressed, an alarming combination for the renin–angiotensin–aldosterone-system (RAAS)," *Psychoneuroendocrinology*, vol. 37, no. 2, pp. 230–237, Feb. 2012.

[4] K. Kinsella, J. Beard, and R. Suzman, "Can populations age better, not just live longer?," *Gener. - J. Am. Soc. Aging*, vol. 37, no. 1, pp. 19–27, 2013.

[5] K. Raworth, *State of the World 2013*, vol. 461. 2013.

[6] U. S. C. Euromonitor International, "Living Alone Statistics," 2015. [Online]. Available: http://www.statisticbrain.com/living-alone-statistics/.

[7] A. Akl, B. Taati, and A. Mihailidis, "Autonomous unobtrusive detection of mild cognitive impairment in older adults," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 5, pp. 1383–1394, 2015.

[8] S. Simoens, M. Villeneuve, and J. Hurst, "Tackling Nurse Shortages in OECD Countries," Paris, 2005.

[9] B. de Boer, J. P. Hamers, H. C. Beerens, S. M. Zwakhalen, F. E. Tan, and H. Verbeek, "Living at the farm, innovative nursing home care for people with dementia - study protocol of an observational longitudinal study," *BMC Geriatr*, vol. 15, 2015.

[10] S. Abbate, M. Avvenuti, and J. Light, "MIMS: A minimally invasive monitoring sensor platform," *IEEE Sens. J.*, vol. 12, no. 3, pp. 677–684, 2012.

[11] N. K. Vuong, S. Chan, C. T. Lau, S. Y. W. Chan, P. L. K. Yap, and A. S. H. Chen, "Preliminary results of using inertial sensors to detect dementia- related wandering patterns," in *Proc. 37th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2015, pp. 3703–3706.

[12] A. Kumar, C. T. Lau, S. Chan, M. Ma, and W. D. Kearns, "A Unified Grid-based Wandering Pattern Detection Algorithm," in *Proc. 38th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2016, pp. 5401–5404.

[13] K.-J. Kim, M. M. Hassan, S. Na, and E.-N. Huh, "Dementia Wandering Detection and Activity Recognition Algorithm Using Tri-Axial Accelerometer Sensors," in *Proc. of the 4th Int. Conf. on Ubiquitous Information Technologies & Applications*, 2009, pp. 1–5.

[14] K. Makimoto, M. Yamakawa, N. Ashida, Y. Kang, and K.-R. Shin, "Japan-Korea joint project on monitoring people with dementia," in *Proc. 11th World Congress on the Internet and Med.*, 2006, pp. 1–5.

[15] W. D. Kearns, D. Algase, D. H. Moore, and S. Ahmed, "Ultra wideband radio: A novel method for measuring wandering in persons with dementia," *Gerontechnology*, vol. 7, no. 1, pp. 48–57, 2008.

[16] C. Price, "Monitoring people with dementia -- controlling or liberating?," *Qual. Ageing*, vol. 8, no. 3, pp. 41–44, Apr. 2007.

[17] J. Cohen-Mansfield, P. Werner, W. J. Culpepper, M. Wolfson, and E. Bickel, "Assessment of Ambulatory Behavior in Nursing Home Residents Who Pace or Wander: A Comparison of Four Commercially Available Devices," *Dement. Geriatr. Cogn. Disord.*, vol. 8, no. 6, pp. 359–365, Oct. 1997.

[18] D. De Venuto, V. F. Annese, and G. Mezzina, "Remote Neuro-Cognitive Impairment Sensing based on P300 Spatio-Temporal Monitoring," *IEEE Sens. J.*, vol. 16, no. 99, pp. 8348–8356, 2016.

[19] D. Chen, A. J. Bharucha, and H. D. Wactlar, "Intelligent video monitoring to improve safety of older persons," in *Proc. 29th Int. Conf. IEEE Eng. in Med. and Biol. Soc.*, 2007, pp. 3814–3817.

