



SafeMove: monitoring seniors with mild cognitive impairments using deep learning and location prediction

Abdulrahman Al-Molegi^{1,2} · Antoni Martínez-Ballesté¹

Received: 31 March 2021 / Accepted: 18 April 2022 / Published online: 4 July 2022
© The Author(s) 2022

Abstract

Due to society aging, age-related issues such as mild cognitive impairments (MCI) and dementia are attracting the attention of health professionals, scientists and governments. Seniors suffering from such impairments notice a slight deterioration in their cognitive abilities, which may lead to memory loss and behavioural disorders. In consequence, such seniors refrain from doing their everyday outdoor activities. Technology, e.g. smartphones, wearables and artificial intelligence, can provide seniors and their relatives with a variety of monitoring tools. In a nutshell, locations are analysed and, under specific situations, alarms are raised so that caregivers urgently informed. In this context, the discovery and prediction of trajectories and behaviours play a key role in deploying effective monitoring solutions. In this paper, we present a real-time smartphone-based monitoring system, called *SafeMove*, to discover and predict elderly people behaviours by analyzing outdoor trajectories. This is achieved by firstly analysing the elder's mobility data previously collected using the proposed model called SpaceTime-Convolutional Neural Network (ST-CNN) in order to predict the most popular locations he/she might visit in the next time. Based on the predicted locations, the elder can be monitored in bounded region. Time and space-related variables, such as the distance traversed, the direction of the movements and the time spent, are analyzed in our abnormal behaviour detection (ABD) model that takes advantage of recurrent neural networks (RNNs). The effectiveness and the efficiency of our system for predicting the next location and detection the abnormal behaviors are evaluated using different datasets comprising real-world GPS trajectories.

Keywords Wandering · Trajectory analysis · Location data · Monitoring system · Mild cognitive impairments · Dementia

1 Introduction

The aging of the population is one of the most important challenges for public healthcare sector in the twenty-first century. Mild cognitive impairment (MCI) is one of the most prevalent diseases suffered by the seniors. People suffering from MCI (*i.e.* patients) may forget their destination while moving from one area to another. As a result, strange trajectory patterns are obtained and the so-called

wandering episodes occur. Fortunately, technology allows their movements to be continuously tracked, and hence, appropriate real-time assisting services to address their difficulties in navigational tasks can be provided. Otherwise they may get lost, which can cause serious injuries or even their death.

GPS and navigation applications are key enablers to many services offered by mobile devices, from state-of-the-art products to low-cost smartphones. As a result, tracking and monitoring systems have emerged as a good solution to assist elderly people during their outdoor mobility issues. In such systems, the patient makes use of a wearable or a smartphone that is capable of obtaining its location (using GPS technology). In addition, the system is able to warn patients or to send alarms to caregivers. Monitoring systems contribute to self-care and reduce stress on patients' relatives and caregivers as well.

✉ Abdulrahman Al-Molegi
abdulrahmanqasem.al-molegi@alumni.urv.cat

Antoni Martínez-Ballesté
antoni.martinez@urv.cat

¹ Universitat Rovira i Virgili, 43007 Tarragona, Spain

² University of Science and Technology, Sana'a, Yemen

Moreover, cognitive environments and smart cities pave the way to the deployment of advanced assistance services for seniors in the smart health and cognitive health paradigms [37].

Some of the existing tracking and monitoring systems require the patients to interact with smartphone application in a variety of ways (*e.g.* checking in to their destination [13], pushing a panic button [34] or selecting the destination area [42]). However, monitoring systems should take into account the inability of some patients to interact with their devices. Hence, it is essential that monitoring systems nothing is required from the patient except to carry the smartphone.

It is commonly assumed that elderly people follow regular mobility routines, *i.e.* they visit the same locations and walk through the same routes from one area to another. This fact makes it possible to detect the abnormal movements. Notwithstanding, most of the proposals are unable to work well unless enough mobility data are available. For example, to detect the pacing pattern in the wanderer movement, he/she must move back and forth between two points or more. However, the abnormal situation should be discovered in its first appearance to prevent the person from getting lost in advance.

Next location prediction can predict the time when a person is going to be present at particular locations [18]. In other words, it can be used to detect anomalous behaviour such as when a disabled patient or child is expected to be at a certain location but is not. Further action can be initiated such as an emergency call. In addition to that, presence prediction could be useful in other scenarios such as an intelligent postal service.

The convolution technique is shown to be quite effective in exploiting the correlation of various types of data which is considered as the key to the success of CNN for a variety of tasks [22]. CNN is especially adapted to various computer vision tasks because of their ability to abstract representations with local operations. The use of CNN has achieved significant success in various applications such as computer vision [32, 33, 43, 44], speech [7, 19, 39, 48] and natural language processing [6, 45]. A CNN assumes that there is a very specific structure in the data, where inputs that are close to each other are related, whereas inputs that are further away are less related. In images, this makes sense since we normally have patches of similar texture, lighting and colour. In text, words that occur close to each other in a sentence or a paragraph are more likely to share some semantic meaning. In sound, there are similar patterns where the sound spectra at time steps close to each other tend to be similar and particular transitions from one phoneme to another are more common than others in a given language. In people's mobility modeling, the movement flow pattern of individuals from one location to

another is required to extract. One promising approach is to adopt the fast, scalable and end-to-end trainable CNN. Although CNN is originally designed to cope with image data, it can be used for sequence modeling tasks such as location prediction.

At initial stages of dementia and other cognitive deficits due to age-related memory loss, the system can be used to provide information related to early disease diagnosis by assessing their movement behaviour in case of abnormal movements.

1.1 Contribution and plan of the article

We propose a monitoring system that can assist elderly people during their outdoor movements. The system contains two models. The first one is called SpaceTime-Convolutional Neural Network (ST-CNN), which runs convolutional neural networks (CNNs) in a server utilizing the senior's historical movement data. This model is responsible for predicting the locations that the elder will visit. Based on that, the route and the expected time spent to reach that locations are obtained. In our system, all information related to the movement should have a key role in order to identify erroneous behaviours. During a movement, the caregivers are warned in case the patient spends a long time to reach one of the predicted locations or moves to unpredicted locations by changing the routes or still moves in the same area. Furthermore, we demonstrate how the abnormal movements can be detected and how the system assists the elderly people to be safe in real-time movement scenarios. It is based on the second model called abnormal behaviour detection (ABD) model that takes advantage of recurrent neural networks (RNNs) to analyse time- and space-related variables. In order to evaluate the system, three different datasets are used, each one of them with its own descriptors. First, outdoor trajectories from Catalan patients diagnosed with MCI are used. Second, two additional online datasets, which contain trajectories from individuals (not necessarily suffering from MCI) are also used; GeoLife and OpenStreetMap datasets.

The system is autonomous, and hence, no explicit input from the user is required. The system is able to learn about a user's movement behaviour. With the aim of minimising patient interventions or interactions with the monitoring system, our proposal generates the predicted destinations and detects the abnormal behaviour based on deep learning models. In our system, the monitoring is performed in real-time and the abnormal behaviour is immediately detected.

The rest of the paper is organized as the following. Section 2 presents the related work. Our proposal is explained in Sect. 3. Next, we present and discuss the

experiment results in Sect. 4. Finally, the article concludes in Sect. 5.

2 Related work

There are a number of proposals in the literature, some of them are related to methods and heuristics focusing on locations and trajectories, assessed using trajectories datasets, whereas some others address real implementations of prototypes and products.

The earliest works in this field began to exploit the combination of mobile phones and GPS receivers to track elderly people. [31] proposed Opportunity Knocks (OK), to guide and assist people with MCI when they are hesitant about their destinations. Smartphones together with physically separated sensor beacon devices, Bluetooth sensor beacon and General Packet Radio Service (GPRS), were carried by patients. The elderly people were asked to specify where they wanted to go, which is not applicable in case of those with MCI. Our proposed system is autonomous, and hence, no explicit input from the user is required. A Hierarchical Dynamic Bayesian Network model was used to predict the on-going route using their previous routes. When it comes to connectivity and latency, combining various platforms in a robust and effective manner is challenging.

In recent study, [5] proposed an app called SOD which includes frontend software for portable devices and a cloud-based backend system. The frontend collects and sends location data to the backend system for processing. The backend is designed to be flexible enough to allow for the deployment of artificial intelligence techniques to detect wandering and getting lost situations, while being able to process a high volume of data efficiently and consistently. The authors left testing the app with real users, comparing it with other methods and discussing automatic detection of abnormal activity for future work.

A simpler solution addressing dementia is OutCare [42], which raised alarms when significant deviations from the daily routines were found. The system was tested with dozens of participants, but aged under 50 (not from elderly groups) and the deterioration capabilities were not mentioned. However, the lack of elderly participants leaves questions regarding the system validation. Our system is evaluated on dataset contains the daily trajectories of elder individuals suffering from MCI, gathered during the SIMPATIC project.

In [47], the authors proposed a social network of caregivers to locate and secure wandering Alzheimer's disease patients. The proposed system is dependent on the safe-

zone idea. The person with dementia wears a GPS and GSM SIM-equipped wrist-watch tracker that sends location data to a remote server. Caregivers carry a smartphone with a communication management app installed. To check the safety status of a person with dementia, an auto intervention mechanism is used. Through the tracker, a no-voice communication GSM call is initiated to inquire about the state of the person with dementia. By making a similar call through the tracker, the patient can ensure a safe status. When a person with dementia fails to make a safety response call, the system alerts the caregivers. However, wandering behaviour detection systems should take into account the inability of some patients to interact with their devices.

