Deep Convolutional Neural Network Classifier for Travel Patterns Using Binary Sensors

Munkhjargal Gochoo, Tan-Hsu Tan
Department of Electrical Engineering
National Taipei University of Technology
Taipei, Taiwan, (R.O.C.)
g.munkhjargal@must.edu.mn
thtan@ntut.edu.tw

Shing-Hong Liu
Department of Information Engineering
Chaoyang University of Technology
Taichung, Taiwan (R.O.C)
shliu@cyut.edu.tw

Vijayalakshmi Velusamy
Department of Electronics Engineering
Manchester Metropolitan University
Manchester, UK
v.velusamy@mmu.ac.uk

Damdinsuren Bayanduuren
Department of Electronics
Mongolian University of Science and Technology
Ulaanbaatar, Mongolia
damdinsurenb@must.edu.mn

Tsung-Yu Liu
Department of Multimedia and Game Science
Lunghwa University of Science and Technology
Taoyuan, Taiwan, (R.O.C.)
tyliu8@gmail.com

***Abstract*— The early detection of dementia is crucial in independent life style of elderly people. Main intention of this study is to propose device-free non-privacy invasive Deep Convolutional Neural Network classifier (DCNN) for Martino-Saltzman's (MS) travel patterns of elderly people living alone using open dataset collected by binary (passive infrared) sensors. Travel patterns are classified as direct, pacing, lapping, or random according to MS model. MS travel pattern is highly related with person's cognitive state, thus can be used to detect early stage of dementia. The dataset was collected by monitoring a cognitively normal elderly resident by wireless passive infrared sensors for 21 months. First, over 70000 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 three 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-invasive, device-free, deep learning, assistive technology, travel pattern, smart house, elder care.**

I. Introduction

According to statistics, the number of people who live alone at home [1]–[6] is increasing worldwide and the elderly prefer an independent and aging-in-place life style due to the high expense of health care services and privacy concern of living with a caregiver [7].

However, elderly cannot live independently if they have dementia. Thus, the early detection of dementia plays crucial role in elderly independent life; because, dementia development can be delayed if the elderly person can be properly treated at the early stage of dementia [8].

Generally, there are three types of monitoring schemes by using: (1) using cameras [16], (2) wearable devices [9]–[15]; (3) binary sensors [8], [17]–[20] such as passive infrared (PIR) sensors, magnetic switches, piezo sensors, passive RFID tags, etc. However, the camera based system is considered as privacy invasive, and wearable devices are difficult to be maintained for a long-term. Thus, device-free and non-privacy invasive systems are the most promising solution for a long-term monitoring applications.

Main objective of this study is to propose Deep Convolutional Neural Network (DCNN) classifier for Martino-Saltzman's (MS) travel patterns of elderly people living alone using an open dataset collected by wireless binary sensors. We employed Naïve Bayes (NB), Gradient Boost (GB), Random Forest (RF) and machine learning classifiers for comparisons.

We utilized the open dataset provided by Center for Advanced Studies in Adaptive Systems (CASAS) project [21], which studies about activity recognition of residents in the smart home using non-privacy invasive binary sensors. The project has the ethical approval from their institutional review board. We summarize our contributions in this study as follows:

- We propose a novel device-free non-invasive MS travel pattern classification method for the elderly people living alone;
- For the first time, we have converted a sequence of passive infrared (PIR) sensor logs into a binary image for the machine learning purpose,
- To our best of knowledge, we have implemented DCNN classifier, which has the highest performance for MS travel pattern classification.

