

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and
Information Systems

School of Computing and Information Systems

12-2023

Spatial-temporal episodic memory modeling for ADLs: Encoding, retrieval, and prediction

Xinjing SONG

China University of Petroleum

Di WANG

Nanyang Technological University

Chai Quek

Nanyang Technological University

Ah-hwee TAN

Singapore Management University, ahtan@smu.edu.sg

Yanjiang Wang

China University of Petroleum

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#), and the [Health Information Technology Commons](#)

Citation

SONG, Xinjing; WANG, Di; Quek, Chai; TAN, Ah-hwee; and Wang, Yanjiang. Spatial-temporal episodic memory modeling for ADLs: Encoding, retrieval, and prediction. (2023). *Complex & Intelligent Systems*. 1-18.

Available at: https://ink.library.smu.edu.sg/sis_research/8472

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.



Spatial-temporal episodic memory modeling for ADLs: encoding, retrieval, and prediction

Xinjing Song¹ · Di Wang² · Chai Quek³ · Ah-Hwee Tan⁴ · Yanjiang Wang¹

Received: 29 March 2023 / Accepted: 15 November 2023
© The Author(s) 2023

Abstract

Activities of daily living (ADLs) relate to people's daily self-care activities, which reflect their living habits and lifestyle. A prior study presented a neural network model called STADLART for ADL routine learning. In this paper, we propose a cognitive model named Spatial-Temporal Episodic Memory for ADL (STEM-ADL), which extends STADLART to encode event sequences in the form of distributed episodic memory patterns. Specifically, STEM-ADL encodes each ADL and its associated contextual information as an event pattern and encodes all events in a day as an episode pattern. By explicitly encoding the temporal characteristics of events as activity gradient patterns, STEM-ADL can be suitably employed for activity prediction tasks. In addition, STEM-ADL can predict both the ADL type and starting time of the subsequent event in one shot. A series of experiments are carried out on two real-world ADL data sets: Orange4Home and OrdonezB, to estimate the efficacy of STEM-ADL. The experimental results indicate that STEM-ADL is remarkably robust in event retrieval using incomplete or noisy retrieval cues. Moreover, STEM-ADL outperforms STADLART and other state-of-the-art models in ADL retrieval and subsequent event prediction tasks. STEM-ADL thus offers a vast potential to be deployed in real-life healthcare applications for ADL monitoring and lifestyle recommendation.

Keywords Spatial-temporal episodic memory · Encoding and retrieval · ADL retrieval · Subsequent event prediction

Introduction

Activity of daily living (ADL), as a term often applied in the healthcare domain, involves a series of basic, ordinary, and repetitive activities that a person performs daily to meet the needs of daily living [1]. The study of ADLs is of significant value as they reflect one's physical status and self-care abilities. In particular, people with mental or physical disabilities and the elderly may have difficulty performing ADL tasks. As such, health assistive technologies leveraging the knowledge

of ADLs have significant application prospects for providing care to these people and reducing the burden on healthcare systems and caregivers.

With advances in artificial intelligence (AI), sensor technologies, and pervasive computing, smart homes have been widely adopted for health monitoring, particularly for the disabled and the elderly. Smart home-based activity prediction is of significant value in real-world applications, such as medication reminders, activity recommendations, security monitoring, etc. However, activity prediction remains a challenging task due to the complexity of the living environment and the diversity of individual's daily routines. Note that in the scope of this research work, we focus on the modeling of high-level ADL patterns (e.g., always taking a nap after lunch) through mining one's ADL routines or habits [2], rather than the modeling of low-level activity types (e.g., sitting on sofa) through analyzing the collected raw sensory inputs [3]. Specifically, different smart environments provide different types of location information and ADLs. In addition, the starting time and duration of the same ADL may vary daily, and the number and sequence of activities performed each day depend on the environment of the day or other ran-

✉ Yanjiang Wang
yjiangwang@upc.edu.cn

¹ College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266580, China

² Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore 639798, Singapore

³ School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore

⁴ School of Computing and Information Systems, Singapore Management University, Singapore 178902, Singapore

dom occurrences. All such uncertainties pose challenges to the smart home-based activity prediction.

