Appl Intell
DOI 10.1007/s10489-017-1062-5

An improved extreme learning machine model for the prediction of human scenarios in smart homes

Zaineb Liouane[1,2] · Tayeb Lemlouma[2] · Philippe Roose[3] · Fréderic Weis[2] · Hassani Messaoud[1]

Abstract One of the main objectives of smart homes is healthcare monitoring and assistance, especially for elderly and disabled people. Therefore, an accurate prediction of the inhabitant behavior is very helpful to provide the required assistance. This work aims to propose a prediction model that satisfies the accuracy as well as the rapidity of the learning phase. To do so, we propose to improve the existing extreme learning machine (ELM) model by defining a recurrent form. This form ensures a temporal relationship of inputs between observations at different time steps. The new model uses feedback connections to the input layer from the output layer which allows the output to be included in the long-term prediction. A recurrent dynamic network, with feedback connections of the output of the network, is proposed to predict the future series representing future activities of the inhabitant. The resulting model, called Recurrent Extreme Learning Machine (RELM), provides the ability to learn the human behavior and ensures a good balance between the learning time and the prediction accuracy. The input data is based on the real data representing the activities of persons belonging to the profile of first level (i.e. P_1) as measured by the dependency model called *Functional Autonomy Measurement System* (SMAF) used in the geriatric domain. The experimental results reveal that the proposed RELM model requires a minimum time during the learning phase with a better performance compared to existing models.

Keywords ELM · RELM · Elderly · Behavior prediction · Smart home · Accuracy · Time series prediction

✉ Zaineb Liouane
liouane.zaineb@yahoo.fr

[1] LARATSI, Monastir University, Ibn El Jazzar, Monastir, Tunisia

[2] IRISA, Rennes I University, Lannion, Rennes, France

[3] LIUPPA/T2i, Pau and the Adour Countries University, Anglet, France

1 Introduction

Every year, the number of the elderly increases considerably. Aged people represent a growing share of the world population [16]. Since 2012, this number has increased to almost 810 million [17]. In order to meet the growing demand of helping this fragile population, current research projects focus on the design of intelligent home environments that consider the monitoring of the person's behavior in order to provide services and help when it is necessary. A smart home environment enables the elderly to live independently and avoid recourse to care centers and institutions. This way improves the quality of life (i.e. QoL), provides financial savings, reduces the strain on aged care centers and provides relief for family caregivers. To ensure an efficient comfort that is adapted to the elderly preferences, the smart home should be able to acquire and use the knowledge about the inhabitant. These system capabilities are based on the effective behavior prediction models according to the observations, the experiments, and the scientific reasoning. This reasoning uses the acquired knowledge to train the prediction techniques and identify the future activities achieved by the occupant.

In this growing trend of smart environments research, the Artificial Intelligence (AI) is often used. AI provides the ability of a computer to perform tasks such as reasoning and learning what human intelligence is capable of doing. Many studies in AI have focused on context modeling, action prediction, motion feature extraction, and machine learning algorithms that are performed using AI tools and techniques. Machine learning has been brought to pervasive applications in order to create an intelligent environment able to have some cognition of the environment and react to the occurring events. Consequently, machine learning and intelligent systems represent good technological candidates to address the main problems of smart environments. Indeed, many machine learning techniques have been used in human daily activity prediction such as Bayesian Networks, Fuzzy Learning, Reinforcement Learning, and Artificial Neural Network (ANN). ANN has been widely applied in the recent smart home search for monitoring, clustering, identifying, and predicting tasks based on the data extracted from several sensors [12, 15, 18–21].

Several ANN models such as Single-hidden Layer Feedforward Neural Network (SLFN), Back Propagation Neural Network (BPNN), and Elman have the interesting properties of extracting the exact pattern between the input and the output data. In addition, ANN provides the system with the ability to learn the behavior of the data. Despite the satisfying results obtained by ANN in the prediction, identification, and monitoring of the daily activities of persons evolving in a smart environment, ANNs present some drawbacks. Mainly, they are sensitive to the selected learning algorithm and parameters and require a long time to achieve a reliable network. To overcome such drawbacks, Huang et al. [5] have proposed an improved algorithm for single-hidden layer Feedforward neural network (SLFN), called *Extreme Learning Machine* (ELM). The algorithm randomly generates the connection weights and the bias between the input layer and hidden layer in the learning step. ELM offers significant advantages such as the fast learning speed, ease of implementation, and minimal human intervention. Due to its remarkable efficiency, simplicity, and impressive generalization performance, ELM was recently applied in a variety of domains, such as biomedical engineering, computer vision, system identification, control, and robotics [14].

Several researchers have demonstrated the importance and the effectiveness of the ELM learning algorithm in different areas. A detailed study concerning the extreme learning machine has been presented and discussed by Huang et al. in [14]. The authors have proposed an overview of ELM from a theoretical and practical perspective. They define several improvements of ELM which further improve its stability, scarcity, and accuracy under general or specific conditions. Wang et al. [13] have proposed a new variant of ELM called *Self adaptive Extreme Learning Machine* (Sa-ELM) to overcome the weakness of ELM regarding its selection of hidden neurons. In order to select the best neuron number in the hidden layer for the training process, Wang et al. have proposed connection weights and thresholds between the input layer and hidden layer. They used a self-adaptive mechanism in the training process to constantly obtain a unique and optimal solution. Thereafter, the Sa-ELM was used to solve the *Italian wine and iris classification problems* [13]. For evaluating the performance of the Sa-ELM, the authors performed a comparison between Sa-ELM and the traditional models. They conclude that Sa-ELM has a faster learning speed and better generalization performance. Yage et al. [10] have proposed a self-adaptive differential evolution algorithm and extreme learning machine (SADE-ELM) for the water quality evaluation and compared the rates of accuracy related to the SADE-ELM, ELM, BPNN, and SVM. This research found that the SADE-ELM provides higher prediction accuracy in the water quality evaluation. In addition, by comparing the learning time of different methods, the study concluded that the ELM is much faster than other methods. Bin et al. [3] compared the echo state network (ESN) and the extreme learning machine (ELM) on nonlinear prediction problems. They concluded that the learning time of ESN is more than ELM. Furthermore, in most cases for sufficient training samples, the ELM algorithm provides a better generalization performance. The authors of [3] found that the performance of ELM is sensitive to the number of hidden neurons. Janakiraman et al. [6] used the ELM to predict the homogeneous charge compression ignition (HCCI) behavior. The authors found that the ELM is a very fast neural network learning algorithm and needs only one iteration process. Qu et al. [7] have proposed to use the ELM algorithm in mammographic risk analysis and found that it was more accurate than other classifier learning. Sridevi et al. [8] compared the accuracy and the quickness of ELM algorithm and Probabilistic Neural Networks (PNN) algorithm in handwritten recognition. Wang et al. [9] analyzed the ELM algorithm with an attractive balance between computational time and generalization performance in the prediction of protein-protein interaction sites compared with Support Vector Machine (SVM) algorithm. Bazi et al. [1] combined the ELM algorithm and differential evolution (DE) algorithm and applied the hybrid algorithm in the classification of hyper-spectral images. In order to improve the prediction accuracy and optimizing the quality of laser brazing, Rong et al. [4] have proposed a novel hybrid model based on extreme machine learning and genetic algorithm (GA). GA is used to optimize the final outputs considering the fitness values that are achieved from the prediction procedure using ELM. The authors compared the results of ELM-GA with the back-propagation algorithm neural network and GA and they concluded that ELM-GA is more accurate and more stable.