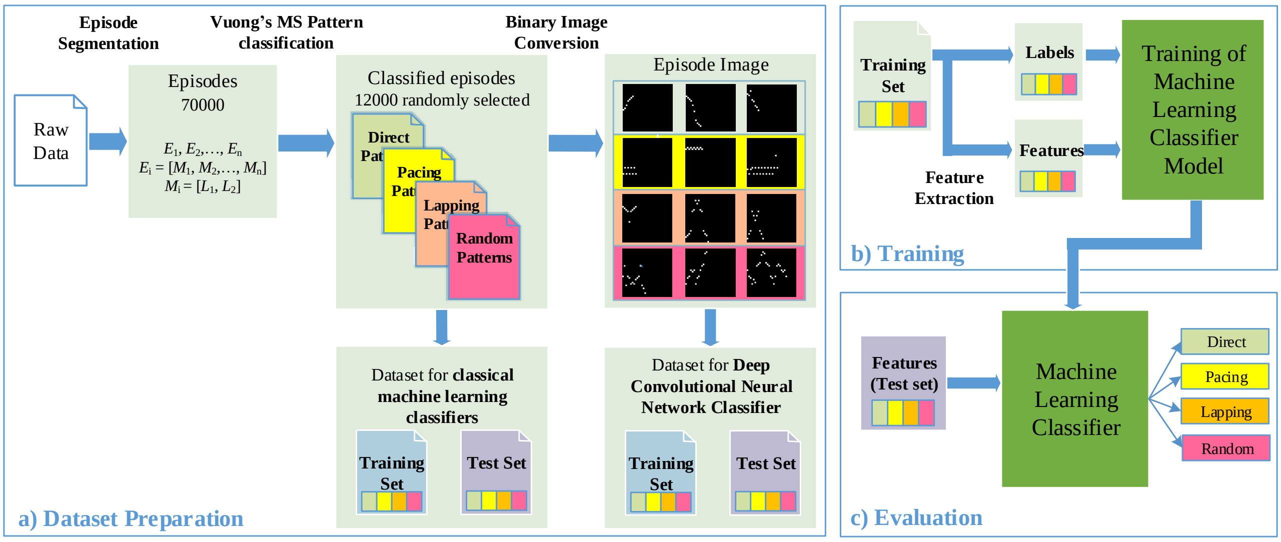


Fig. 2. Framework of the study: a) Dataset preparation; b) Training; and c) Evaluation.

II. BACKGROUND

A. Martino-Saltzman's Travel Patterns of PwD

Researchers [22], [23] have found that Martino-Saltzman's (MS) travel pattern model is a useful tool to detect wandering patterns of PwD. A few studies [11], [24], [25] have employed MS model to monitor wandering of elderly people. Fig. 1 represents MS travel patterns: direct, pacing, lapping, and random.

Vuong *et al.* [24] used a dataset which was offered by Makimoto et al. [13] to evaluate a MS travel pattern detection algorithm. The dataset was collected 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 the algorithm to prepare the ground truth dataset for this study. Studies [11], [25] made wandering detection based on MS travel patterns using GPS, a wireless tag for outdoor, and indoor localization. However, these methods are not a device-free MS travel pattern detection system. This motivates us to propose a novel device-free non-privacy invasive supervised machine learning classifier of MS travel patterns using PIR sensors. PIR sensors cannot identify the person but it is suitable for monitoring single person in the house. Moreover, the resident can be remotely monitored without raising any privacy issues.

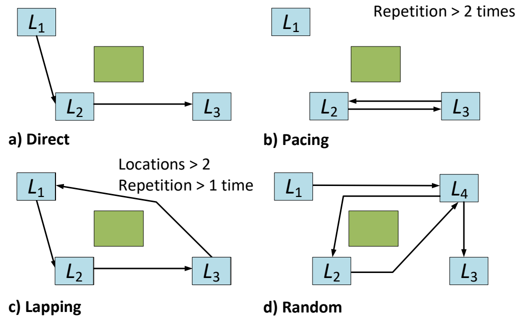


Fig. 1. Martino-Saltzman's travel patterns: (a) Direct; (b) Pacing; (c) Lapping; and (d) Random.

B. Location, Movement, and Episode

The three concepts for a wandering patterns are: location, movement, and episode [24]. A "location" can be represented as coordinates in the grid layout or places such as a bed, a toilet, or a sofa. A "movement" is an action defined as moving from the current location to the next location, thus each movement must have only two locations. An "episode" consists of one or more sequential movements, and each episode has start and stop locations.

If we denote L_1, L_2, L_3, and L_4 as locations. Direct pattern is a single straightforward path from one location to another without diversion or crossing in between. If a travel path intersects at some point, the travel is not considered as direct because it contains redundant sub-path

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.