To properly model high-level ADL patterns, inspired by the memory function of the human brain, we consider using an episodic memory neural network model to encode and retrieve ADL streams. Specifically, ADLs and the associated specific information are encoded into event patterns, and events in a day are encoded into episode patterns. Thus, episodic memory models are suitable for learning typical behavioral patterns from experience, allowing generalization across events, and remaining sufficiently adaptable to new events.

A prior study presents Spatial-temporal ADL Adaptive Resonance Theory (STADLART) [2] for daily routine learning. STADLART is a three-layer neural network, including the input layer, the category layer, and the routine layer. Attributes of the input layer include the ADL type, its spatial information, and its temporal information, which comprises the starting time, duration, and the associated day information. The spatial-temporal ADL pattern nodes in the category layer have the same activation value when they are activated. Thus, in the activation vector for routine learning, a value of 1 indicates that the pattern is activated while 0 indicates not activated. This setting implies that STADLART is not applicable for activity prediction tasks because the order of events is not encoded.

This paper proposes a neural network model named Spatial-Temporal Episodic Memory for ADL (STEM-ADL), which substantially extends STADLART to encode ADL sequences in the form of episodic memory. STEM-ADL is a three-layer hierarchical network model based on the dynamic characteristics of fusion ART [4, 5]. The three layers of STEM-ADL are (i) the input layer which contains event-specific knowledge, (ii) the event layer which encodes the multi-modal information into events, and (iii) the episode layer which encodes the related events in a day into an episode and models the order of events by time decays. In STEM-ADL, memory retrieval is first triggered in the input layer with the retrieval cues. Through the bottom-up memory search, STEM-ADL retrieves the related events and episodes sequentially. Then, it predicts the activity by retrieving the specific memory through the top-down readout process (see “[STEM-ADL architecture](#)” section for details).

STEM-ADL is able to retrieve specific ADLs from the user’s spatial-temporal preferences and predict subsequent events based on past experience. For model evaluation, we conduct experiments using two real-world ADL routine data sets, namely Orange4Home [6] and OrdonezB [7]. The results indicate that STEM-ADL has a high level of robustness in event retrieval using partial and noisy cues. Furthermore, STEM-ADL outperforms STADLART and other state-of-the-art (SOTA) methods in the ADL retrieval and subsequent event prediction tasks. To the best of our knowl-

edge, this is the first research on smart home-based ADL prediction using a brain-inspired computational episodic memory model.

The main contributions of this paper include:

1. STEM-ADL draws inspiration from human episodic memory, encodes the defined ADL and its associated multi-modal information as an event, and further encodes the sequence of events in a day into an episode.
2. STEM-ADL can retrieve the ADL type based on the given spatial and temporal information.
3. STEM-ADL models the temporal relationship of events and can predict both the ADL type and starting time of the subsequent event in one shot.
4. STEM-ADL is evaluated on two publicly available real-world data sets, achieves a high level of robustness in event retrieval using partial and noisy cues, and outperforms SOTA methods in ADL prediction tasks.

The rest of this paper is arranged as follows. “[Related work](#)” section reviews the literature of the relevant work. “[Preliminaries: fusion ART and STADLART](#)” section recalls some fundamentals of fusion ART and STADLART. “[STEM-ADL architecture](#)” section describes the proposed approach in detail. “[Experimental setup](#)” section deals with the experimental setup. “[Experimental results](#)” section discusses the experimental results. “[Conclusion](#)” section summarizes the conclusions and presents future work.

Related work

Among the cognitive functions of the human brain, memory is an advanced function related to learning and decision-making. In particular, episodic memory is a collection of events that a person has experienced in the past. The general idea of episodic memory modeling is to store events with the temporal sequence and apply statistical methods to deal with noisy and incomplete cues [8]. The integration of episodic memory was proposed in Soar cognitive architecture [9], using the working memory tree data structure to encode, store, and retrieve episodes. However, the system may be inefficient due to a large number of snapshot storage.