As we can observe from the previous researches, the ELM model had a great and rapid success in different areas. However, ELM is still sensitive to the application that requires large datasets modeling [25]. The same situation occurs in smart home applications where the data is generated from different sensors and required to model the behavior of persons. Moreover, it is difficult to exploit the prediction of ELM using an important volume of time series coming from real data and which require important storage capacities [11].

In this work, in order to guarantee an efficient prediction regarding the behavior of persons in their smart homes, it is necessary to consider an important set of data that requires large storages to model and predict future values of time series. For this reason, we focus on the improvement of ELM by carrying it out in a recurrent way. Indeed, in order to be able to process large sets of data and effectively learn the time series to predict the person's behavior, we propose to handle the prediction model in a recurrent way. Consequently, in the proposed model, the present input data takes into account the result yielded by the previous one. Recurrent models showed promising results in modeling complex data sets and extracting the important model from time series [15, 24]. This recurrent approach has been applied to mainly ensure the following benefits:

- Strong capacities in modeling temporal dependencies between outputs and inputs.
- An efficient prediction and identification of complex problems that require the evaluation of a huge amount of data.
- Best results of prediction for dynamic and control systems.
- Stimulation of the accuracy rate by involving previous data in the current data.

In this work, the ELM model is managed to cope with the recurrent use of data in the context of smart environments, we define the resulting model as "Recurrent Extreme Learning Machine" (RELM). The new model is mainly applied in order to describe the future activities of elderly evolving in their own homes. The main crucial features that we target with RELM are the accuracy of predictions and the rapidity of the model during the learning phase. Our experimentations are based on real data representing the human activities and the actions performed by persons. We focus on persons belonging to the first level of dependency as evaluated using the Functional Autonomy Measurement System (SMAF) scale adopted in the geriatric domain. Specifically, the data is related to persons with the ability to perform the activities of daily living (ADL). This category of persons is denoted as the first profile (P_1) using the SMAF terminology [22].

This paper is organized as follows. Section 2 presents a brief survey of related works regarding the prediction models and discusses the monitoring of elderly with the used dataset. In Section 3, a basic extreme learning machine is defined. Section 4 is devoted to discussing the process of our proposed behavior prediction model for elderly. In Section 5, we experiment our model to illustrate its performance during the learning and testing phases. We also compare RELM prediction against the baseline counterpart. Finally, Section 6 concludes the paper.

2 Prediction and monitoring of the human behavior

2.1 Prediction models

A great effort has been devoted to identifying the techniques that can effectively model the human behavior in smart environments. In the literature, many techniques based on machine learning and pervasive computing were used to achieve a better prediction model concerning the human activities and scenarios. Different combinations of Artificial Neural Networks (NNs) were used to learn the occupant's activities in smart homes. For example, in Zheng et al. [27], the authors use a self-adaptive neural network based on Growing-Self Organizing Maps (GSOM) as a learning algorithm. In [27], the work presented a GSOM-based data mining approach to cluster and analyze the human activities in a smart home environment. Hussein et al. [29] use two types of Neural Networks, namely the Feed-Forward Neural Networks, and the Recurrent Neural Networks, in order to help disabled people in their everyday life. The work confirmed the importance of the NN in the prediction of activities in a smart home. Moreover, the NN experimental results showed that the virtual data are close to the real data. Fang et al. [32] use a Back Propagation Neural Network (BPNN) with a feed-forward strategy during the learning process to recognize the activities of daily living in a smart home. They reveal that if the hidden neurons number increases, the accuracy of the model increases. However, the approach required a large learning time. The results reveal that using a bigger number of neurons in the hidden layer is more accurate [32]. However, it may result in a large learning time. The authors deduce that the BPNN achieves a good recognition accuracy. Liu et al. [28] develop a complex assistive system to predict the behavior of the person regarding the human activity and the health monitoring. The authors propose an activity recognition system concerning the arm and body posture such as Standing, Sitting, Walking, Falling-down. The results show a good recognition rate with the use of the NN. Teich et al. [30] build a Stable Neural Network (SNN) based on a feed-forward neural network in the learning step. The results show that a supervised feed-forward neural network with multiple hidden layers provides a good result after a short training period. In the context of human activity recognition from image sequences, Uddin et al. [31] proposed a new method based on the Independent Component Analysis (ICA) to extract the image sequences concerning the activity shape information. They used the HMM for the recognition of human activities. In [31], they show that a superior recognition is achieved using ICA based on HMM compared to the Principal Component Analysis (PCA) based on HMM. Concerning the healthcare applications, the work of [34] investigated the problem of recognizing the activities in a smart home. The work focused on handling the domination of major activities over minor activities. They proposed a new Evolutionary Ensemble Model (EEM) to process both minor and major activities independently. The EEM model is based on a Genetic Algorithm (GA) to handle the non-deterministic and complex nature of activities. The results of [34] show that the EEM model ensures better performances compared with HMM and the approach of Jehad et al. [40] in terms of precision, recall, F-measure, and accuracy. The authors of [26] examined the problem of physical monitoring for youth due to the downward trend in physical activities and the high prevalence of obesity. Hence, for the activity recognition and intensity estimation, the authors combine a time frequency feature extraction and a local distance metric learning (K-Nearest Neighbors) in order to design a new model called DML-KNN. This model allows the classification of a wide range of common activities. The results showed that DML-KNN is better than the support vector machine for the classification of tasks. In [36], we proposed an improved ELMAN-NN for daily living activity recognition concerning elderly in a smart home. The use of the Differential Evolution Algorithm (DE) in the learning step of ELMAN-NN was proposed in order to improve the accuracy of the prediction compared to the conventional algorithm. The results show that ELMAN-NN based on DE (Elman-NN-DE) ensure a better performance than ELMAN based on Genetic algorithms and the classic ELMAN.

2.2 Monitoring of elderly behavior

A smart home represents an environment which is equipped with a set of several sensors widespread in the house and used to collect, monitor, and understand the inhabitant's activities of daily living. Sensors have different "intelligence" levels. At the lowest level, the collected information is transformed into electrical signals, digitalized from the analog signals, and sent to microcontrollers for processing. The sensitivity of elder health status requires a deep focus on a long-term care and the adoption of a predictive approach to continuously evaluate the disability and the state of autonomy. Therefore, it is necessary to monitor the evolution of the person's autonomy and satisfy the following requirements:

- The autonomy degree of persons must be measured accurately using a well-established model (e.g. AGGIR and SMAF) that generates a score with test fidelity;
- Data must be collected and stored in a database that allows successive evaluations. These evaluations should be linked to the person in order to produce a long tracking.