The iRoute system [13] was proposed to track people with dementia during their outdoor movements and assist them in case of disorientation. The system was capable of learning new routes and guide the patients towards learned routes if they were lost. The system followed a Belief-Desire-Intention agent model using the preferences and historical records of wanderers.

Vuong et al. [41] proposed a framework using both classical machine learning classifiers and long short-term memory network (LSTM) to recognize dementia-related wandering patterns. The orientation data are used in the study which is gathered from inertial sensors in mobile devices. Experimental results show that LSTM achieves the best results compared to classical machine learning classifiers. The time series data are utilized to classify the movement characteristics: direct, pacing, lapping and random where each time series is associated with one of the four classes only.

The problem of wandering recognition in indoor environments is addressed in [14]. The Random Forest, Neural Network and the Long Short Term Memory (LSTM) networks are utilized for the wandering behaviour prediction. The results show that the LSTM network achieved the best prediction results. The proposed models are evaluated using datasets that include simulated wandering movements but do not include real-world data from dementia patients.

Lin et al. [24] presented a method to determine if the people with MCI were wandering by searching sharp changes of directions along with their GPS traces. This work was based on the assumption that inefficient patterns (e.g. random, pacing and lapping) have a loop-like locomotion nature, and the direction changes are highly frequent in this kind of patterns.

The SIMPATIC project [26] focused on the development of an autonomous system that monitored real-time trajectories from people with dementia, also counting with

an application for the caregivers that received alarms under certain circumstances. A server that processed the locations received, extracted features from the on-going trajectory, and raised alarms to the caregivers' application when needed. The system was tested with 16 patients diagnosed with early or middle stages of dementia from the area of Tarragona (Catalonia, Spain).

Khodabandehloo and Riboni [16] proposed a collaborative learning approach for recognizing symptoms of cognitive impairment in smart homes. The abnormal movements are identified using a personalized model that takes into account the design of the home and the personalities of the people who live there. The proposed approach is evaluated using dataset that is collected and labeled by researchers for people with dementia, MCI persons and cognitively healthy people. The recognition approach assumes that a single person lives in the home; however, multi-inhabitant scenarios introduce additional challenges.

In Sposaro et al. [38], Bayesian theory was used to calculate a wandering probability. The authors implemented the iWander application, which asked the wanderer if he/she was disoriented when a possible wandering behaviour was detected. In case of disorientation, the application guided the patient to a safe area and then notified caregivers. In contrast to the SIMPATIC solution, iWander needed the interaction of the wanderer with button prompts, which may pose some trouble for elderly.

LaCasa [12] used Markov decision process and contextual information to provide wandering assistance, whereas learned from the trajectories of the wanderers using Bayesian methods. The authors assumed that the individuals were at a known location as long as their smartphones were connected to a known WiFi. This assumption was not always true where individuals might be wandering even in well-known areas.

Lin et al. [25] proposed a method called Isolation-Based Disorientation Detection to detect abnormal trajectories. Patient's trajectories previously collected were modeled as a graph in which the vertices are the frequent visited locations, and the edges are the routes among those locations. Presence of looping inside the graph or deviation of a defined route were considered as potential instances of disorientation.

While tracking elderly people is considered as a violation of privacy rights and loss of independence, the relatives and caregivers consider such tracking as a solution to keep elderly people safe. In this context, researchers proposed balanced solutions to support and consider the views of all parties [8, 20].

3 The proposed system: *SafeMove*

In this section, first we overview our proposed system and then we describe with details all system components.

The overall architecture of the system is shown in Fig. 1. The system contains seven parts. The primary parts are as follows:

1. **Monitoring Unit:** is the main part in *SafeMove* which is responsible for receiving and distributing the data from and to the different system parts. Moreover, each patient movement is analyzed in this unit to determine whether or not he/she is located in a safe area.
2. **Prediction Unit:** is the part that is responsible for predicting people's mobility.
3. **Places Identification Unit:** detects the significant places in the patient movement area and provides mainly the place ID that represents those places.
4. **Abnormal Detection Unit:** is the part that is responsible for detecting the abnormal movement of the patient.

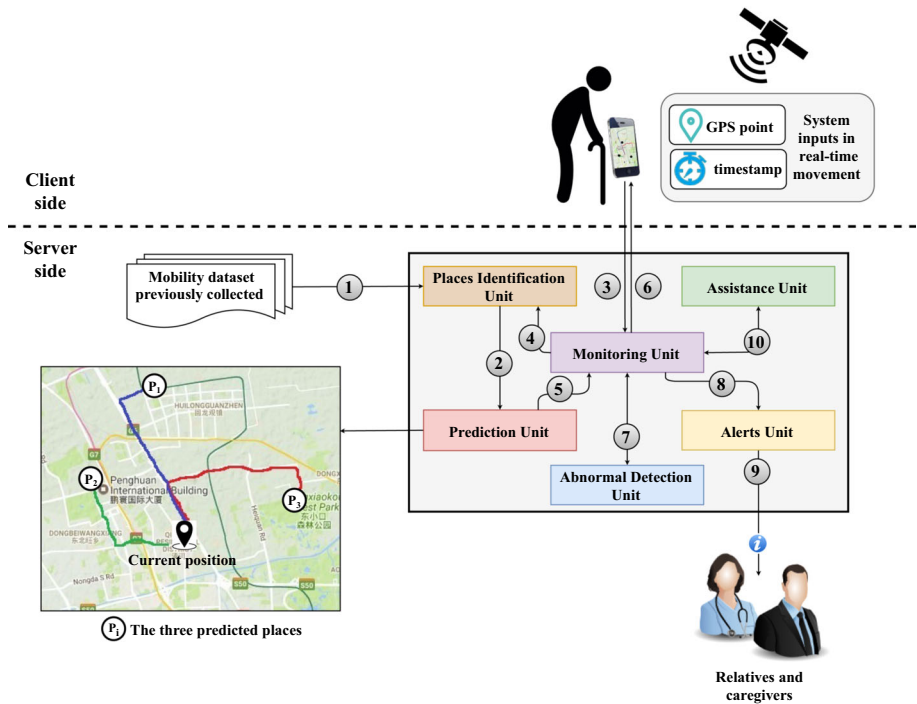
The secondary parts are as follows:

1. **Patient's smartphone application:** is responsible for sending the data related to the patient's positions to a remote server and displaying the outputs of the system.
2. **Alert Unit:** is responsible for sending notifications to the relatives or caregivers.
3. **Assistance Unit:** is responsible for helping elderly people until reaching the desired and safe destination.

The overall functioning of the system is as follows:

1. Mobility data previously collected is converted from GPS coordinates into discrete values associated with specific places, in the Place Identification Unit.
2. The output is then sent to the Prediction Unit, where each patient has a different prediction model.
3. The Patient's Smartphone Application provides the Monitoring Unit with the current GPS coordinates and the timestamp through the available Internet connection.
4. After receiving patient current location (GPS coordinates and the timestamp), the Monitoring Unit sends these information to the Places Identification Unit to provide the place ID, and then to the Prediction Unit to predict the next places.
5. Next, '*N*' locations are sent to the Monitoring Unit which in turn finds the routes and computes the expected time spent to reach that locations.
6. The Monitoring Unit forwards back all the system output information to the Patient's Smartphone

Fig. 1 *SafeMove* system architecture. The server side consists of six parts to track and monitor patient movements to ensure that the patient is located in a safe area. The client side consists of the patient’s smartphone application which sends patient’s locations to the server and displays the system output



Application for displaying through the user interface (will be available for reference for the patient).

7. Every time threshold (5 seconds, for instance), the Monitoring Unit continuously receives the patient current location from Patient’s Smartphone Application. Then, each patient movement is analyzed to detect the abnormal behaviour using the Abnormal Detection Unit.
8. In case of an abnormal movement behaviour, warning notifications are sent to the Alerts Unit.
9. The Alerts Unit will notify the relatives or caregivers by sending alerts.
10. Finally, the Assistance Unit is activated to help the elderly patient to reach a safe place.

Next sections cover with details the individual components. We specially focus on the Prediction, Monitoring and Abnormal Behaviour Detection units.

3.1 Patient’s smartphone application

Due to the widespread usage of smartphones, they have been used as our client hardware. The system is designed to run on Android-enabled devices with GPS in mind. Collecting data from the device’s sensors such as GPS location and timestamps can run as a background service. The system uses the built-in location technologies of the smartphone for location purposes without any mobile user interaction.

3.2 Places identification unit

This is the part of the system that is responsible for mapping each GPS point into the matching place, using place IDs.