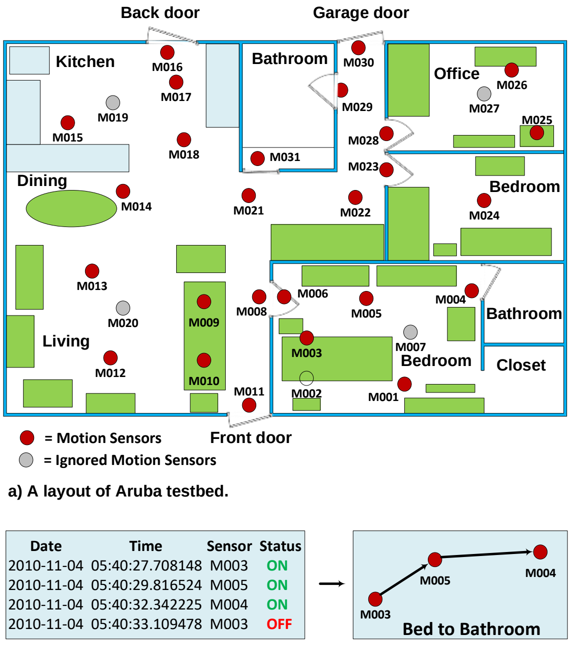
Random is a path, which has multiple locations with no particular order. A random pattern must include at least one location that occurred more than once and it must be non-direct. Because of these two conditions, lapping and pacing patterns can be included in random patterns.

II. METHODS

Fig. 2 illustrates a framework of the proposed MS travel pattern classifiers that consists of dataset preparation, training, and evaluation parts. In dataset preparation part, episodes (E_1, E_2, ... E_n) are segmented from the raw data which is collected via non-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). Then the segmented episodes are classified into four patterns (direct, pacing, lapping, and random) according to Vuong's [24] MS travel pattern classification algorithm. Totally 12000 episodes (3000 for each pattern) were randomly selected from over 70000 episodes for training and testing the machine learning classification models except, DCNN. For DCNN classification model, 12000 classified episodes were converted into 32×32 binary episode images as shown in Fig. 2 (a). Fig. 2 (b) illustrates the training process of the machine learning models with labels and extracted features of the training set as inputs. Fig. 2 (c) represents the evaluation part of the models where the extracted features of the test set are inputted to the trained classification model, and then the model yields the predicted labels for the corresponding features. We have used a desktop computer, which has i7-7700 CPU at 3.6 Ghz speed and NVIDIA GeForce GTX 1080 graphic card. GTX 1080 has a graphical processor unit (GPU).

A. *Smart Home Environment*

Aruba testbed, shown in Fig. 3, is one of the testbeds of CASAS project [21] that is chosen for this study. Fig. 3 (a) illustrates a layout of Aruba testbed. Aruba testbed has a kitchen, a dining area, a living room, an office, two bedrooms, two bathrooms, a pantry, a garage, and a backyard. The testbed is equipped with 31 wireless motion sensors, four door sensors, and four temperature sensors. However, only motion sensors are represented in Fig. 3 (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. 3. (a) A layout of Aruba testbed and locations of the passive infrared motion sensors; (b) Raw samples from the dataset and their representation.

A single voluntary elderly woman lived in the Aruba testbed, and her children and grandchildren regularly visited her during the experimental period. There is no information about the resident's exact age, cognitive state, daily activity level, etc. available in the dataset, thus we consider her as a mentally healthy person.

A. *Binary Sensors*

All binary sensors have 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. Any detected motion or no motion is an event, and events are logged chronologically in the server. Each event log consists of four parts, i.e., date, time, sensor type, and status (Fig. 3 (b)). In Fig. 3 (a), M0XX are PIR motion sensors, represented by red and grey circles that was installed on the ceiling. The red sensors 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 ignored in this study.

B. *Raw Dataset*

During 625 days, 5228655 events were logged in the raw dataset from 31 PIR motion sensors, five temperature sensors, and four door switch sensors for 625. Fig. 3 (b) shows typical samples from the raw dataset. According to the samples, we realize that the resident walked from the bed to the bathroom. Supposedly, positions of the motion sensors were strategically chosen so that resident's common visited locations are not missed.

C. *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. 4. 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.

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. 4 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 $E_1, E_2, \dots, E_n$.
$i = 0$                                   # episode index
for all "ON" signals of PIR sensors:
    $interval = timestamp_{new} - timestamp_{previous}$
    $timestamp_{previous} = timestamp_{new}$
    if $interval > 10$ s or the first "ON" signal:
        $i$ ++
        $E_i = []$                        #start a new episode list
        $E_i \leftarrow label_{new}$      #append a new label to the list
    else if $interval < 10$ s and $label_{previous} \neq label_{new}$:
        $E_i \leftarrow label_{new}$      #append a new label to the list
    end if
end for
```

Fig. 4. A pseudocode of an episode segmentation algorithm.

2) *Ground truth*

We employed Vuong's algorithm [24] 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).

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.

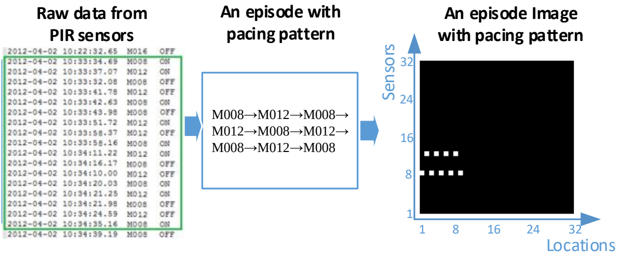
We explain the algorithm to check for direct, pacing, and lapping patterns. An episode is considered as direct if the episode has no repeated location or any shorter or more efficient path that connects the start and end locations. Checking for pacing pattern is done by looking for the repeated pacing 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*. [24].

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. 5 shows a flowchart which generates the episode image. 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.

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. 6 illustrates three sample episode images for each travel pattern.


Raw data from PIR sensors
An episode with pacing pattern
M008→M012→M008→
M012→M008→M012→
M008→M012→M008
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
```