Figure 1 shows the evolution of the dependency levels (as measured by SMAF) per age of the subjects living at home [35]. From this figure, we observe that the loss of autonomy aggravates over the time (age) but sometimes it risks to decrease abruptly. In this work, we focus on persons having difficulties with domestic tasks (e.g. cell use, meal preparation, shopping, house keeping, medication use, and laundry) [33] and ranked as the most autonomous profiles using the SMAF scale. Consequently, we can follow the persons from a state considered more autonomous to a less autonomous state. In our work, the real data are collected from the *e-health monitoring open data* project [22]. It provides a set of activities and actions achieved by the occupant in a chronological order. Data represent the activities as defined in the geriatric SMAF model. The provided data describes several scenarios throughout a year. The format of each event includes the beginning and the end time of each activity or basic action. It is worth noting that the SMAF model is a widely used clinical rating scale that measures the functional autonomy of persons especially elderly and dependent persons. In geriatrics, SMAF is used in order to rehabilitate individuals by providing them appropriate care and services and assessing needs to alleviate their disabilities. SMAF includes fourteen profiles of dependency

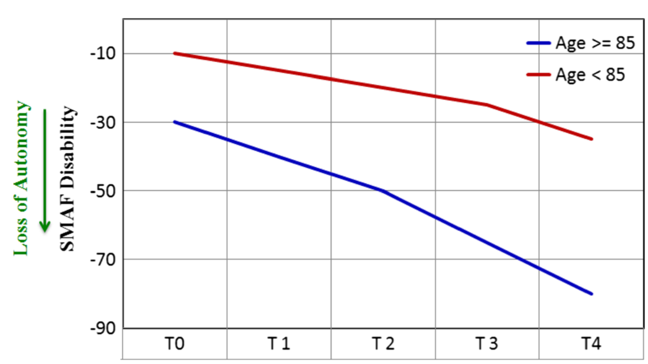

Fig. 1 Evolution of the SMAF scores by age for subjects living at home. T_0 to T_4 corresponds to the annual measurement times [35]

patterns called *iso-SMAF* profiles. Each profile is associated with a specific amount of nursing, support services, needs of supervision and costs of services based on the disabilities of the patient groups. In this paper, we focus on the first profile of SMAF which represents the persons that are autonomous. Table 1 summarizes the activities of such profile used in this work with a specific code for each activity.

We process several daily scenarios to build a behavior model of the elderly which is ultimately used to predict future activities. The achieved activities and actions by the inhabitant are represented with the following temporal relationship form: [day, start-time, end-time, activities-code]. Table 2 illustrates an example of data collected from a home. Collected data are processed and transformed into a continuous time series format.

Table 2 An example of collected data from a smart home

Day	Start-Time	End-Time	Activities-code
01	08:03:32	08:22:40	3
01	08:23:46	08:26:53	5
01	08:28:50	08:38:39	2
01	08:40:37	08:50:24	9
01	08:52:12	08:55:38	4
01	08:57:36	09:05:53	13
01	09:07:38	09:12:52	4
01	09:13:57	09:21:10	15
01	09:23:08	09:43:11	1
01	09:46:00	09:55:21	14
01	09:56:45	10:03:03	9

3 Extreme learning machine

ELM is a learning machine algorithm proposed by Huang et al. in order to improve the single-hidden layer feed-forward neural network (SLFN) [5]. The ELM model is identified as an emerging learning technique which provides a globally optimal solution with fast learning and good generalization performance. The model is easy to use and only needs to set the number of hidden layer neurons. In the learning step, ELM randomly selects the input weights and hidden biases and analytically determines the output weights without further adjustment. It determines the hidden output matrices through a simple generalized inverse operation using the Moore-Penrose inverse [2]. In ELM, the used structure is a typical SLFN structure which consists of an input layer, a hidden layer, and an output layer. Each neuron is linked using a weighted connection called *weight* (w). Other parameters are also used, mainly the *bias* (b) providing additional adjustable parameters of the model and the *transfer function* (f) which calculates the *output* (y). The transfer function f can be logarithmic, linear, Tang hyperbolic, radial basis, or sigmoid functions. We can, therefore, describe the network using the triplet (w, b, f).

Table 1 The activities/actions of daily living

Activity $\{action_1$ **(code1), . . . ,** $action_N$ **(code N)**$\}$
Eating {Eating (1)}
Dressing {Wear take off shoes (2)}
Washing {Take shower (3)}
Grooming {Washing hand/face (4), Hair dry (5), Move dish (34), Make up (6)}
Unary function {It take same toileting action (7)}
Bowel function{It take same toileting action (8)}
Toileting {It take same toileting action (9)}
Housekeeping {Vacuuming (10)}
Laundry {Wash machine (11)}
Meal preparation {Wash dish (11), Make coffee (13), Make tee (14), Make sandwich (15), Make hot food (16)}
Telephone {Cell use (17)}
Medication use {Take Medication (18)}
Watching TV {TV (19)}
Sleeping {Sleep (24)}
Go out {Go out (33)}
Reading {Reading (35)}

Figure 2 presents an overview of the ELM structure. For N distinct input samples x_i, where $x_i = \left[x_{i1}, x_{i2}, \ldots, x_{in}\right]^T \in \mathbb{R}^n$ is the n-dimensional input, the m-dimensional output of the ELM is $y_i = \left[y_{i1}, y_{i2}, \ldots, y_{im}\right]^T \in \mathbb{R}^m$. y_i is given by:

$$y_i = \sum_{j=1}^{k} \beta_j f(w_j.x_i + b_j), \quad i = 1, \ldots, N \tag{1}$$

Where $w_j = \left[w_{j1}, w_{j2}, \ldots, w_{jn}\right]$ is the weight vector connecting the input neurons to the j^{th} hidden neuron, $\beta_j = \left[\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}\right]^T$ is the weight vector connecting the output neuron to the j^{th} hidden neuron, k is the number of hidden layer neurons, b_j is the bias of the j^{th} hidden neuron, $w_j.x_i$ indicates the inner product of w_j and x_i, and f is a sigmoid function. Equation (1) can be written in a matrix format as:

$$H\beta = y \tag{2}$$

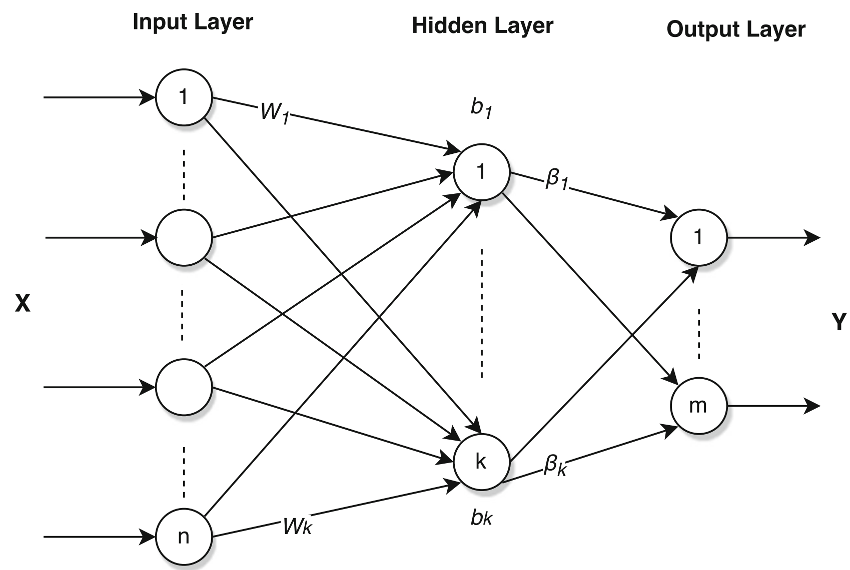

Fig. 2 Architecture of Extreme Learning Machine

Where H is the hidden layer output matrix of the neural network:

$$H = \begin{bmatrix} f(w_1.x_1 + b_1) & \ldots & f(w_k.x_1 + b_k) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ f(w_1.x_N + b_1) & \ldots & f(w_k.x_N + b_K) \end{bmatrix}_{N \times k} \tag{3}$$

$$\text{with } \beta = \begin{pmatrix} \beta_1^T \\ \vdots \\ \vdots \\ \beta_k^T \end{pmatrix}_{k \times m} \quad \text{and} \quad \mathbf{y} = \begin{pmatrix} y_1^T \\ \vdots \\ \vdots \\ y_k^T \end{pmatrix}_{N \times m}$$