Definition 1 A GPS point is a tuple $G = (x_i, y_i, t_i)$ where (x_i, y_i) denotes a GPS coordinate, x_i is latitude, y_i is

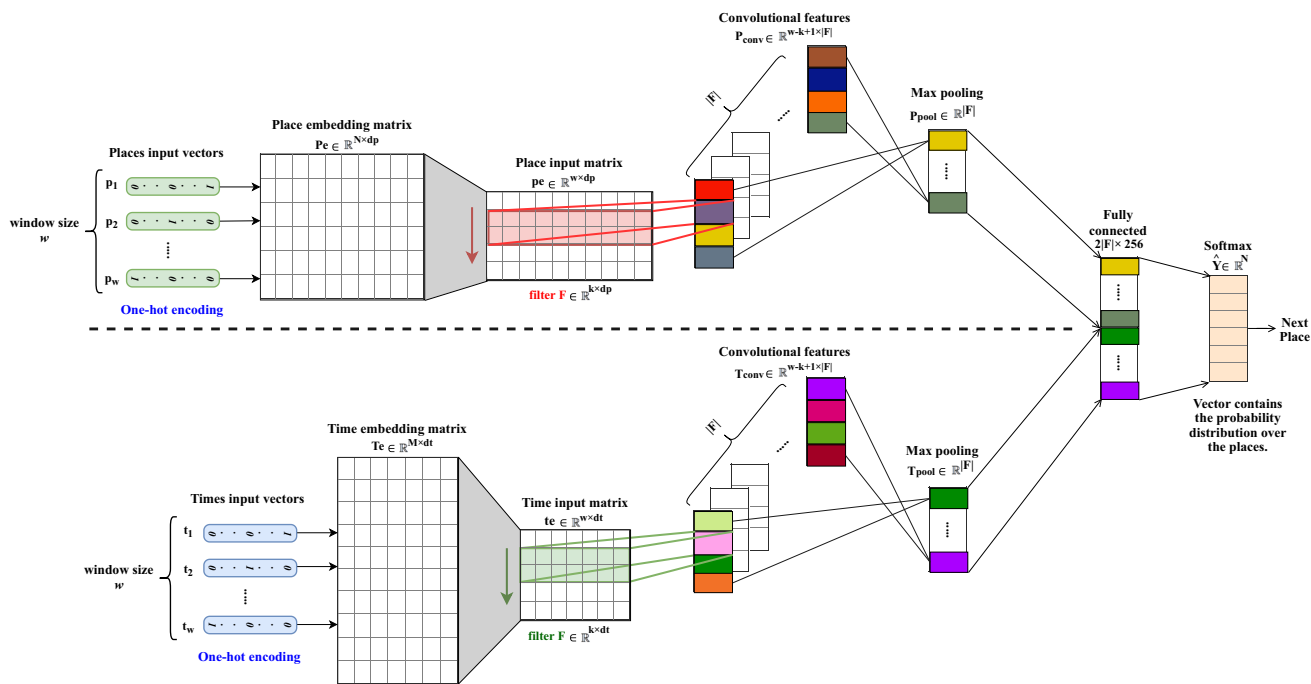


Fig. 2 ST-CNN model architecture. The place and time vectors are the input to the model, while the output is the next place the person will visit

longitude, and t_i is the timestamp when GPS coordinates were recorded.

We convert the sequence of GPS points of each user mobility data into a sequence of places by detecting the *significant places* [3]. Therefore, the mobility routines can be obtained.

Definition 2 We denote the set of places by $\mathcal{P} = \{p_1, p_2, \dots\}$ and the set of movement times by $\mathcal{T} = \{t_1, t_2, \dots\}$. The user trajectories are represented as a sequence of movements $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_n\}$, where n is the length of the user’s trajectories. Each movement \mathcal{M}_i is represented by a tuple (p_i, t_i) where $p_i \in \mathcal{P}$ is the place identifier and $t_i \in \mathcal{T}$ is the corresponding movement time.

To detect the significant places, we use firstly the algorithm proposed by [23]. The significant places represent those spatial regions where the user has stayed for more than a pre-determined time threshold providing that

the distance between the start and end points of the place are under a specific threshold. Secondly, the obtained places are clustered into several geospatial regions using Density-based spatial clustering of applications with noise (DBSCAN) [9], and then each place is given a unique ID. Finally, to formulate the place history of each user, the GPS points located in the same region of the clustered places are replaced by their IDs. Furthermore, during real-time movement, each GPS point sent to the system is converted into a place ID it belongs to.

3.3 Prediction unit

The overall structure of ST-CNN model is shown in Fig. 2. The ST-CNN model (Algorithm 1) is composed of an embedding layer for both places and time (lines 3–6) followed by a convolutional and max pooling layers (lines 7–16), fully connected network (lines 17–19) and a soft-max classification layer (lines 20–23).

Algorithm 1: ST-CNN Model.

```

Input :  $Traj$ : user's trajectories represented as a sequence of movements,
          each movement contains  $(p, t)$ ,  $p$  is place  $\in \mathcal{P}$  and  $t$  is movement
          time  $\in \mathcal{T}$ ,  $w$ : number of visited places, kernels,  $|F|$ : number of
          filters.

Output:  $PP$ : list of predicted places.
1  $PP \leftarrow []$  //Empty list of predicted places.
2 for  $x \leftarrow 1$  to  $Traj.length - w + 1$  do
3   for  $i \leftarrow x$  to  $x + w$  do
4      $pe.append(embedding(Traj[i].p))$  //Get the Place input matrix.
5      $te.append(embedding(Traj[i].t))$  //Get the Time input matrix.
6   end
7   // Places conv
8   for  $k \leftarrow 1$  to kernels do
9     //Convolution, nonlinearity and pooling operations.
10     $P_{conv} \leftarrow Conv1D(|F|, k, pe)$ 
11     $P_{conv} \leftarrow relu(P_{conv})$ 
12     $P_{pool} \leftarrow MaxPool1D(w - k + 1, P_{conv})$ 
13  end
14  // Time conv
15  for  $k \leftarrow 1$  to kernels do
16    //Convolution, nonlinearity and pooling operations.
17     $T_{conv} \leftarrow Conv1D(|F|, k, te)$ 
18     $T_{conv} \leftarrow relu(T_{conv})$ 
19     $T_{pool} \leftarrow MaxPool1D(w - k + 1, T_{conv})$ 
20  end
21   $PT_{concat} \leftarrow Concat(P_{pool}, T_{pool})$  //Concatenation operation.
22   $fc \leftarrow Dense(256, PT_{concat})$  //Fully connected layer.
23   $fc \leftarrow relu(fc)$  //Relu activation function.
24  //Classifier
25   $fc \leftarrow Dense(|\mathcal{P}|, fc)$  //Fully connected layer.
26   $\hat{y} \leftarrow softmax(fc)$  //Softmax activation function.
27   $Idx \leftarrow argmax(\hat{y})$  //Get the place index with highest probability.
28   $PP.append(\mathcal{P}[Idx])$  //Add the places to predicted places list.
29 end
30 return  $PP$ 

```

Given a user u in a place p_i at time t_i , the ST-CNN predicts user's future place \hat{p}_{i+1} on the basis of his/her historical movement records, which is processed in a sliding window from \mathcal{M}_{i-w}^u to \mathcal{M}_i^u , by modeling:

$$P(\hat{p}_{i+1} = p_j | \mathcal{M}_i, \mathcal{M}_{i-1}, \dots, \mathcal{M}_{i-w}) \quad (1)$$

where p_j is a place $\in \mathcal{P}^u$ and w is the number of visited places taken as inputs to the model.

The input layer consists of two vectors. The first vector $p_i \in \mathbb{R}^N$ is the place ID where N is the number of places. The second vector $t_i \in \mathbb{R}^M$ represents the leaving time (in hours) from the place [4] and M is the number of hours per day (i.e. 24 hours). These two vectors are encoded using *one-hot encoding* [11] (i.e. for a given input data, only one out of the vector values will be 1, and all the others are 0).

The two input vectors are passed through an embedding layer in order to learn a meaningful representation of the places and the leaving times input features. This representation enables the model to capture the embedded semantic information about user behaviour and as a consequence improving the prediction performance. The **Place** embedded matrix $Pe \in \mathbb{R}^{N \times d_p}$ represents a set of places where d_p is the dimensionality of the embedded vector. Similarly, $Te \in \mathbb{R}^{M \times d_t}$ is the **Time** embedded matrix that represents a set of times and d_t is the dimensionality of the embedded vector. If w is the number of movements \mathcal{M} taken as inputs to the model, then $pe \in \mathbb{R}^{w \times d_p}$ and $te \in \mathbb{R}^{w \times d_t}$ are the place and time inputs, respectively, as follows:

$$pe_i = p_i \cdot Pe \tag{2}$$

$$te_i = t_i \cdot Te \tag{3}$$

To learn to capture and compose features of movement sequences, the neural network applies a series of transformations to the place and time input matrices using convolution, nonlinearity, pooling and concatenation operations.

A convolution operation involves a *filter* $F_i \in \mathbb{R}^{k \times d_p}$, which is applied to each window size w comes from Pe matrix $\{pe_{1:k}, pe_{2:k+1}, \dots, pe_{w-k+1:w}\}$ where k is the filter height. This operation results in a vector $P_{conv_i} \in \mathbb{R}^{w-k+1}$ which is computed as follows:

$$P_{conv_i} = f(F_i \otimes pe_{i:i+w-1} + b) \tag{4}$$

where \otimes is the element-wise multiplication, and f is a nonlinear function such as a Rectified linear unit (ReLU) and b is a bias vector.

The previous operation is applied using a single filter. For a richer feature representation of the input data, we apply a set of filters that work in parallel generating multiple feature maps. If the number of filters are $|F|$, then a feature map matrix $P_{conv} \in \mathbb{R}^{w-k+1 \times |F|}$ is obtained.

$$P_{conv} = [P_{conv_1}, P_{conv_2}, \dots, P_{conv_{|F|}}] \tag{5}$$

The same procedures are applied to the time input matrix and finally, feature map matrix $T_{conv} \in \mathbb{R}^{w-k+1 \times |F|}$ is obtained.