Fig. 5. Flowchart which generates an episode image.

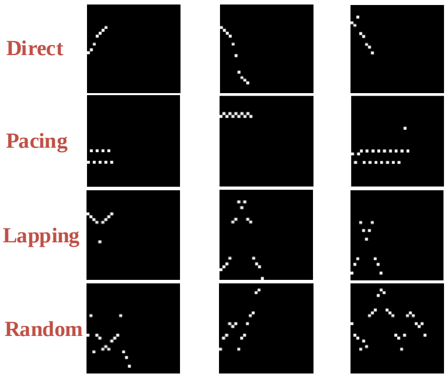


Fig. 6. Samples of episode image, three samples for each pattern.

5) *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 [24] in the machine learning classifiers for the travel pattern classification. F5 measures randomness in each episode. F6, F7, and F8 count, respectively, the occurrence of the repeated locations, the occurrence of repeated travel directions, and the occurrence of pairs of opposite travel directions in each episode. 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 [24]:

$$E = (L_1, L_2, \dots, L_n) \tag{1}$$

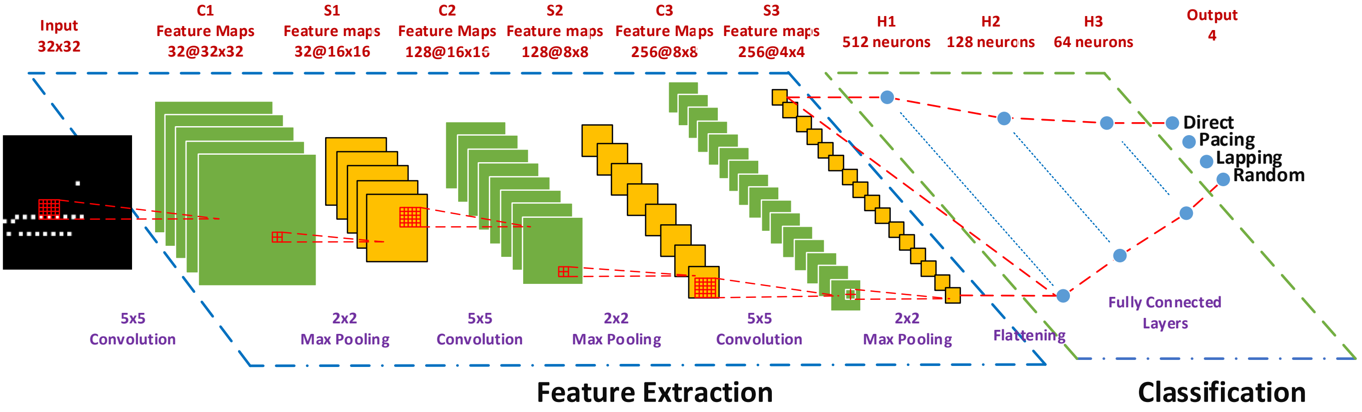


Fig. 7. DCNN architecture.

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:

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

The set of distinct elements in vector E:

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

The set of distinct elements in vector M:

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

The frequency of occurrence of each element in S_E:

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

Then, the eight features are calculated as follows:

$$F1 = n - 1 \qquad (6)$$

$$F2 = timestamp_{stop} - timestamp_{start} \qquad (7)$$

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

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.

$$F4 = \frac{F3}{F2} \qquad (9)$$

$$F5 = -\sum_{i=1}^{n} f_i log f_i \qquad (10)$$

$$F6 = n - \|S_E\| \qquad (11)$$

$$F7 = n - 1 - \|S_M\| \qquad (12)$$

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

D. *DCNN Architecture*

Fig. 7 summarizes the architecture of our proposed DCNN classifier. The model has three convolutional layers and three fully connected layers. Each convolutional layer is followed by subsampling (max-pooling) layer. Convolutional layers have multiple feature filters of size 5×5, and max-pooling layers have 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 used for the convolutional operation, thus the sizes of the episode image and the feature maps can be the same. In the max-pooling layers, outputs are two times smaller than the inputs since the pooling window is 2×2.