When the input weight and hidden layer bias are determined randomly according to the output matrix H, the learning of SFLN is simply equivalent to finding the least square solution, which can be mathematically modeled as:

$$\beta = H^+ y \tag{4}$$

Where H^+ represents the Moore-Penrose generalized inverse of the H with

$$H^+ = (H^T H)^{-1}.H^T \tag{5}$$

Generally, β, which contains the optimal solution of the SFLN, can yield the optimal generalization capability of the output connection weights and the network. β is unique, which avoids producing the local optimal solution. Depending on β, the training error can be improved. Algorithm 1 represents a pseudocode of the ELM.

Algorithm 1 Pseudo code of ELM

1. Begin
2. Randomly choose the values of the **weight** w_i and the **bias** b_i, $i = 1, \ldots, k$;
k is the number of hidden layer neurons
3. Calculate the **output matrix** of the hidden layer;
4. Calculate the **output weight** using (4);
5. End

4 Elderly behavior prediction using the recurrent model

In order to provide a behavior prediction mechanism for the monitoring of elderly and dependent persons, we propose to define a recurrent approach and apply it to the conventional ELM model. The idea is to guarantee an efficient processing of large datasets coming from the multiple data sources and sensors of the smart environment. Another important feature is to ensure a fast prediction process which is of paramount importance in the daily monitoring. Such process is very convenient in e-health, for instance by anticipating risks and emergency situations. The recurrent approach is used to improve the convergence speed and the accuracy of the ELM model. The most significant aspect of our behavior prediction process is to provide the same (or approximate) input/output mapping relationship between the environmental parameters data and the user behavior habits. Parameters and behavior are considered as the input and output of our RELM model. In order to reach the previous requirements, a continuous training is applied on the collected data. Data is obtained from the sensors of the smart environment and, possibly, from other historical sources such as available clinical data of the monitored person. Once the training is completed, the proposed model is used to predict the future behavior of the elderly. Figure 3 presents the approach of our proposed model. The main steps of the elderly behavior model are: pre-processing of the data, selection of learning set, and finally the implementation of RELM.

4.1 Pre-processing

The *pre-processing* step is devoted to the processing of raw data collected from different sources existing in the smart environment. This step must be carried out in order to provide a better data presentation. The pre-processing consists of three main tasks: data sensors, time series, and normalization.

Data sensors (Fig. 4.A) from the original measures collected by the several behavior sensors, it is difficult to understand (i.e. identify and extract) the overall scenario and behavior of the elderly. For this reason, we convert the real data into time series in order to gain a clear observation captured at regular intervals. These intervals represent the time slots used in order to predict the behavior of the person.

Time series (Fig. 4.B) a time series is a succession of events (a_1, a_2, ..., a_n) associated to a time sequence (t_1, t_2, ..., t_n). To convert the sensor measurements, we focus on the determination of activities durations. First, the data is converted based on the start and end time into duration values using the simple (6) where ST and ET are the start time, respectively the end time in seconds, of a given activity or action.

$$Duration = ET - ST \tag{6}$$

Algorithm 2 is proposed to extract the time series from the data of activities of daily living.

Algorithm 2 Pseudo code of time series extraction

```
% act = current activity/basic action;
% {code act_1, ..., code act_n} = list of predefined codes of activities/actions;
Begin
var k=1;
    For i=1:length(MXData(:,1)) %process the i^th line of MXData table
                % process the 3^rd column of MXData i.e. "code of activity":
        act = MXData(i, 3)
        Switch(act)
            Case{code act_1}
                Duration = MXData(i, 2) − MXData(i, 1);
                For j=1:Duration
                    DurTab(k) = act;
                    k = k + 1;
                End For
            Case{code act_2}
                ⋮
        End
    End For
End
% MXData: table of data [start time, end time, code of activity]
% DurTab: table of successive sequences (traces) for each activity
```

Data normalization (Fig. 4.C) before the training of the dataset, we normalize the input data in order to erase the redundancy. Hence, the input data range becomes the interval [0, 1], which is ready to be injected as the inputs for the predictive process. The value of the input data is normalized as follows.

$$\overline{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{7}$$

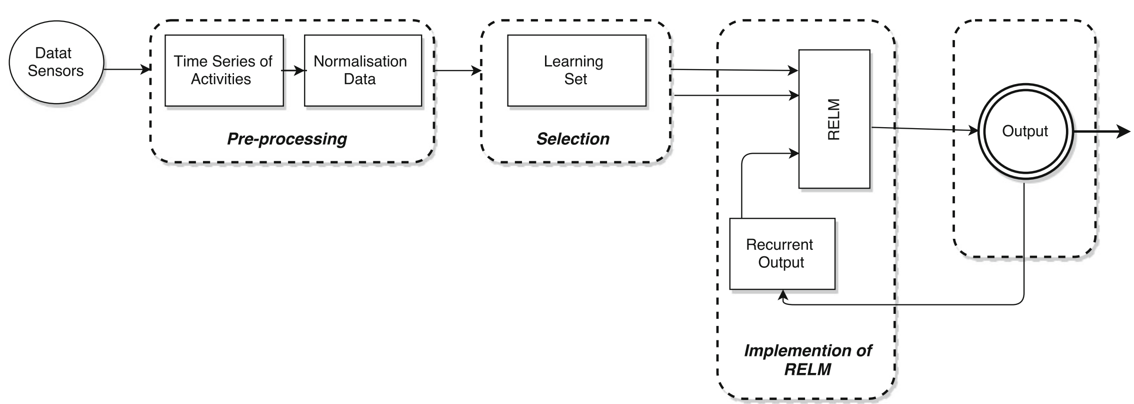

Fig. 3 Flowchart of elderly behavior prediction recurrent model

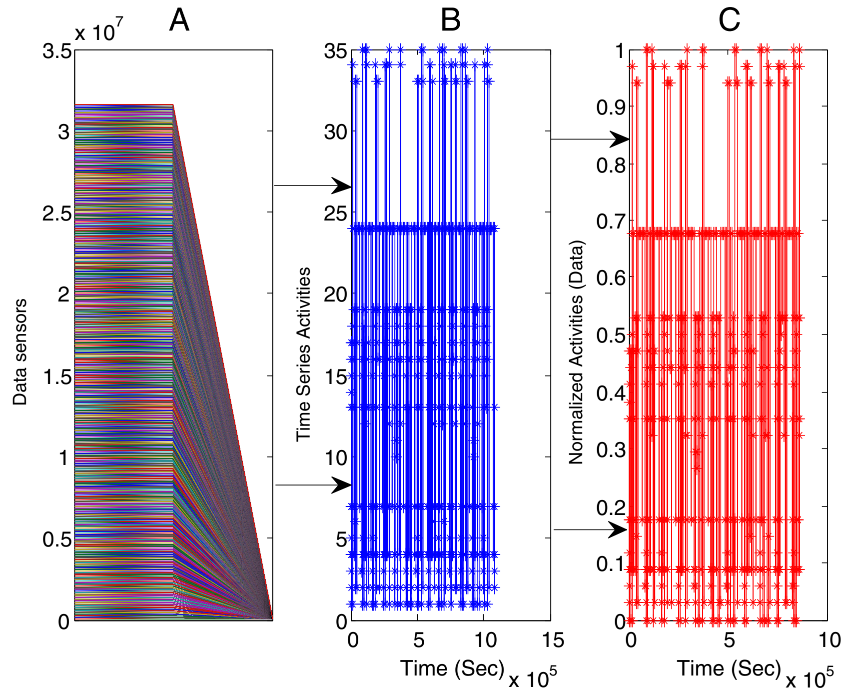

Fig. 4 Pre-Processing Step

Where x is the current value; $max(x)$ is the largest value of the input set, and $min(x)$ is the smallest value of the input set.