After passing the convolutional layer outputs (P_{conv} and T_{conv}) through the activation function, it is passed to the pooling layer in order to aggregate the information and reduce the representation. We apply a max pooling operation which simply returns the maximum value to capture the most important feature P_{pool} and T_{pool} .

The output of the pooling layer PT_{concat} is passed to a fully connected softmax layer to obtain the probability distribution over the places.

$$\begin{aligned} P(\hat{p}_{i+1} = p_j | \mathcal{M}_i, \dots, \mathcal{M}_{i-w}) &= \hat{y}_i \\ &= \text{softmax}(PT_{concat} \cdot V + b_o) \\ &= \frac{e^{PT_{concat} \cdot V_i + b_{s_i}}}{\sum_{j=1}^N e^{PT_{concat} \cdot V_j + b_{s_j}}} \end{aligned} \tag{6}$$

where V and b_o are the model parameters to be trained.

Adam optimization algorithm [17], an extension of Stochastic Gradient Descent (SGD), is used to train the network, while Back-Propagation Through Time algorithm [35] is used to compute the gradients. The model parameters are $\theta = [Pe, Te, F, V, b, b_o]$ where Pe and Te are the embedding matrices of the places and time, respectively. F is the set of filters, and b is the convolutional bias. V and b_o are the weight and bias of the softmax layers, respectively. The cost function used is the cross entropy which is defined as:

$$J = - \sum_{i=1}^n y_i \cdot \log(\hat{y}_i) \tag{7}$$

where n is the number of training samples, y is the real user next location, and \hat{y} is the predicted next location probability.

3.4 Monitoring unit

Monitoring unit is the core component that is used to track and monitor movements, in addition to distributing the data from and to the different system parts. Due to the necessity to take action as soon as possible, the system must work in real time. Therefore, every time the system receives a new GPS location, the monitoring unit is executed to ensure that the patient behaves normally.

There are four system statuses: *no moving*, *predicting*, *moving* and *stop*. The system status is set to *no moving* to indicate that the patient is indoor such as at night or inside a building. The patient movement can be detected using the GPS and accelerometer sensor [46]. Therefore, when the patient is moving, the patient’s smartphone application will send the current position to the system. Then, the monitoring unit will operate and forward the current position to the prediction model. Thus, the system status is set to *predicting*. After user progress to one of the predicted location, the system status is set to *moving* to indicate that there is no need to predict and every patient movement will be monitored. After reaching to one of the predicted locations, the system status is set to *no moving*. The status property is set to *stop* where the system will stop working in case the patient in a holiday with his/her relatives, for instance.

Algorithm 2: *SafeMove* system.

```

Input : Traj: a sequence of GPS-Point and Time, Dis_threshold: the
traversed distance for a set of GPS points, Length_threshold:
the direction sequence length to be checked.

Output: Normal or abnormal behaviour.
//Get the first GPS point.
1 FirstGPS  $\leftarrow$  Traj.GPS[1]
//Get the first movement time.
2 FirstTime  $\leftarrow$  Traj.Time[1]
//Get the current place ID.
3 Firstloc  $\leftarrow$  Places_identification(FirstGPS)
//Get the three predicted places.
4 PP1, PP2, PP3  $\leftarrow$  ST-CNN(Firstloc, FirstTime)
//Get the routes to the predicted places.
5 Rot1, Rot2, Rot3  $\leftarrow$  getRoute(FirstGPS, PP1, PP2, PP3)
//Get the distances to the predicted places.
6 Dis1, Dis2, Dis3  $\leftarrow$  getDistance(FirstGPS, PP1, PP2, PP3)
//Get the expected times to reach the predicted places.
7 ET1, ET2, ET3  $\leftarrow$  getExpectedTime(Dis1, Dis2, Dis3)
8 for i  $\leftarrow$  2 to Traj.length do
| //Get the direction between current GPS point and the previous one.
9 | directions  $\leftarrow$  getBearingValue(Traj.GPS[i - 1], Traj.GPS[i])
| //Get spent distance between the current and first GPS point.
10 | SpentDis  $\leftarrow$  getSpentDis(FirstGPS, Traj.GPS[i])
11 | //Get the updated distances between the current GPS point and the
| predicted places.
12 | D̄is1, D̄is2, D̄is3  $\leftarrow$  getDistance(Traj.GPS[i], PP1, PP2, PP3)
| //Get spent time between the current and first time.
13 | SpentTime  $\leftarrow$  getSpentTime(FirstTime, Traj.Time[i])
14 | if direction.length = Length_threshold then
| | //Check the direction sequence for abnormal behaviour.
15 | | if ABD_DetectionModel(directions, Dis_threshold) == True
| | | then
16 | | | return abnormal
17 | | | end
18 | | end
| //Compare the current route with predicted places routes.
19 | if CurrRot  $\neq$  Rot1, Rot2, Rot3 then
20 | | return abnormal
21 | | end
| //Compare the spent distance with the expected distances to reach
| the predicted places.
22 | if SpentDis > Dis1, Dis2, Dis3 then
23 | | return abnormal
24 | | end
| //Compare the updated distances with the actual distances to reach
| the predicted places.
25 | if D̄is1 > Dis1 and D̄is2 > Dis2 and D̄is3 > Dis3 then
26 | | return abnormal
27 | | end
| //Compare the spent time with the expected times to reach the
| predicted places.
28 | if SpentTime > ET1, ET2, ET3 then
29 | | return abnormal
30 | | end
31 end

```

During the movement from a region to another, the state of a user generally changes from indoor to outdoor and vice versa. The abnormal behaviour could happen in the outdoor environment. In this paper, we use the term “sub-trajectory” to refer to the movement between two regions. Each sub-trajectory starts from the first GPS point in the first region until the first GPS point belongs to the next region. Given a trajectory, we divide it into sub-trajectory and each one is labeled as normal or abnormal.

[28] investigated travel patterns of nursing home residents with dementia. Four different travel patterns were identified: direct, random, pacing and lapping. In addition to that, we add two different patterns related to the predicted location: indirect and stopping patterns (Fig. 3).

The movement behaviour of a person is considered as abnormal if one of the following constraints are satisfied (Fig. 3):

1. Time spent to reach the predicted place is more than the expected time.
2. Taking the opposite direction for all predicted places.
3. The trajectory is random, pacing and lapping.

Note that the patient tends to forget the location and as a result, the patient may spend a long time to reach one of the predicted locations. The patient may also change the direction to unpredicted location or may wander in the same area.

The main steps of the *SafeMove* system are described in Algorithm 2 that takes a sequence of GPS points either from patient’s mobility data previously collected or in real time. The place ID of the current location is obtained by

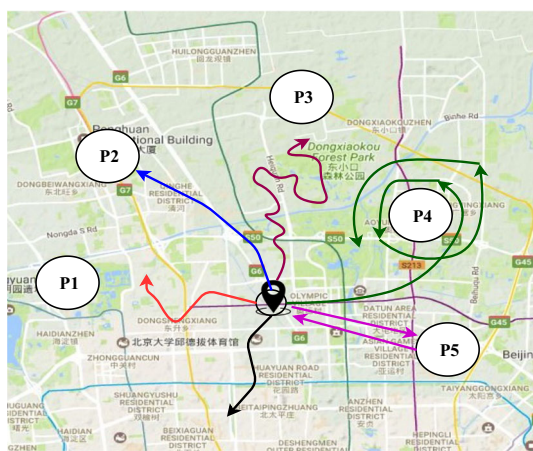


Fig. 3 Movement behaviour types: direct(trajec-tory to P2), random(-trajectory to P3), lapping (trajectory to P4), pacing (trajectory to P5), stopping (trajectory to P1) and indirect patterns (trajectory to unknown place)

calling Places identification unit (line 3). Based on the obtained place ID at a certain time, the ST-CNN model will predict ‘N’ places (line 4), three places are specified in the algorithm. Based on the predicted places, the routes, the distances and the expected times to reach that places are obtained (lines 5-7).

The distance is computed using Haversine Distance, Eq. 8:

$$distance = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right) \tag{8}$$

where r is the radius of the sphere, ϕ is latitude, λ is longitude and $\Delta\phi$, $\Delta\lambda$ are the differences in latitude and longitude, respectively.

After user progress on a route, each user movement is analysed. Three different values are evaluated during the movement: direction, distance and time. The direction is computed between the current GPS point and the previous one (line 9). The spent distance is computed between the current and the first GPS points (line 10), while the updated distances are computed between the current GPS and the predicted places (line 11). Finally, the spent time is computed between the current time and the first GPS point time (line 12).

After that, the system updates the route by matching the current route with all other predicted routes (lines 19-21) taking into account that the patient can take different routes to the predicted places. The obtained spent distance is compared with the distances to the predicted places (lines 22-24) which can be used to detect the change in the direction. If the spent distance value are increased, it means that there is a change in the direction. The updated distances are compared with the distances to the predicted places (lines 25-27) to know exactly the place the patient is very close to. However, if the updated distance still almost similar, it is an evidence of stopping pattern. The obtained spent time is also compared with the expected time to reach the predicted places (lines 24-26). Regarding lines 14-18, they are related to the abnormal detection unit which is described in the next subsection.

Meanwhile, the new position is compared against the predicted places in order to know exactly the place the patient is very close to. During the movement, the patient is represented in a geographical map, thus, relatives or caregivers can keep track of the patient’s progress continuously. Moreover, using various calculated values, several conditions can be performed to ensure that every movement of the patient is fully monitored.