A series of self-organizing neural networks, named fusion ART, integrated different learning paradigms into a universal learning model [5]. The fuzzy choice and match functions and the complement coding technique [10] in fusion ART generalize the input patterns and suppress irrelevant attributes. Based on fusion ART, the memory model STEM [11] encoded the multi-modal input of the hall entrance monitoring video into events with relevant context information without an episode layer. Another fusion ART-based episodic memory model was used to assist a robot in observ-

ing and memorizing five kinds of contextual information, i.e., object, people, place, time, and activity [12]. Closely related to episodic memory, the fusion ART-based autobiographical memory model [13, 14] encoded the emotional state as one of the inputs to mimic the difference in human memory recall between happy and sad memories. STADLART [2] model discovers the execution patterns of ADLs throughout the day and clusters them into ADL daily routines. To the best of our knowledge, STEM-ADL is the first study to employ episodic memory for smart home-based activity prediction.

There has been an extensive literature on human activity recognition [15–17]. The activity recognition problem in smart home is to automatically recognize the activities of residents by utilizing the sensor data collected in the smart environment. In [18], an online daily habit modeling and anomaly detection (ODHMAD) model was proposed to recognize ADL, model habits, and detect abnormal behavior for the elderly living alone in real time. ODHMAD uses online activity recognition (OAR) to recognize activities by learning the activation status of sensors and employs the dynamic daily habit modeling (DDHM) component to model the elderly's daily habits, offering personalized knowledge to recognize abnormal behaviors. ODHMAD deals with raw sensor data collected from a simulation environment and lacks real-world data to evaluate the system. However, what we are exploring is the spatial-temporal relationship between the user's ADL, time and place in the intelligent environment, so as to predict the activity from the user behavior sequence. The activity prediction problem in the smart home is to automatically predict the future activities of the residents according to their past and present situation, including predicting the subsequent activities and the time of occurrence. For smart home-based activity prediction, there are two main approaches. One approach is sequence mining, which mines the relevant behavior patterns through a series of sequence analyses [19]. SPEED is a sequence prediction model that constructs decision trees based on ADL events recognized by chopping the data streams into sliding windows [20]. The other approach is to combine sequence matching with machine learning methods. CRAFFT [21] uses a dynamic Bayesian network with four input context information to predict activity. Similarly, PSINES [22] extended the dynamic bayesian network architecture. Deep learning methods (e.g., LSTM) have also been applied for smart home-based activity prediction [23, 24]. Sequential pattern mining was applied to find the time pattern of the activity sequence, and then conditional random field (CRF) was used to model the activity sequence for predictions [25]. Our STEM-ADL model belongs to the latter approach. Therefore, we select baseline models of the same approach for performance comparisons (see “Experimental results” section).

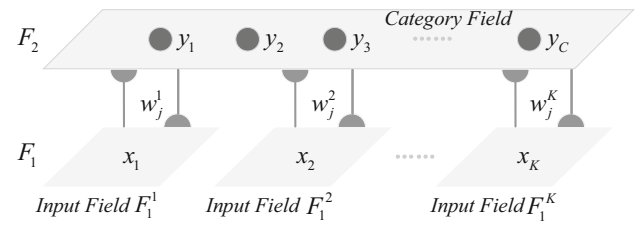


Fig. 1 Fusion ART network architecture. Each parallelogram in F_1 indicates an input channel or input field, and each black dot in F_2 indicates a category node. The black semi-circles between the input and category fields represent the bidirectional conditional links, including code activation and competition processes (bottom-up processing) and resonance and readout processes (top-down processing)

Preliminaries: fusion ART and STADLART

The episodic memory model proposed in this paper is based on fusion ART. The network is designed to learn cognitive node encoding multi-modal input pattern groups, supporting the recognition and recall of the stored patterns. In this section, we review the basics of the fusion ART and STADLART model.