[20] H. H. Dodge, N. C. Mattek, D. Austin, T. L. Hayes, and J. A. Kaye, "In-home walking speeds and variability trajectories associated with mild cognitive impairment," *Neurology*, vol. 78, no. 24, pp. 1946–1952, 2012.

[21] P. N. Dawadi, D. J. Cook, and M. Schmitter-Edgecombe, "Automated Cognitive Health Assessment Using Smart Home Monitoring of Complex Tasks," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 43, no. 6, pp. 1302–1313, Nov. 2013.

[22] B. Das, D. J. Cook, N. C. Krishnan, and M. Schmitter-Edgecombe, "One-Class Classification-Based Real-Time Activity Error Detection in Smart Homes," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 5, pp. 914–923, 2016.

[23] J. Petersen, D. Austin, N. Mattek, and J. Kaye, "Time out-of-home and

cognitive, physical, and emotional wellbeing of older adults: A longitudinal mixed effects model," *PLoS One*, vol. 10, no. 10, pp. 1–16, 2015.

[24] "Wearables have a dirty little secret: 50% of users lose interest - TechRepublic." [Online]. Available: http://www.techrepublic.com/article/wearables-have-a-dirty-little-secret-most-people-lose-interest/. [Accessed: 16-Dec-2016].

[25] T. H. Tan, M. Gochoo, F. R. Jean, S. C. Huang, and S. Y. Kuo, "Front-Door Event Classification Algorithm for Elderly People Living Alone in Smart House Using Wireless Binary Sensors," *IEEE Access*, vol. 5, pp. 10734–10743, 2017.

[26] D. J. Cook, "Learning Setting-Generalized Activity Models for Smart Spaces," *IEEE Intell. Syst.*, vol. 27, no. 1, pp. 32–38, 2012.

[27] C. K. Y. Lai and D. G. Arthur, "Wandering behaviour in people with dementia," *J. Adv. Nurs.*, vol. 44, no. 2, pp. 173–182, 2003.

[28] P.-N. Wang, C.-L. Yang, K.-N. Lin, W.-T. Chen, L.-C. Chwang, and H.-C. Liu, "Weight loss, nutritional status and physical activity in patients with Alzheimer's disease. A controlled study," *J. Neurol.*, vol. 251, no. 3, pp. 314–20, 2004.

[29] N. K. Vuong, S. Chan, and C. T. Lau, "Automated detection of wandering patterns in people with dementia," *Gerontechnology*, vol. 12, no. 3, pp. 127–147, 2014.

[30] E. Batista, F. Borras, F. Casino, and A. Solanas, "A study on the detection of wandering patterns in human trajectories," in *Proc. 6th Int. Conf. on Information, Intelligence, Systems and Applications (IISA)*, 2015, pp. 1–6.

[31] W. Zhao, X. Zhou, H. Ni, and Q. Lin, "A light-weight system for detecting indoor wandering of demented elders living alone," in *Proc. IEEE Int. Conf. on Orange Technologies*, 2014, pp. 53–56.

[32] D. Martino-Saltzman, B. B. Blasch, R. D. Morris, and L. W. McNeal, "Travel behavior of nursing home residents perceived as wanderers and nonwanderers," *Gerontologist*, vol. 31, no. 5, pp. 666–672, 1991.

[33] D. L. Algase, C. Antonakos, E. R. A. Beattie, C. A. Beel-Bates, and L. Yao, "Empirical Derivation and Validation of A Wandering Typology," *J. Am. Geriatr. Soc.*, vol. 57, no. 11, pp. 2037–2045, 2009.

Munkhjargal Gochoo Munkhjargal Gochoo was born in Ulaanbaatar City, Mongolia, in 1984. He received B.S. and M.S. degrees in Electronics Engineering from the Mongolian University of Science and Technology in 2004 and 2005, respectively. He stayed at Department of Electronics, Mongolian University of Science and Technology as a Lecturer during 2005-2011. In 2017, he got his Ph.D. degree from National Taipei University of Technology in electrical engineering field. Currently, he is a Research Assistant Professor of Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan. His main research interests are telecare, eldercare, IoT, machine learning, and deep learning classification algorithms.