3.5 Abnormal detection unit

This part is responsible for detecting the abnormal behaviour in the elder movements that can occur due to the random, pacing and lapping pattern. For the sake of brevity, such variety of travel pattern will be referred to as random.

In order to detect the random pattern, we use the Bearing Angle formula that used to find the direction between two consecutive GPS locations. Bearing can be defined as an angle or direction between the north-south line of earth (initial bearing) and the line connecting two GPS points, see Fig. 4a. Given two GPS locations: $G_1(\phi_1, \lambda_1)$ and $G_2(\phi_2, \lambda_2)$, the bearing is computed using Eq. 9:

$$\beta = \text{atan2}(\sin \Delta\lambda \cdot \cos \phi_2, \cos \phi_1 \cdot \sin \phi_2 - \sin \phi_1 \cdot \cos \phi_2 \cdot \cos \Delta\lambda) \tag{9}$$

where ϕ is latitude, λ is longitude, and $\Delta\lambda$ is the difference in longitude. The output of this formula is a value between 0 and 360.

As shown in Fig. 4b, we have four intervals that describe the four directions a user could move towards: (45-135), (135-225), (225-315) and (315-360 and 0-45). These intervals are normalized in order to obtain the similar corresponding directions. We used four values: 1 (45-135), 2 (135-225), 3 (225-315) and 0 (315-360 and 0-45). If the normalized bearing values belong to the same interval, it means that the points are located in the same line Fig. 5a; otherwise, the points are in different lines (Fig. 5b).

From the above analysis, we can see that the change in the bearing value intervals can serve as an indicator for random or abnormal behaviour. However, human behaviour mobility is essentially contains change in direction in the daily life. In order to distinguish an abnormal from normal behaviour, we add two threshold variables. The first one refers to the number of GPS readings that the system

should wait before the final decision (*i.e.* normal or abnormal behaviour). The second one is length of the trajectory being evaluated (*i.e.* the distance between the first and last GPS points of that trajectory should be greater than a distance threshold). In this paper, we consider a sequence of 20 GPS points within 100 meters as abnormal behaviour. Those values can be adjusted to meet individual needs.

If the monitoring system read the patient position every 5 seconds, 20 GPS points will be read within almost 1.6 minutes. This is means that if we have four different direction values, then number of different possible sequences are $4^{20} = 1,099,511,627,776$. Therefore, in order to train the abnormal detection model, we build a dataset which contains that number of records. Then, we manually label a portion of that records with normal or abnormal behaviour based on the principles proposed in [28], see an examples in Table 1. Then, we used the labeled dataset to train RNNs model in order to label the remaining unlabeled records. Finally, we used all dataset records to train ABD model using RNNs as follows.

The user directions are represented as a sequence of values $\mathcal{S} = \{s_1, \dots, s_n\}$, where $s_i \in \{0, 1, 2, 3\}$ and n is the length of the user’s directions. Given a user u with a sequence of directions, the model classifies user’s behaviour as normal or abnormal.

Figure 6 shows the graphical illustration of the model. The model contains an input, embedding, recurrent and classification layers as well as inner weight matrices.

The input layer consists of one vector $s_i \in \mathbb{R}^N$ which represents the direction value where N is the number of different direction values. This vector is encoded using one-hot encoding then passed through an embedding layer to produce a vector with d dimension. If the number of direction values is N and the dimensionality of the embedded vector is d , then the dimensionality of the embedded matrix \mathcal{S} is $N \times d$ where \mathcal{S} represents a set of direction values. The embedded vector $se_i \in \mathbb{R}^d$ is given by multiplying the embedded matrix \mathcal{S} and the input vector s_i .

$$se_i = s_i \cdot \mathcal{S} \tag{10}$$

The values of the recurrent layer $h_i \in \mathbb{R}^{d_h}$ are computed as below where d_h is the dimensionality of the recurrent layer vector:

$$h_i = f\left(se_i \cdot WS + h_{i-1} \cdot Wh_{i-1} + bh\right) \tag{11}$$

where $WS \in \mathbb{R}^{d \times d_h}$ and $Wh_{i-1} \in \mathbb{R}^{d_h \times d_h}$ are the weight matrices and $bh \in \mathbb{R}^{d_h}$ is the hidden layer bias. Hyperbolic tangent is used as the nonlinear activation function for the recurrent layer.

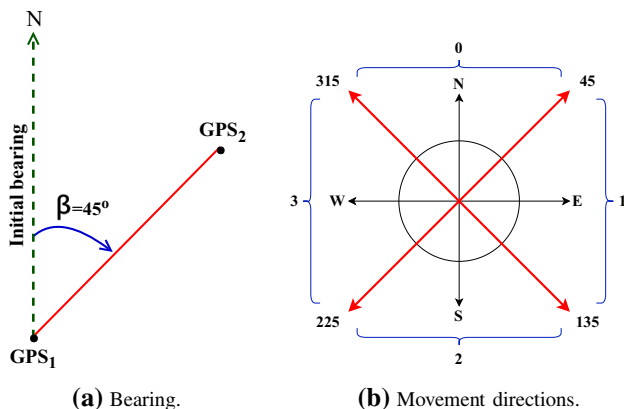


Fig. 4 Bearing and directions

Fig. 5 Changes in directions

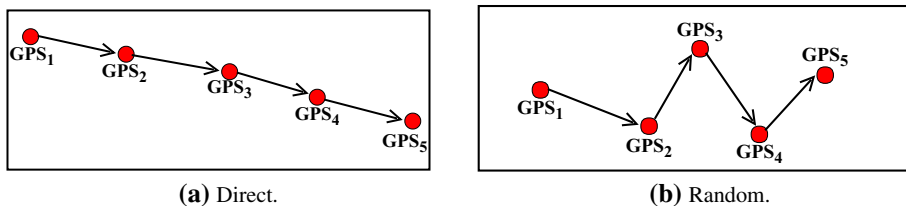


Table 1 Examples of pattern evaluation

Direction sequence	State
1,1,1,2,2,2,1,1,3,3,3,3,2,2,2,0,0,0,0,0	Normal
0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0	Normal
1,0,0,0,0,0,3,3,3,3,3,3,2,2,2,0,0,0,0,0	Normal
1,1,2,2,1,1,2,1,3,0,3,0,3,0,0,2,2,0,0,2	Abnormal
1,2,3,0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3,0	Abnormal
1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1	Abnormal

layer bias. The classification layer is a *Sigmoid* layer which is suitable for this case.

$$g(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

There are three major steps to training a neural network. It begins by doing a forward pass and making a prediction. Second, it uses a loss function to compare the prediction against the ground truth. The error value produced by the loss function is an estimate of how poorly the network is performing. Finally, it uses that error value to calculate the gradients for each node in the network using back-propagation. The cost function used is the cross entropy which is defined as:

$$J = - \sum_{i=1}^n y_i \cdot \log(\hat{y}_i) \tag{15}$$

where n is the number of training samples, y is the real user behaviour, and \hat{y} is the classified one. Optimization is performed using Adam update rule [17], a simple and computationally efficient approach for gradient-based optimization. Back-Propagation Through Time is the training algorithm applied to sequence data that used to update the model parameters in order to minimize the error of the network outputs. In order to minimize the loss, we need to compute:

$$\frac{\partial J}{\partial \theta} = \frac{\partial}{\partial \theta} \left(- \sum_{i=1}^n y_i \cdot \log(g((f(se_i \cdot WS + h_{i-1} \cdot Wh_{i-1} + bh)) \cdot Wh + bo))) \right) \tag{16}$$

where θ is the model parameters, $\theta = [\mathcal{S}, WS, Wh_{i-1}, Wh, bh, bo, h_0]$ and h_0 is the initial vector for the recurrent layer.

3.6 Alert unit

When the system detects patient’s abnormal movement behaviour, it will notify the relatives and caregivers by sending an alerts accompanied by the patient current location (GPS coordinates).

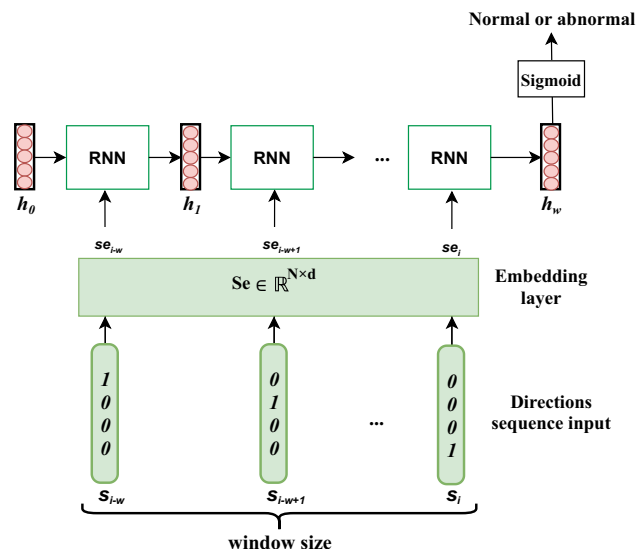


Fig. 6 Abnormal behaviour detection model. The direction sequences are the input to the model, while the output is normal or abnormal behaviour

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{12}$$

The classification layer $\hat{y}_i \in \mathbb{R}$ produces a scalar value ranges from 0 to 1. The value more than 0.5 means normal class, while less than 0.5 is abnormal class. Its values are computed as:

$$\hat{y} = g(h_i \cdot Wh + bo) \tag{13}$$

where $Wh \in \mathbb{R}^{d_h \times N}$ represents the weight matrix between the hidden and output layers and $bo \in \mathbb{R}^N$ is the output

3.7 Assistance unit

This unit is responsible of providing the way that keeps disoriented patient safe. It can help by displaying a map on their phone and creating a routes towards the nearest predicted locations or any location stored in the patient's previous history including the starting location of the movement. Then, the patient follows a series of navigation instructions sounds. The monitoring system will keep track the patient until reaching the desired destination.