4.2 Selection of sets for learning and test

In order to perform an efficient prediction, a sufficient number of representative samples is required in the learning phase. Hence, it is important to select the learning and the test sets in the aim to satisfy the requirements of the RELM. For this purpose, it is crucial to split the input real data into two parts: a *learning set*, which will be used to build and train our model, and a *test set*, which is used to experiment our model on new data that was not used in the training step. Typically, a testing set is used to evaluate how well the model behaves with data outside the scope of the training set. We split the input data of the activities of daily living into three weeks for the learning step and nine days for the test step.

4.3 RELM description

As presented previously, at the outset of this paper, recurrent techniques have recently shown promising results in a variety of applications, especially when the data presents some sequential dependencies [23]. It is worth noting that the daily living activities in a smart space consist of scenarios that describe a succession of activities where each new activity depends on the previous activity. Hence, the best way to consider theses relationships is to adopt a dynamic model. In RELM, we define a feedback connection from the output neuron to the input layer. This way allows the integration of the output in the behavior prediction. RELM guarantees a dynamic characteristic by providing a temporal relationship between observations at different times. From another perspective, RELM holds a memory, which captures information about what has been calculated so far. Consequently, thanks to the incorporated feedback connections, our model becomes able to address the temporal relationships of the input data. Figure 5 shows the architecture of RELM.

4.4 RELM core implementation

In the core implementation of RELM, we aim to approximate the following function:

$$y(t+1) = f(x(t),\ x(t-1),\ \ldots,\ x(t-N_x),\ y(t),\ y(t-1),\ \ldots,\ y(t-N_y)) \tag{8}$$

Where $y(t+1)$ denotes the predicted output expected at time $t+1$, $x(t)$ is the input of the model at time t, f is the activation function, and N_x, N_y are the input and the output memory orders respectively.

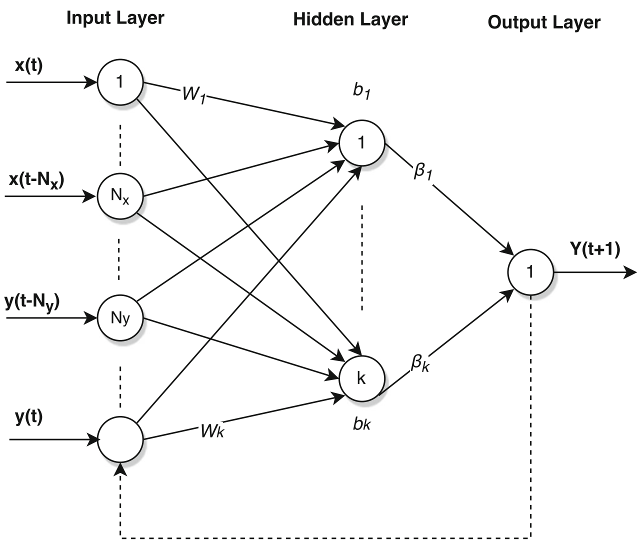

Fig. 5 The architecture of the Recurrent Extreme Learning Machine (RELM)

By considering the (1), the predicted output $y(t+1)$ can be written as:

$$y(t+1) = \sum_{j=1}^{k} \beta_j f(w_j \sum_{i=1}^{N_x} x(i) + b_j) + \sum_{j=1}^{k} \beta_j f(w_j \sum_{m=1}^{N_y} y(m) + b_j) \tag{9}$$

Where,

$X = [x(t), x(t-1), \ldots, x(t-N_x)]$, $Y = [y(t), y(t-1), \ldots, y(t-N_y)]$,

$w_j = [w_{j1}, w_{j2}, \ldots, w_{j(N_x+N_y)}]^T$ is the weighting vector connecting the input neurons to the j^{th} hidden neuron, $\beta = [\beta_1, \ldots, \beta_k]^T$ is the weighting vector connecting the output neurons to the j^{th} hidden neuron, and b_j is the bias.

It is worth repeating that the input weighting and the hidden layer bias are determined randomly, hence, based on the output matrix $H(t)$, the learning of RELM is equivalent to the identification of the least square solution of the following Equation.

$$\beta = (H(t)^{+}) \,.\, y(t+1) \tag{10}$$

Where $H(t)^{+}$ is the general inverse of the matrix $H(t)$. Equation (10) can be simplified as follows.

$$H(t) \,.\, \beta = y(t+1) \tag{11}$$

$$\text{With, } H(t) = \begin{bmatrix} \left.\begin{matrix}\text{feedforward}\\ \text{matrix}\end{matrix}\right\rangle_{k \times N_x} & \\ & {}_{k \times N_y}\left\langle\begin{matrix}\text{backforward}\\ \text{matrix}\end{matrix}\right. \end{bmatrix}, \textit{more specifically}$$

$$H(t) = \begin{bmatrix} \left\{\begin{matrix} \overset{Forward}{\longrightarrow} & \longrightarrow \longrightarrow & \overset{Forward}{\longrightarrow} \\ f(w_1.x(t)+b_1) & \ldots \;\ldots & f(w_k.x(t)+b_k) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ f(w_1.x(t-N_x)+b_1) & \ldots \;\ldots & f(w_k.x(t-N_x)+b_k) \end{matrix}\right\} \\ \left\{\begin{matrix} \overset{Backward}{\longleftarrow} & \longleftarrow \longleftarrow & \overset{Backward}{\longleftarrow} \\ f(w_1.y(t)+b_1) & \ldots \;\ldots & f(w_k.y(t)+b_k) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ f(w_1.y(t-N_y)+b_1) & \ldots \;\ldots & f(w_k.x(t-N_y)+b_k) \end{matrix}\right\} \end{bmatrix}_{k \times (N_x+N_y)} \tag{12}$$

Generally, β, which contains the optimal solution of the RELM, can yield the optimal generalization capability of the output connection weights and the network. β is unique, which avoids producing the local optimal solution. Depending on β, the training error can be improved. As we can notice, in our recurrent model, the hidden layer output matrix is composed of two sub-matrices: the feedforward matrix and the backforward matrix. The first matrix is the same as defined in the basic version of ELM (i.e. the feedforward matrix (4). The second matrix is the backforward matrix contains the outputs at times $(t - N_y)$ to t. Algorithm 3 represents the pseudocode of the ELM.