Fusion ART

Biologically-inspired fusion ART network simulates information processing in the human brain and performs a class of fuzzy operations for multi-modal pattern recognition and association. Fusion ART [4, 5] extends the ART network [10] to handle multi-channel inputs (see Fig. 1). The network model consists of K input channels or input fields $F_1^1, F_1^2, \dots, F_1^K$ (the terms “input channel” and “input field” are interchangeably used in fusion ART networks) and one category field F_2 , which are connected by bidirectional conditional links.

The composition and dynamics of fusion ART are described as follows:

Input vectors: Let I^k denote the M -dimensional input vector of channel k , $I^k = (I_1^k, I_2^k, \dots, I_m^k, \dots, I_M^k)$, where I_m^k denotes input m to channel k , for $m = 1, 2, \dots, M$ and $k = 1, 2, \dots, K$.

Input fields: Let F_1^k for $k = 1, 2, \dots, K$ denote the input field of channel k that receives input vector I^k .

Activation vectors: Let x^k denote the activation vector of input field F_1^k that receives input vector I^k , $x^k = (x_1^k, x_2^k, \dots, x_m^k, \dots, x_M^k)$, where $x_m^k \in [0, 1]$. Because fusion ART uses fuzzy operations (see (1), (3), (4)), the activation vector needs to be further expanded with the complement vector \bar{x}^k , where $\bar{x}_m^k = 1 - x_m^k$. This expansion is called complement coding, which is employed to avoid the problem of “category proliferation” in fuzzy ART [10].

Category field: Let F_2 denote the category field and y represent the activation vector of F_2 , $y = (y_1, y_2, \dots, y_C)$, where C is the number of category nodes in F_2 . Note that, $C - 1$ nodes are committed (learned) and one node is uncommitted (see template matching for details). Initially, fusion ART has only one uncommitted node in F_2 with a weight vector of 1s.

Weight vectors: Let w_j^k denote the weight vector of category node j in F_2 for learning the input of F_1 . Due to the use of complement coding, the weight vector is initialized to $w_{j1}^k = w_{j2}^k = \dots = w_{j2M}^k = 1$. Interested readers may refer to [26] for details on how to use complement coding and fuzzy AND operations to generalize the knowledge encoded in the weight vectors.

Parameters: Each field/channel of the input layer in the fusion ART model comprises four parameters, namely the choice parameters $\alpha^k > 0$, vigilance parameters $\rho^k \in [0, 1]$, contribution parameters $\gamma^k \in [0, 1]$, where $\sum \gamma^k = 1$, and learning rate parameters $\beta^k \in [0, 1]$.

Code activation: For each activation vector x^k , for $k = 1, 2, \dots, K$ in F_1 and the weight vector w_j^k of each category node j in F_2 , the choice value of each category node is defined by the following choice function:

$$T_j = \sum_{k=1}^K \gamma^k \frac{|x^k \wedge w_j^k|}{\alpha^k + |w_j^k|}, \quad (1)$$

where the fuzzy AND \wedge is defined as $(a \wedge b)_i \equiv \min(a_i, b_i)$, and the norm $|\cdot|$ is defined as $|X| = \sum_m^M |x_m|$. This is the bottom-up category search process (F_1 to F_2 in Fig. 1). The choice value T_j reflects the similarity between the activation vectors in F_1 and weight vectors of node j in F_2 . The larger T_j is, the more similar the activation vector is to the category node j .

Code competition: The winner node J in the code competition procedure is the one with the highest choice value, where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \quad (2)$$

Then update the activation vector of category field in F_2 to $y_J = 1$ and $y_j = 0, \forall j \neq J$, i.e., winner takes all.

Template matching: If the matching function of the winner node satisfies the vigilance requirement at each channel, then resonance occurs, such that

$$m_j^k = \frac{|x^k \wedge w_j^k|}{|x^k|} \geq \rho^k, 1 \leq k \leq K. \quad (3)$$

A mismatch reset will occur if any channel does not meet the criteria. Specifically, the choice value T_J is reset to 0, and

a new category node J is selected. The search continues until the selected J satisfies the resonance. If the winner category node is uncommitted, it becomes committed after template learning, and then a new uncommitted node is added to F_2 .