Tan-Hsu Tan received B.S. degree from National Taiwan Institute of Technology and M.S. degree from National Tsing Hua University, in 1983 and 1988, respectively, both in Electrical Engineering, and Ph.D. in Electronics Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1998. Since 1988, he has been with Department of Electrical Engineering at the National Taipei University of Technology, Taipei, Taiwan, where he is currently a Full Professor. His research interests are wireless communications, telecare, machine learning, and optimization algorithms.

Vijayalakshmi Velusamy received B.E. degree in Electrical and Electronics Engineering from Bharathiar University, Coimbatore, India, in 2002 and M.Sc. (Eng.) degree in micro and nanotechnology from the University of Ulster, Northern Ireland, U.K., in 2007. She received her PhD in DNA Biosensor Development from University of Limerick, Ireland in 2012. Her current research interests include Cancer Biomarker Detection; Biomedical Device Development; Characterization of Biomaterials; Micro/Nano fabrication; Sensor Interface and Handheld Biosensor Development.

Shing-Hong Liu received B.S. degree in electronic engineering from Feng-Jia University, Taizhon, Taiwan, R.O.C., in 1990, M.S. degree in biomedical engineering from the National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1992, and Ph.D. degree from the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in 2002. August 1994, he has been a Lecturer in Department of Biomedical Engineering, Yuanpei University, Hsinchu, Taiwan, and is an Associate Professor from 2002 to 2008. Now, he is a Professor in Department of Computer Science and Information Engineering, Chaoyang University of Technology. His current research interests are digital signal processing, fuzzy control and designing biomedical instrument.

Damdisuren Bayanduuren received his B.S. and M.S. degrees in communication engineering from Odessa National Academy of Communication, USSR in 1970 and 1971, respectively, and Ph.D. degree in communication engineering from Odessa National Academy of Communication, USSR in 1986. He worked as a Professor, the director of School of Information and Communication Technology (SICT), Mongolian University of Science and Technology (MUST) and the president of MUST. Currently he is a Professor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2771287, IEEE Sensors Journal

in Department of Electronics, SICT. His research interests are telecommunication, wireless networks, speech recognition and synthesis.

Shih-Chia Huang (M’09-SM’14) is a Full Professor with the Department of Electronic Engineering at National Taipei University of Technology, Taiwan, and an International Adjunct Professor with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Canada. He has been named a senior member of the Institute of Electrical and Electronic Engineers (IEEE). He is currently the Chair of the IEEE Taipei Section Broadcast Technology Society, and was a Review Panel Member of the Small Business Innovation Research (SBIR) program for the Department of Economic Development of Taipei City and New Taipei City, respectively. Professor Huang has published more than 50 journal and conference papers and holds more than 50 patents in the United States, Europe, Taiwan, and China. In 2009, he received a doctorate degree in Electrical Engineering from National Taiwan University, Taiwan. Dr. Huang was presented with the Kwoh-Ting Li Young Researcher Award in 2011 by the Taipei Chapter of the Association for Computing Machinery, as well as the Dr. Shechtman Young Researcher Award in 2012 by National Taipei University of Technology. Professor Huang was the recipient of an Outstanding Research Award from National Taipei University of Technology in 2014 and the College of Electrical Engineering and Computer Science, National Taipei University of Technology in 2014-2016. In addition, he is an associate editor of the Journal of Artificial Intelligence and a guest editor of the Information Systems Frontiers and the International Journal of Web Services Research. He is also the Services and Applications Track Chair of IEEE CloudCom conference in 2016-2017, the Deep learning, Ubiquitous and Toy Computing Minitrack Chair of Hawaii International Conference on System Sciences in 2017-2018, and was the Applications Track Chair of IEEE BigData Congress in 2015 and General Chair of IEEE BigData Taipei Satellite Session in 2015-2016. His research interests include intelligent multimedia systems, image processing and video coding, video surveillance systems, cloud computing and big data analytics and mobile applications and systems.