Algorithm 3 Pseudo code of RELM

1. **Begin**
2. **For** each **sequence** in the training set of sequences
3. Randomly choose the values of the **weight** w_i and the **bias** b_i of the $(k \times N_x)$ **forward matrix**; % where $i \in \{1, \ldots, k\}$ is the index of the hidden layer neuron
4. Randomly choose the values of the **weight** w_i and the **bias** b_i of the $(k \times N_y)$ **backward matrix**; % where $i \in \{1, \ldots, k\}$ is the index of the hidden layer neuron
5. Calculate the **output matrix** $H(t)$ of the hidden layer given by (12);
6. Calculate the **output weight** β according to (10);
7. **End for**
8. **End**

4.5 Evaluation settings

In order to substantially strengthen the effectiveness of RELM, it is important to determine a suitable setting and architecture of our model. It should be underlined that the tuning of RELM is simplified since there are only three introduced parameters, namely the number of hidden neurons, the number of recurrent output neurons and the activation function. The hidden neuron number (HNN) and the recurrent neuron number (RNN) have an important impact on the performance of RELM in terms of learning time and prediction error. To build an optimal model, a reasonable setting should be identified. Thus, many tests have to be performed in order to select the optimal settings.

To reach this objective, we first analyze the relationship between HNN and RNN in term of error (Fig. 6).

Then, we study how the HNN affects the learning time (Fig. 7). The number of parameters used in our evaluation is $HNN = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 50, 100]$ and $RNN = [1, 2, 3, 4, 5]$. Figure 6 shows that when HNN is between 1 and 5 and RNN is between 1 and 3, the RMSE is not sufficient hence the RELM is not ready for the best prediction. When $HNN > 5$ and RNN is between 1 et 3, we observe that the RMSE starts to be stabilized but not enough if compared to $RNN \geq 4$. Indeed, RELM reaches the maximal accuracy when $HNN \geq 5$ and $RNN \geq 4$ with $\mathrm{RMSE} = 1.5625 \times 10^{-6}$. Furthermore, when $HNN > 5$ and $RNN > 4$, the RMSE is almost unchanged. Figure 7 shows the impact of HNN on the learning time of RELM. We can observe that the learning time gradually increases when the number of hidden neurons increases. Thus, we can conclude

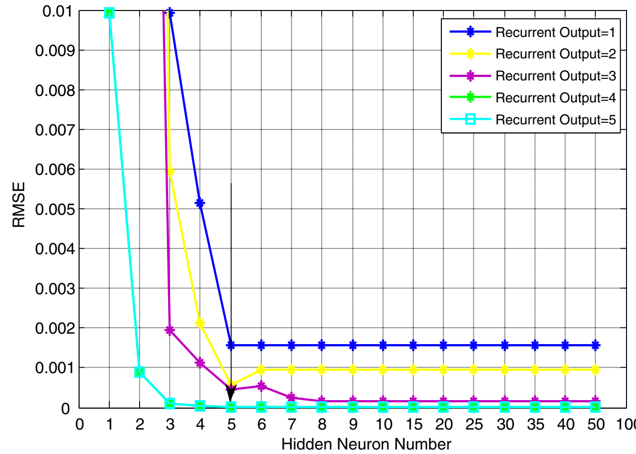

Fig. 6 Evaluation of RELM in terms of RMSE based on the hidden neuron (HNN) and recurrent neuron number (RNN)

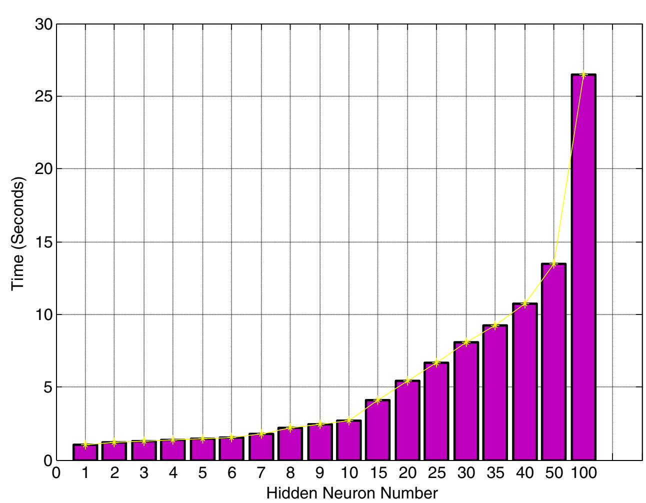

Fig. 7 Evaluation of RELM in terms learning time-based on the HNN number

an important number of hidden nodes require more learning time.

In addition, the activation function plays a very important role in the behavior of the neurons. The sigmoid function is the most used in time series prediction problems [37–39]. For this reason, we use a sigmoid function in order to perform a global and continuous approximation about the integrality of the input space. Our deep experimentations during the RELM tuning leads us to determine the best behavior of RELM by setting HNN to 5 in the hidden layer and fixing RNN to 4 in the input layer using the sigmoid activation function.

Table 3 The evaluation of the RELM prediction during the testing step

	RMSE	Similarity	PE(%)	PS(%)
$Period_1$	2.2776×10^{-4}	0.9982	9.7757	90.2243
$Period_2$	2.2818×10^{-4}	0.9981	9.0219	90.9781
$Period_3$	2.2630×10^{-4}	0.9978	8.8479	91.1521
$Period_4$	2.2911×10^{-4}	0.9981	9.3887	90.6113
$Period_5$	2.3236×10^{-4}	0.9984	10.0653	89.9347
$Period_6$	2.3230×10^{-4}	0.9983	10.0648	89.9352
$Period_7$	2.2146×10^{-4}	0.9979	8.1065	91.8935
$Period_8$	2.2860×10^{-4}	0.9982	8.9060	91.0940
$Period_9$	2.3457×10^{-4}	0.9993	12.0979	87.9021

5 Experimentation results

We conducted performance evaluations of the proposed model RELM by analyzing its results during the two main phases namely the learning and testing steps. The second step of our evaluation considers existing prediction models in order to provide a baseline for the comparison of RELM with existing models.

5.1 Performances of RELM

In the learning step, we estimate the parameters of the model and evaluate the prediction error during the first three weeks of the input dataset. We first focus on the evaluation of the obtained results in the learning step then; the predicted values are compared against the real values. In Fig. 8, we show a sample of six days prediction results of the elderly activities scenario compared to the real data in the learning step.

For each day, the blue dashed line represents the real data and the red dashed line represents the predicted data. As we can observe from Fig. 8, the predicted values fit perfectly with the real measurements provided by sensors. The results of the training step promise good predictions in the testing step using the trained parameters.