The second convolutional layer takes the output of the first max-pooling layer as inputs and convolute them with 128 feature filters. The third convolutional layer takes 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, all neurons of the third fully connected layer are connected to four outputs i.e. direct, pacing, lapping, and random. The softmax function is applied to obtain a probability distribution of the four travel pattern classes:

$$p(k) = \frac{g_s}{\sum_{j=1}^{N_a} g_j}, \text{ where } g_i = \max(0, \textstyle\sum_i f_i \cdot w_{ij} + h_j) \qquad (14)$$

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.

E. *Performance Validation*

k-fold cross-validation technique is employed to validate the performance of the trained models. The results are obtained by weighted averages of seven validation metrics: precision, recall

TABLE I.
WEIGHTED AVERAGE AND STANDARD DEVIATION OF 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	
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	

(sensitivity), specificity, F_1-score, accuracy, error, and latency. In addition, we evaluate the latency of the classifiers, which is the time period that spent for classifying an episode.

III. EXPERIMENTAL RESULTS

Table 1 represents the results of 10-fold cross-validation. For each classifier, weighted average (mean) and standard deviation of seven measures, which are averaged performances of four different patterns (direct, lapping, pacing, and random), are calculated.

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.

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.

III. DISCUSSIONS

DCNN yields considerably high performance on MS travel pattern. However, the 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.

REFERENCES

[1] N. S. Lawand et al., *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 et al., "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 et al., "Can populations age better, not just live longer?," *Gener. - J. Am. Soc. Aging*, vol. 37, no. 1, pp. 19–27, 2013.

[5] Worldwatch Institute, *State of the World 2013: Is Sustaiability Still Possible*, Island Press/Center for Resource Economics, 2013.

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

[7] S. Simoens et al., "Tackling Nurse Shortages in OECD Countries," Paris, 2005.

[8] A. Akl et al., "Autonomous unobtrusive detection of mild cognitive impairment in older adults," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 5, pp. 1383–1394, 2015.

[9] S. Abbate et al., "MIMS: A minimally invasive monitoring sensor platform," *IEEE Sens. J.*, vol. 12, no. 3, pp. 677–684, 2012.

[10] N. K. Vuong et al., "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.

[11] A. Kumar et al., "A Unified Grid-based Wandering Pattern Detection Algorithm," in *Proc. 38th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2016, pp. 5401–5404.

[12] K.-J. Kim et al., "Dementia Wandering Detection and Activity Recognition Algorithm Using Tri-Axial Accelerometer Sensors," in *Proc. 4th International Conference on Ubiquitous Information Technologies & Applications*, 2009, pp. 1–5.

[13] K. Makimoto et al., "Japan-Korea joint project on monitoring people with dementia," in *Proc. 11th World Congress on the Internet and Med.*, 2006, pp. 1–5.

[14] W. D. Kearns et al., "Ultra wideband radio: A novel method for measuring wandering in persons with dementia," *Gerontechnology*, vol. 7, no. 1, pp. 48–57, 2008.

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

[16] D. Chen et al., "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.

[17] H. H. Dodge et al., "In-home walking speeds and variability trajectories associated with mild cognitive impairment," *Neurology*, vol. 78, no. 24, pp. 1946–1952, 2012.

[18] P. N. Dawadi et al., "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.

[19] B. Das et al., "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.

[20] J. Petersen et al., "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.

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

[22] 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.

[23] P.-N. Wang et al., "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.

[24] N. K. Vuong et al., "Automated detection of wandering patterns in people with dementia," *Gerontechnology*, vol. 12, no. 3, pp. 127–147, 2014.

[25] E. Batista et al., "A study on the detection of wandering patterns in human trajectories," in *Proc. 6th Int. Conf. on Information, Intelligence, Systems and Applications*, 2015, pp. 1–6.

[26] D. Martino-Saltzman et al., "Travel behavior of nursing home residents perceived as wanderers and nonwanderers," *Gerontologist*, vol. 31, no. 5, pp. 666–672, 1991.