While different assistance means can be used, the most important issue is the ability of the patient to understand that assistance type; otherwise, it will be useless. Thus, the assistance type must be added to the system based on what the patient wants. This is one aspect that supports the idea of customizable system.

4 Experimental evaluation

In this section, we aim at demonstrating the performance of the deep learning core of our *SafeMove* system, by conducting tests using three real-world datasets.

4.1 Datasets

There are many online trajectory datasets, but finding datasets containing trajectories for elderly people with/without abnormal movement behaviour is not a straightforward task [27]. We use three real-world datasets in our experiments, i.e., SIMPATIC, GeoLife and OpenStreetMap.

- **SIMPATIC** project dataset [36]: It contains the daily trajectories (from end 2013 to mid 2016) of 18 Catalanian elder individuals suffering from MCI, gathered during the SIMPATIC project. The dataset contains around 2000 trajectories with low sampling (3 minutes), for design reasons.
- **GeoLife** dataset [49]: It is an open GPS trajectory dataset released by Microsoft Research Asia that contains more than 17000 trajectories of 182 individuals from 2007 to 2012. Most of the trajectories were created in Beijing (China), but there are a few in USA and Europe. The trajectories were recorded with high sampling (1-5 seconds or every 5-10 meters per point). This dataset is widely used in many research fields: mobility pattern mining, user activity recognition and location privacy, among others.
- **OpenStreetMap** dataset [29]: It is a collaborative project aiming at creating free editable maps. In addition, an individual can upload a personal trace to the OpenStreetMap public repository. The repository is

being updated continuously, and it already contains more than one million trajectories gathered from thousands of individuals around the world since 2005.

We use the three datasets to evaluate system monitoring. For SIMPATIC dataset, we select normal and abnormal trajectory and label them manually. We evaluate 496 trajectories, 417 of them presenting some kind of abnormal behaviour. Additionally, we can observe that SIMPATIC dataset contains shorter trajectories (elderly does not use to walk large distances), while GeoLife dataset is longer and with more GPS locations due to its high sampling rate. For OpenStreetMap datasets, we chose 16 individuals' GPS traces as our test datasets, while all individuals' GPS traces of GeoLife dataset are used. Since the datasets are not of elderly people, they do not contain abnormal movement patterns. Thus, in order to test the performance of the monitoring system in detecting abnormal patterns in trajectories, we added several trajectories with abnormal patterns manually.

4.2 Experimental setup

The evaluation process is divided into two phases. The first phase is the evaluation of the prediction model. The second phase is the evaluation of the abnormal detection part.

4.2.1 Experimental setup: prediction model

We compare the prediction model performance with four outstanding proposals found in the literature:

- ***n*-MMC** [10] is a classical sequential model which exploits the transition probabilities.
- **NN** [21, 30, 40] has been successfully applied in computer vision, speech recognition, etc.
- **RNN** [15] is widely used for time series prediction.
- **STF-RNN** [1] used embedding representation of space and time as inputs to RNN.
- **GTR** [2] used embedding representation and neural pooling function of input data.

We only use GeoLife dataset to evaluate the prediction model. The parameters of our model are as follows: the window size w is set to 2. The dimensionality of the embedded vector of the places (d_p) and time (d_t) are 160 and 6, respectively. The height of the convolution filters k is set to 2, and the number of convolutional feature maps is 175. We use ReLU activation function and a simple max-pooling function.

Recall and Precision are employed as our evaluation metrics in all experiments in order to assess the efficiency of the prediction models. The **Recall@N** is defined as the ratio between the number of correct predictions (*i.e.*

locations) over the total number of real visited locations. The **Precision@N** is defined as the ratio between the number of correct predictions over the total number of predictions. In our study, we only report $N = 1, 2, 3$.

Supposing that L_u denotes the set of corresponding real visited locations by a user u in the test data, $PL_{N,u}$ denotes the set of top N predicted locations and U is the set of users, the definitions of Precision@N and Recall@N are formulated as below:

$$R@N = \frac{1}{|U|} \sum_{u \in U} \frac{|L_u \cap PL_{N,u}|}{|L_u|} \tag{17}$$

$$P@N = \frac{1}{|U|} \sum_{u \in U} \frac{|L_u \cap PL_{N,u}|}{N|PL_{N,u}|} \tag{18}$$

The model of each user is trained using threefold cross-validation technique where mobility data are partitioned into three sub-data of equal size. The Precision and Recall scores of each case from each user is then calculated, and the final results of all users are averaged.

4.2.2 Experimental setup: ABD model

We compare the detection model performance with an outstanding proposal in the literature. θ_WD [24] is method to determine wandering patterns by searching sharp changes of directions along their GPS traces. In addition to that, the comparison is conducted when using the detection model ABD together with the prediction model ST-CNN.

The most important step is to determine the most suitable parameters of that model in that dataset. Since detecting abnormal behaviour depends on the dataset, the threshold depends on it too. The parameters of our model are as follows: the window size w is set to 20. The dimensionality of the embedded vector of the direction d and the hidden layer d_r are 50 and 20, respectively. Different distance values are used $\delta \in \{10, 50, 100, 150, 200\}$.

When applying the monitoring system given a user movement, it returns a prediction with two possible outcomes: (i) the movement contains normal behaviour which is the default situation, or (ii) the movement contains abnormal behaviour. However, we know in advance whether the trajectory actually contains abnormal movement or not. So, we validate whether the classification is correct, by comparing the prediction of the system monitoring and the trajectory’s label. For this reason, we apply a binary classification. Since the goal of the system monitoring is to detect abnormal movement, we consider that the “positive” class is “abnormal”, and the “negative” class corresponds to “normal”. In practice, a classified trajectory falls into one of the four categories: True Positive (TP), an abnormal trajectory has been successfully classified as

abnormal, False Negative (FN), an abnormal trajectory has been classified as normal, False Positive (FP), which a normal trajectory has been classified as abnormal, and True Negative (TN), a normal trajectory has been successfully classified as normal.

From the evaluation perspective, we consider some statistical measures that can be derived from the confusion matrix after classifying each trajectory such as: Recall, Precision, Specificity, F1-score and Accuracy.

4.3 Experimental results

4.3.1 Prediction model results

Results of our model against five other models are shown in Table 2 in terms of Precision@N and Recall@N with $N = 1, 2, 3$. We can find that the worst results are obtained by n -MMC and NN due to only considering the movement sequences without the time. Furthermore, the results demonstrate the ability of the models that used a RNN structure to successfully analyse movement sequences by taking into account historical dependencies which enables the models to get more accurate results. A great improvement is achieved by STF-RNN though using the internal representations learning of the space and time features. In addition, it can be observed from the table that GTR slightly improves the results compared with STF-RNN due to using the time encoding scheme in the model operations which helps the model to effectively capture the temporal effects. We further notice that our model outperforms STF-RNN, GTR and other baseline models. For instance, ST-CNN outperforms STF-RNN and GTR by 1.1% and 0.64%, respectively, in terms of P@1. This is mainly because that CNN explicitly employs the interactive information, while most other methods only rely on the global information.

Table 2 Performance comparison. Best scores are in bold

Dataset	Model	@1		@2		@3	
		R	P	R	P	R	P
GeoLife	n -MMC	58.9	58.9	78.7	39.3	87	29
	NN	58.1	58.1	82.1	40.9	91.1	30.3
	RNN	67	67	86.5	42.4	92.7	30.8
	STF-RNN	73.3	73.3	87.7	44	93.1	31.1
	GTR	73.6	73.6	88.62	44.31	93.73	31.2
	ST-CNN	74.07	74.07	88.62	44.32	94.2	31.2

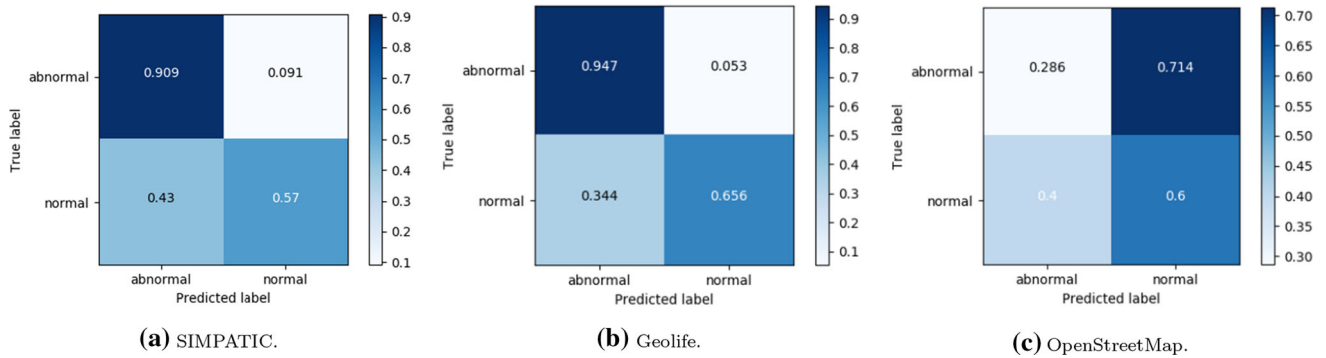


Fig. 7 Confusion matrix of evaluating the abnormal detection model on the datasets. Each entry in column c and row r represents the percentage of action r that was classified to be action c

4.3.2 Abnormal detection model results

Figure 7 shows the confusion matrix obtained from our model classification procedure on the three datasets where main diagonal values represent the correct classifications, whereas off diagonal values are incorrect classifications. It is clear from the figure that our model is able to detect the abnormal trajectories more than the normal ones. For instance, the confusion matrix in Fig. 7a shows that more than 90% from the abnormal trajectories were classified correctly, while the model mostly misclassifies normal trajectories as abnormal. This could be because of the shorter trajectories with low sampling rate.