The next step of our evaluation is the experimentation of the RELM prediction after the training phase. Indeed, in the testing step, we apply the model built upon the training

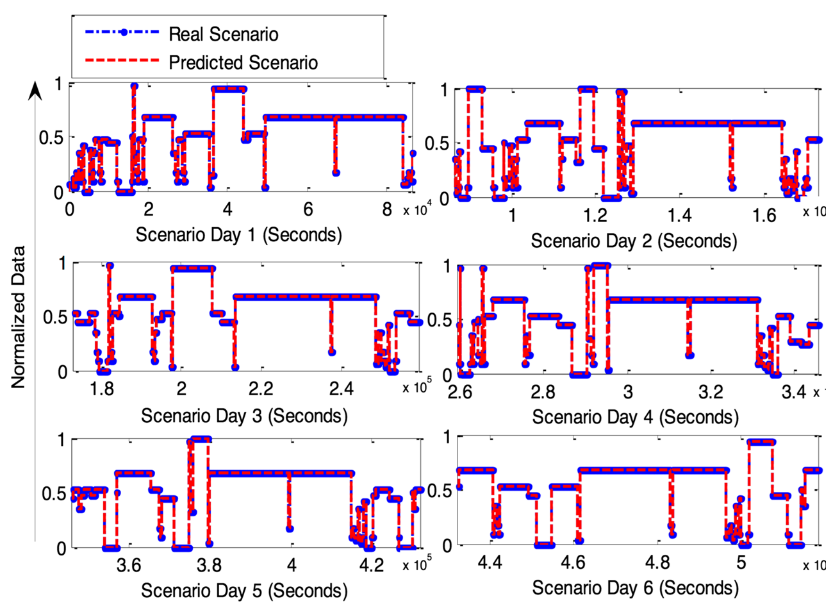

Fig. 8 Comparison of real and predicted scenarios in the learning step during 6 days

Table 4 The evaluation of the RELM in terms of learning time

Simulation Number	1	2	3	4	5	6	7	8	9	10	**Average**
learning Time (Sec)	1.389	1.597	1.493	1.543	1.587	2.645	1.816	1.494	1.463	1.439	**1.492**

parameters and data in order to predict future scenarios of the elderly.To do so, we use the test data and evaluate the RELM prediction by selecting the three following factors: the root means squared error (RMSE), the similarity, and the percentage of the error (PE). These metrics are given by:

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - \widehat{y_i})^2} \tag{12}$$

$$Similarity = \frac{y - \widehat{y}}{\|y\| \times \|\widehat{y}\|} \tag{13}$$

$$PE = 100.\frac{|y - \widehat{y}|}{y} \tag{14}$$

Where, y is the observed data, N is the number of the observed data, and $\widehat{y}$ is the predicted data. To evaluate the prediction, RELM is carried out during nine different periods in each one we evaluate the similarity, the RMSE, the percentage of error (PE), and the percentage of similarity (PS). Each time slot lasts one day. Table 3 shows the obtained results.

The results presented in Table 3 show that RELM has good performances regarding the prediction of scenarios. This observation is confirmed by the results of RMSE, similarity, the percentage of error (PE) and similarity (PS) in different periods during the test step. Indeed, we observe that the RMSE in the testing step is bounded by 2.2146×10^{-4} and 2.3457×10^{-4} with a high similarity varying from 0.9978 to 0.9993. The observed error rate is low and varies from 8.11% to 12.10% while the observed percentage of similarity is high with at least 87.90% and can reach 91.90%.

For instance, concerning $period_3$, the observed metrics are PS = 91.1521%, RMSE = 2.2630×10^{-4}, similarity = 0.9978, and a PE of 8.8479% which means that the elderly activities and actions were well anticipated over the time. In the e-health and smart environments realm, in addition to the accuracy of predictions, another significant parameter is the speed of prediction. This factor is of paramount importance for the anticipation of risks and emergency situations especially when the answer of caregivers should not be delayed. In addition to the best results provided by RELM concerning the accuracy of predictions, we evaluate RELM regarding the computation time.

After the evaluation of the prediction's accuracy of RELM, we focus on the evaluation of the learning and testing time. For this purpose, we evaluate the learning time of RELM in several times (using 10 simulations) in order to identify average values. The results of Table 4 show a reasonable average time of learning which is 1.492 seconds. This result confirms the rapidity of our model in the learning step. Table 5 shows the results of a similar experimentation applied for the testing step. The testing time of RELM was evaluated during 10 simulations. As a result, the observed average is 0.0493 seconds, which, in turn, confirms the rapidity of our model in the testing step.

5.2 Comparative study

To confirm the supremacy of our model, we evaluate RELM against other existing prediction models. In order to produce a representative comparison between the models, we select the following metrics: RMSE, the learning time and the testing time. We focus our comparison of RELM on the following models: basic ELM, Single-Hidden Layer Feed Forward Neural Network based on Differential Evolution algorithm (SLFN-DE), Elman Neural Networks based on Differential Evolution (Elman-NN-DE), Self-adaptive extreme learning machine (Sa-ELM), and Hybrid ELM and GA model (ELM-GA). Table 6 summarizes the observed

Table 5 The evaluation of the RELM in terms of testing time

Simulation Number	1	2	3	4	5	6	7	8	9	10	**Average**
Testing Time (Sec)	0.050	0.044	0.044	0.043	0.039	0.065	0.061	0.044	0.059	0.044	**0.0493**

Table 6 RMSE evaluation for RELM, classic ELM, SLFN-DE, Elman-NN-DE, Sa-ELM, and ELM-GA during the learning and testing phases

Models	RMSE	
	Learning	Testing
Proposed model RELM	$\mathbf{1.5625}\times10^{-6}$	$\mathbf{2.2818}\times10^{-4}$
Classic ELM	0.3872	0.073
Sa-ELM	1.0147×10^{-3}	9.0701×10^{-2}
ELM-GA	1.6263×10^{-3}	1.5909×10^{-1}
SLFN-DE	8.9167×10^{-5}	0.0518
Elman-NN-DE	6.1813×10^{-4}	0.0852

RMSE values during the learning and the testing steps for all the studied algorithms. We can observe that RELM succeeds in providing the best performance related to the accuracy of predictions with the smallest RMSE values compared to existing models.

In Table 7, we show the results of our evaluation related to the learning and the testing time for all the selected prediction models. The results reveal that even the two times obtained by RELM (in learning and testing phases) are bigger than those obtained with the classical ELM, the difference is not significant if compared to the other models. Indeed, both of the RELM times are much smaller than those obtained with SLFN-DE, Elman-NN-DE, Sa-ELM and ELM-GA especially concerning the learning step where the observed difference is very significant.

By observing the results of Tables 6 and 7, we can conclude that the prediction accuracy of RELM based on RMSE is much lower than ELM, SLFN-DE, Elman-NN, Sa-ELM, and ELM-GA. Furthermore, RELM spends less learning time compared to other prediction models.

Table 7 Time evaluation for RELM, classic ELM, SLFN-DE, Elman-NN-DE, Sa-ELM, and ELM-GA during the learning and testing phases

Models	Time (seconds)	
	Learning	Testing
RELM	**3.1996**	**0.060300**
Classic ELM	0.0583821	0.044905
Sa-ELM	308.816570	0.085641
ELM-GA	83.179903	0.065465
SLFN-DE	3469.728034	0.394624
Elman-NN-DE	578.255760	0.895468

6 Conclusion

In the context of smart homes dedicated to the monitoring of fragile populations such as elderly people and dependent persons, one of the main interests is to model the behavior of the inhabitant in his own environment. The behavior modeling enables the design of new models and smart systems that continuously provide context-aware services adapted to the person's profile and real needs. In this realm, one of the paramount objectives of such modeling is to be able to anticipate and predict risks, deteriorations, emergency situations, and provide required help and assistance without delay. In this paper, we have proposed an improved version of the ELM model which ensures an accurate and fast prediction of the elderly behavior. The proposed model, called RELM, is a recurrent form of the ELM and involves the inputs as well as the lagged outputs for the prediction operations.

The performed experimentations of RELM show good performances regarding the accuracy with a root mean square error of 1.5625×10^{-6} during the learning step and 2.2818×10^{-4} during the testing step. Concerning the timeliness of the learning, RELM quickly succeeded to achieve this phase with a reasonable average time of 3.1996 seconds and a testing time of 0.06030 seconds which confirmed the efficiency of our model. We provided a comparative evaluation of the proposed model with some existing models in the literature: classic ELM, SLFN-DE, Elman-NN-DE, Sa-ELM, and ELM-GA. This evaluation confirmed the superiority of RELM in terms of prediction's accuracy and learning rapidity. Consequently, RELM model can ultimately be used to predict future values representing the expected activities and actions of the monitored person with a great confidence and a great learning rapidity.

Acknowledgements The authors would like to express their thanks to all the team of the project "e-Health Monitoring Open Data project".

References

1. Bazi Y, Alajlan N, Melgani F (2014) Differential evolution extreme learning machine for the classification of hyperspectral images. IEEE Geosci Remote Sens Lett 11(6):1066–1070
2. Barata J, Carlos A, Hussein M (2012) The Moore–Penrose Pseudoinverse: A tutorial review of the theory. Braz J Phys 42(1-2):146–165
3. Bin L, Yibin L, Xuewen R (2011) Comparison of echo state network and extreme learning machine on nonlinear prediction. J Comput Inf Syst 7(6):1863–1870
4. Rong Y, Zhang G, Chang Y, Huang Y (2016) Integrated optimization model of laser brazing by extreme learning machine and genetic algorithm. Int J Adv Manuf Technol 87(9-12):2943–2950

5. Huang G, Zhu Q, Siew C (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
6. Janakiraman VM, Nguyen X, Assanis D (2016) Stochastic gradient based extreme learning machines for stable online learning of advanced combustion engines. Neurocomputing 177:304–316
7. Qu Y, Shen Q, Parthaláin N (2010) Extreme learning machine for mammographic risk analysis. In: UK workshop computational intelligence (UKCI). IEEE, Piscataway, pp 1–5
8. Sridevi N, Subashini P (2013) Combining Zernike moments with regional features for classification of handwritten ancient tamil scripts using extreme learning machine. emerging trends in computing communication and nanotechnology (ICE-CCN). In: International conference on IEEE, pp 158–162
9. Wang D, Wang R, Yan H (2014) Fast prediction of protein–protein interaction sites based on extreme learning machines. Neurocomputing 128:258–266
10. Yage Z, Zhihua C, Wenyin G, Xinxin W (2015) Self-adaptive differential evolution extreme learning machine and its application in water quality evaluation. J Comput Inf Syst 11(4):1443–1451
11. Ortín S, Soriano MC, Pesquera L, Brunner D, San-Martín D, Fischer I, Gutiérrez JM (2015) A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron. Sci Report 5:14945
12. Goudarzi A, Banda P, Lakin M (2014) A Comparative Study of Reservoir Computing for Temporal Signal Processing. Tech. Rep. University of New Mexico
13. Wang GG, Lu M, Dong YQ, Zhao XJ (2016) Self-adaptive extreme learning machine. Neural Comput & Applic 27(2):291–303
14. Huang G, Huang GB, Song S, You K (2015) Trends in extreme learning machines: a review. Neural Netw 61:32–48
15. Mahmoud S, Lotfi A, Langensiepen C (2013) Behavioral pattern identification and prediction in intelligent environments. Appl Soft Comput 13(4):1813–1822
16. Undesa (2012) Population division united nations department of economic and social affairs
17. Skouby KE, Kivimäki A, Haukiputo L (2014) Smart cities and the ageing population. The 32nd Meeting of WWRF
18. Zaineb L, Tayeb L, Philippe R, Fréderic W, Hassani M (2016) A markovian-based approach for daily living activities recognition. In: International conference on sensor networks (SENSORNETS'16)
19. Zheng H, Wang H, Black N (2008) Human activity detection in smart home environment with self-adaptive neural networks. In: IEEE international conference on networking, sensing and control, 2008. ICNSC, vol 2008, pp 1505–1510
20. Oniga S, Suto J (2014) Human activity recognition using neural networks. In: 15th international carpathian control conference (ICCC). IEEE, Piscataway, pp 403–406
21. Oniga S, Suto J (2016) Activity recognition in adaptive assistive systems using artificial neural networks. Elektronika ir Elektrotechnika 22(1):68–72
22. Mshali HH, Lemlouma T, Magoni D (2015) Context-aware Adaptive Framework for e-Health Monitoring. In: IEEE international conference on data science and data intensive system, pp 276–283
23. Guo T, Xu Z, Yao X, Chen H, Aberer K, Funaya K (2016) Robust online time series prediction with recurrent neural networks. In: IEEE international conference on data science and advanced analytics, pp 816–825
24. Prasad SC, Prasad P (2014) Deep recurrent neural networks for time series prediction. arXiv:1407.5949
25. Çatak FÖ (2015) Classification with boosting of extreme learning machine over arbitrarily partitioned data. Soft Comput 21(9):2269–2281
26. Ren X, Ding W, Crouter SE, Mu Y, Xie R (2016) Activity recognition and intensity estimation in youth from accelerometer data aided by machine learning. Appl Intell 45(2):512–529
27. Zheng H, Wang H, Black N (2008) Human activity detection in smart home environment with self-adaptive neural networks. Networking, Sensing and Control 9:1505–1510
28. Liu Z, Song Y, Shang Y, Wang J (2015) Posture recognition algorithm for the elderly based on BP neural networks. In: Control and decision conference, pp 1446–1449
29. Hussein A, Adda M, Atieh M (2014) Smart home design for disabled people based on neural networks. Procedia Computer Science 37:117–126
30. Teich T, Roessler F, Kretz D et al (2014) Design of a prototype neural network for smart homes and energy efficiency. Procedia Engineering 69:603–608
31. Uddin MZ, Lee JJ, Kim TS (2010) Independent shape component-based human activity recognition via Hidden Markov Model. Appl Intell 33(2):193–206
32. Fang H, He L (2012) BP Neural network for human activity recognition in smart home. In: International conference on computer science and service system, pp 1034–1037
33. Graf C (2009) The lawton instrumental activities of daily living (IADL) scale. MEDSURG Nursing: Official journal of the Academy of Medical-Surgical Nurses 18(5):315–316
34. Fahim M, Fatima I, Lee S, Lee Y (2013) EEM: Evolutionary ensembles model for activity recognition in Smart Homes. Appl Intell 38(1):88–98
35. Hébert R, Raîche M, Dubois MF, Gueye NR, Tousignant M (2012) Development of indicators to promote measures for the prevention and rehabilitation of functional decline in older people. Revue d'epidemiologie et de sante publique 60(6):463–472
36. Liouane Z, Lemlouma T, Roose P, Weis F, Messaoud H (2016) An improved elman neural network for daily living activities recognition. In: International conference on intelligent systems design and applications. Springer, Cham, pp 697–707
37. Lotfi A, Langensiepen C, Mahmoud SM, Akhlaghinia MJ (2012) Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. J Ambient Intell Humaniz Comput 3(3):205–218
38. Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175
39. Patra A, Das S, Mishra SN, Senapati MR (2017) An adaptive local linear optimized radial basis functional neural network model for financial time series prediction. Neural Comput Applic 28(1):101–110
40. Jehad A, Lee Y, Lee S (2010) A smoothed naive Bayes-based classifier for activity recognition. IETE Tech Rev 27(2):107–119