For GeoLife dataset (7b), the biggest confusion happened when the trajectories are normal but classified as abnormal by the model. Since the GeoLife trajectories contain a huge number of GPS points, they have been classified as abnormal when detecting a change in the direction with a small distance value. On the contrary with the trajectories of OpenStreetMap dataset, our detection model needs more direction changes within a certain distance value to ensure abnormal behaviour.

The confusion matrix in Fig. 7c shows that our model does not work correctly for OpenStreetMap dataset. This is due to that the sampling rate for location acquisition is not fixed and the distances between the consecutive GPS points are almost high. When a low location acquisition rate is used together with long distances values between the consecutive GPS points, a huge number of abnormal trajectories will be missed. In addition to that, since this dataset is imbalanced, most of the correct classifications are normal behaviour trajectories.

These results could be improved by modifying the values of the model parameters (*i.e.*, number of GPS readings and distance between the GPS points). Additionally, training the model with more data would allow for better generalize on its classifications and then improve the

performance. In spite of this, the proposed detection model achieved good performance.

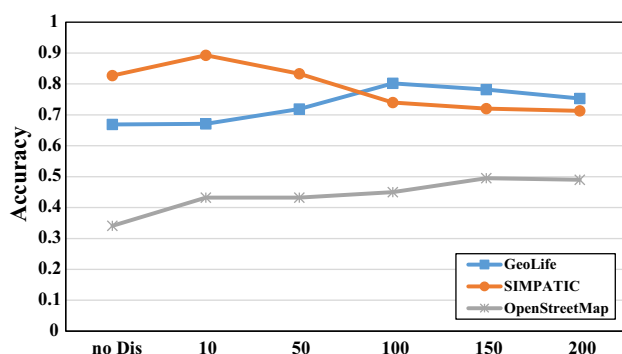
Table 3 illustrates the classification results in terms of Recall, Precision, Specificity, F1-score and Accuracy. It is shown that ABD and ABD+ST-CNN that used a RNN and CNN structures perform better than θ_WD on SIMPATIC and OpenStreetMap datasets. ABD+ST-CNN performs better than ABD. This indicates that using the prediction model is effective in achieving more control on the patient's outdoor movement, which enables the system to obtain more accurate results by considering where the patient is going next. Focusing on the recall attribute, we see that ABD+ST-CNN detects up to 90% of the abnormal trajectories on SIMPATIC and GeoLife datasets. However, when the model detects abnormal, it is true with 67% and 73% of the cases on SIMPATIC and GeoLife datasets, respectively. Regarding OpenStreetMap, we can clearly see that our model is not appropriate for this dataset where recall is 19% and 28.6% with ABD and ABD+ST-CNN, respectively. 38% and 41% from all abnormal predictions are detected with ABD and ABD+ST-CNN, respectively. As mentioned before, any individual can upload a personal trace to the OpenStreetMap public repository. This means that the traces can contain trajectories using transportation means and sport activities which cannot be used for abnormal behaviour detection.

Regarding θ_WD method, it is hard to detect abnormal trajectories, but, when it happens, the trajectory is often abnormal with 48%, 34% and 86.7% on SIMPATIC, GeoLife and OpenStreetMap datasets, respectively. We observe that the poor results are on OpenStreetMap dataset, since it has a low recall and precision values (14% and 34%, respectively). Moreover, we conclude that this method performs well when GeoLife is used. Consequently, we could infer that this method works better when analysing trajectories with high sampling rate.

To investigate the impact of the distance parameter, we conduct several experiments to check the detection model

Table 3 Performance comparison. Best scores are in bold

Datasets	Models	Recall	Precision	Specificity	F1-score	Accuracy
SIMPATIC	θ_WD	0.605	0.484	0.356	0.538	0.48
	ABD	0.73	0.652	0.61	0.689	0.67
	ABD+ST-CNN	0.909	0.679	0.57	0.777	0.74
GeoLife	θ_WD	0.784	0.867	0.879	0.823	0.831
	ABD	0.68	0.66	0.65	0.67	0.665
	ABD+ST-CNN	0.947	0.734	0.656	0.827	0.802
OpenStreetMap	θ_WD	0.143	0.349	0.733	0.203	0.438
	ABD	0.19	0.388	0.7	0.255	0.445
	ABD+ST-CNN	0.286	0.417	0.6	0.339	0.443

**Fig. 8** Distance effect on the datasets

performance with various distance values as shown in Fig. 8. The best performance is obtained when the distance parameter value is 10 for SIMPATIC. However, the best performance reaches its peak when the distance parameter values are 100 and 150 in GeoLife and OpenStreetMap, respectively. This is due to the short trajectories of SIMPATIC compared to the trajectories of other datasets. Since GeoLife and OpenStreetMap trajectories are more dense (low sampling rate) than SIMPATIC, they are more likely to contain cycles.

4.4 Running time

In order to assess whether our models can be used in a real-time system, we measure the exact running time in seconds taken by ST-CNN and ABD models. We simply run the models on trajectories and measure the amount of time for the on-line operations (*i.e.* predicting and detecting the abnormal behaviour). The experiment is repeated 10 times for each model. The average and standard deviation are then calculated. All experiments were conducted on PC with 2.7 GHz Intel Core i5 CPU and 8 GB memory.

The average ST-CNN predicting running time is 0.00037 with ± 0.00006 standard deviation, while the average ABD detecting running time is 0.00003 with

± 0.0000004 standard deviation. It can be noted that the on-line operations are very quick due to the fast computations.

5 Conclusions and future work

We have presented a system called *SafeMove* which utilizes deep learning techniques to provide cognitive assistance to elderly people. It relies on the historical mobility data as a basis for predicting likely locations and then detecting abnormal behaviour. The system in the server runs a Convolutional Neural Networks (CNNs) utilizing an elder historical movement information in order to learn his/her movements. It is responsible for predicting the locations that he/she will visit, the route and the expected time spent to reach that locations. Moreover, three different variables are evaluated during the movement: distance, direction and time. We then developed a model called abnormal behaviour detection (ABD) that take advantage of recurrent neural networks (RNNs) to detect the different abnormal behaviours scenario in real-time. The findings of this study indicate the potential utility of this system for monitoring and detecting wandering behaviour in a real-world scenario with patients suffering from MCI.

The success of any monitoring system depends mainly on the accuracy and availability of mobile user's location information. In addition to that, the quality of the Internet communication between the user's smartphone and the server is essential for the continued operation of the monitoring system.

A major drawback of this system is related to the dataset utilized to build the prediction model. The system assumes that the dataset is available and that it was collected in the period when the patient had no movement problem. If the historical movement dataset is unavailable or contains abnormal behavior, the system will not be able to obtain the user's regular mobility routines. This issue could be addressed by manually determining the significant places by the patient's relatives. Each place can be given a probability to be the next location the patient will visit

depending on the importance of the place and the time of movement. Furthermore, similarities between trajectories belonging to different people can be computed in order to identify the persons with similar mobility routines. On the other hand, the question that might be asked is: How long should movement histories of a patient be stored and used to improve prediction accuracy? The need for large data is essential so that the model will be able to learn the regular mobility routines. In general, the length of time the data should be stored is determined by the ultimate objective. For example, if the data is collected to obtain the user's regular mobility routines and predict the next location the user will visit, it should be stored for a longer period of time. However, if the purpose is to detect immediate wandering behaviour, then the data can be discarded after a shorter amount of time. Regarding the amount of data transferred between patient's smartphone and the server, it can be expected that a considerable connections are to be used, which means more power consumption. In order to reduce high demand for the network and energy consumption, different measures are taken into consideration while building the monitoring system. All processes are running at the system side. The GPS coordinates and the timestamp are sent every specific time from the client side to the system side. Moreover, the caregivers and relatives can modify how often the patient's smartphone sends the GPS coordinates and the timestamp. Last but not least, the system is autonomous, and hence, no explicit input from the patient is required which means less interaction between the server and client side.

In this article, the theoretical framework corresponding to the deep learning components of *SafeMove* has been tested. Our future challenge will be focused on the complete development of an Internet enabled system for *SafeMove* and its corresponding mobile application, so as to build a proof-of-concept and conduct a test with potential users. In addition, we plan to develop the system using a graph-based model that can be applied to obtain the optimal solving of the path between points on the road during patient movements.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the

source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Al-Molegi A, Jabreel M, Ghaleb B (2016) Stf-rnn: space time features-based recurrent neural network for predicting people next location. In: SSCI 2016, IEEE, pp 1–7
2. Al-Molegi A, Martínez-Ballesté A, Jabreel M (2017) Geo-temporal recurrent model for location prediction. In: 20th International Conference of the Catalan Association for Artificial Intelligence, pp 126–135
3. Al-Molegi A, Alsmadi I, Martínez-Ballesté A (2018) Regions-of-interest discovering and predicting in smartphone environments. *Pervasive Mob Comput* 47:31–53
4. Al-Molegi A, Jabreel M, Martínez-Ballesté A (2018) Move, attend and predict: An attention-based neural model for people's movement prediction. *Pattern Recogn Lett* 112:34–40. <https://doi.org/10.1016/j.patrec.2018.05.015>
5. Andersen NS, Chiarandini M, Mauro J (2021) Wandering and getting lost: the architecture of an app activating local communities on dementia issues. arXiv preprint [arXiv:2103.06777](https://arxiv.org/abs/2103.06777)
6. Borjali A, Magnéli M, Shin D, Malchau H, Muratoglu OK, Varadarajan KM (2021) Natural language processing with deep learning for medical adverse event detection from free-text medical narratives: A case study of detecting total hip replacement dislocation. *Comput Biol Med* 129(104):140. <https://doi.org/10.1016/j.combiomed.2020.104140>
7. Celano GGA (2021) A ResNet-50-based convolutional neural network model for language ID identification from speech recordings. In: Proceedings of the Third Workshop on Computational Typology and Multilingual NLP, Association for Computational Linguistics, Online, pp 136–144. <https://doi.org/10.18653/v1/2021.sigtyp-1.13>, <https://aclanthology.org/2021.sigtyp-1.13>
8. Doyle J, Bailey C, Scanaill CN, van den Berg F (2014) Lessons learned in deploying independent living technologies to older adults' homes. *Univ Access Inf Soc* 13(2):191–204
9. Ester M, Kriegel H, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd International Conference on KDD pp 226–231
10. Gambs S, Killijian MO, Cortez Del Prado, MNn, (2012) Next place prediction using mobility Markov chains. Proceedings of the First Workshop on Measurement Privacy and Mobility MPM 2012:1–6
11. Harris D, Harris S (2012) Digital Design and Computer Architecture, Second Edition, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
12. Hoey J, Yang X, Quintana E, Favela J (2012) Lacasa: location and context-aware safety assistant. In: pervasive computing technologies for healthcare (PervasiveHealth), 2012 6th international conference on, IEEE, pp 171–174
13. Hossain S, Hallenborg K, Demazeau Y (2011) Iroute: cognitive support for independent living using bdi agent deliberation. In: trends in practical applications of agents and multiagent systems, Springer, pp 41–50

14. Khaertdinov B, Semerci YC, Asteriadis S (2021) Dementia wandering recognition using classical machine learning and deep learning techniques with skeletal trajectories. In: The 14th pervasive technologies related to assistive environments conference, association for computing machinery, New York, NY, USA, pp 446–452,
15. Kaaniche H (2010) Mobility prediction in wireless ad hoc networks using neural networks. *J Telecommun* 2(1):95–101
16. Khodabandehloo E, Riboni D (2020) Collaborative trajectory mining in smart-homes to support early diagnosis of cognitive decline. *IEEE Trans Emerg Topics Comput*. <https://doi.org/10.1109/TETC.2020.2975071>
17. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint [arXiv:14126980](https://arxiv.org/abs/1412.6980)
18. Krumm J, Brush AB (2011) Learning time-based presence probabilities. In: International Conference on Pervasive Computing, Springer, pp 79–96
19. Kubanek M, Bobulski J, Kulawik J (2019) A method of speech coding for speech recognition using a convolutional neural network. *Symmetry*. <https://doi.org/10.3390/sym11091185>
20. Landau R, Werner S (2012) Ethical aspects of using gps for tracking people with dementia: recommendations for practice. *Int Psychogeriatr* 24(3):358–366
21. Leca CL, Nicolaescu I, Rîncu CI (2015) Significant location detection & prediction in cellular networks using artificial neural networks. *Comput Sci Inform Technol* 3(3):81–89
22. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436
23. Li Q, Zheng Y, Xie X, Chen Y, Liu W, Ma WY (2008) Mining user similarity based on location history. In: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems, ACM, p 34
24. Lin Q, Zhang D, Huang X, Ni H, Zhou X (2012) Detecting wandering behavior based on gps traces for elders with dementia. In: Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on, IEEE, pp 672–677
25. Lin Q, Zhang D, Connelly K, Ni H, Yu Z, Zhou X (2015) Disorientation detection by mining gps trajectories for cognitively-impaired elders. *Pervasive Mob Comput* 19:71–85
26. Martínez-Ballesté A, Budesca FB, Solanas A (2015) An autonomous intelligent system for the private outdoors monitoring of people with mild cognitive impairments. In: Advanced Technological Solutions for E-Health and Dementia Patient Monitoring, IGI Global, pp 137–152
27. Martínez-Ballesté A, Al-Molegi A, Batista E (2018) On the detection of wandering using trajectories datasets. In: 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA), pp 1–6
28. Martino-Saltzman D, Blasch BB, Morris RD, McNeal LW (1991) Travel behavior of nursing home residents perceived as wanderers and nonwanderers. *Gerontologist* 31(5):666–672
29. OpenStreetMap (2018) Openstreetmap: Public gps traces. <http://www.openstreetmap.org/traces>
30. Parija S, Sahu PK, Nanda SK, Singh SS (2013) A functional link artificial neural network for location management in cellular network. In: Information communication and embedded systems (ICICES), 2013 International Conference on, IEEE, pp 1160–1164
31. Patterson DJ, Liao L, Gajos K, Collier M, Livic N, Olson K, Wang S, Fox D, Kautz H (2004) Opportunity knocks: a system to provide cognitive assistance with transportation services. In: International Conference on Ubiquitous Computing, pp 433–450
32. Połap D, Kęsik K, Książek K, Woźniak M (2017) Obstacle detection as a safety alert in augmented reality models by the use of deep learning techniques. *Sensors* 17(12):2803
33. Połap D, Woźniak M, Korytkowski M, Scherer R (2020) Encoder-decoder based cnn structure for microscopic image identification. *Neural Information Processing*. Springer International Publishing, Cham, pp 301–312
34. Rodríguez MD, Navarro RF, Favela J, Hoey J (2012) An ontological representation model to tailor ambient assisted interventions for wandering. In: Proceedings of the AAAI Fall Symposium: Artificial Intelligence for Gerontechnology, pp 34–37
35. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323:533–536
36. SmartHealth (2014) Simpatic: Sistema intel.ligent de monitoritzacio privada i autonoma basat en tecnologies de la informacio i les comunicacions. <https://www.smarthealthresearch.com/projectes/simpatic/>
37. Solanas A, Patsakis C, Conti M, Vlachos IS, Ramos V, Falcone F, Postolache O, Pérez-Martinez PA, Di Pietro R, Perrea DN et al (2014) Smart health: a context-aware health paradigm within smart cities. *IEEE Commun Mag* 52(8):74–81
38. Sposaro F, Danielson J, Tyson G (2010) iwander: An android application for dementia patients. In: Engineering in Medicine and Biology Society (EMBC), 2010 annual international conference of the IEEE, IEEE, pp 3875–3878
39. Taher KI, Abdulazeez AM (2021) Deep learning convolutional neural network for speech recognition: a review. *Int J Sci Bus* 5(3):1–14
40. Vintan L, Gellert JPA, Ungerer T (2004) Person movement prediction using neural networks. In: First Workshop on Modeling and Retrieval of Context, Ulm, Germany 114(4):1–12
41. Vuong NK, Liu Y, Chan S, Lau CT, Chen Z, Wu M, Li X (2020) Deep learning with long short-term memory networks for classification of dementia related travel patterns. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), pp 5563–5566,
42. Wan J, Byrne C, O’Hare GM, O’Grady MJ (2011) Orange alerts: Lessons from an outdoor case study. In: Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on, IEEE, pp 446–451
43. Woźniak M, Silka J, Wiczorek M (2021a) Deep neural network correlation learning mechanism for ct brain tumor detection. *Neural Computing and Applications* pp 1–16
44. Woźniak M, Wiczorek M, Silka J, Połap D (2021) Body pose prediction based on motion sensor data and recurrent neural network. *IEEE Trans Industr Inf* 17(3):2101–2111. <https://doi.org/10.1109/TII.2020.3015934>
45. Wu L, Chen Y, Ji H, Li Y (2021) Deep learning on graphs for natural language processing. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials, Association for Computational Linguistics, Online, pp 11–14, <https://doi.org/10.18653/v1/2021.naacl-tutorials.3>, <https://aclanthology.org/2021.naacl-tutorials.3>
46. Xia H, Qiao Y, Jian J, Chang Y (2014) Using smart phone sensors to detect transportation modes. *Sensors* 14(11):843–865
47. Yüce YK, Gülkesen KH (2013) Development of a social support intervention with a network of caregivers to find wandering alzheimer’s patients as soon as possible: A social computing application in healthcare. In: 2013 8th International Symposium on Health Informatics and Bioinformatics, pp 1–8, <https://doi.org/10.1109/HIBIT.2013.6661678>
48. Zacarias-Morales N, Pancardo P, Hernández-Nolasco JA, Garcia-Constantino M (2021) Attention-inspired artificial neural networks for speech processing: a systematic review. *Symmetry*. <https://doi.org/10.3390/sym13020214>

49. Zheng Y, Xie X, Ma W (2010) Geolife: a collaborative social networking service among user, location and trajectory. *IEEE Data Eng Bulletin* 2(33):32–40

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.