Template learning: The template learning process is performed to encode the knowledge of the input into the weight vector of the winner node. The learning rate β is set to 1 for fast learning.

$$w_J^{k(new)} = (1 - \beta^k)w_J^{k(old)} + \beta^k(x^k \wedge w_J^{k(old)}). \quad (4)$$

Memory readout: The top-down readout process (F_2 to F_1 in Fig. 1) presents the weight vectors of the winner node J in F_2 to the input fields:

$$x^{k(readout)} = w_J^k. \quad (5)$$

STADLART

STADLART aims to learn ADL daily routines and can be seen as two stacked fusion ART networks in three layers. The F_1 layer contains four fields that correspond to the ADL type, time, day, and spatial information. Each category node in the F_2 layer represents a spatial-temporal ADL and is activated as a response to the contextual information presented in the F_1 layer. The activation values of activated spatial-temporal ADL patterns are set to 1s, and the rest are 0s. The F_3 layer contains the daily routine field, where the ADL routine nodes in F_3 represent the serial combinations of spatial-temporal ADLs.

STEM-ADL architecture

To encode daily activities into episodic memory, we propose spatial-temporal episodic memory for ADL (STEM-ADL). STEM-ADL is a three-layer neural network based on the fusion ART model (see Fig. 2). The F_1 layer contains three input fields that encode event-specific attributes, including time, place, and ADL type. The F_2 layer contains only one event field and serves as the category field of event encoding and the input field of episode encoding. Based on activations of F_1 , a cognitive node in the F_2 layer is selected and activated as an event pattern. After all sensor events in a day are learned from F_1 to F_2 , F_2 will form an activation vector y , the length of which is equal to the number of all event nodes currently existing in the event field. In this activation vector y , the activation values decay over time, forming a descending activation pattern to represent the sequence of events. Based on activation vector y , a category node in the F_3 layer is selected and activated as an episode pattern. The bidirectional conditional links between layers still apply to

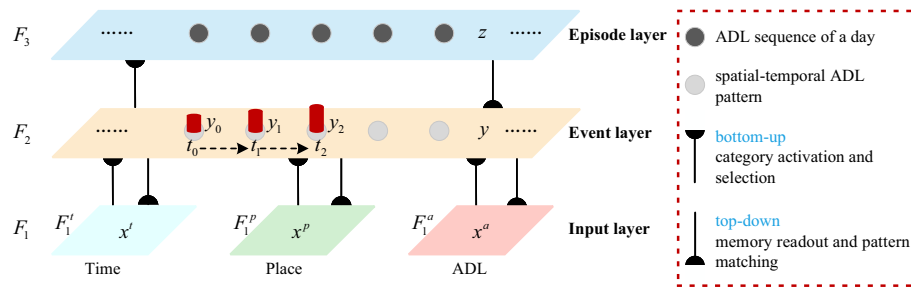


Fig. 2 STEM-ADL architecture. The three-layer self-organizing structure contains an input layer F_1 , an event layer F_2 , and an episode layer F_3 . F_1 has three input channels corresponding to the attributes of time, place, and ADL type. Each light gray dot in F_2 encodes a spatial-

temporal ADL pattern, while each dark gray dot in F_3 encodes the sequence of events in a day. The node activation values decaying over time in F_2 characterize the temporal order of the events in an episode

STEM-ADL. When an episode node in F_3 is selected, the whole episode can be reproduced through the top-down operation (readout process) from F_3 to F_2 . Events in the selected episode can also be reproduced through the top-down operation (readout process) from F_2 to F_1 in the sequence they were saved in the F_2 layer.

Event encoding and retrieval

In STEM-ADL, the input fields encode the event-related attributes in F_1 . The value of time attribute is normalized, and the values of place and ADL type attributes are converted into category variables.

Time vector

In a daily activity data set, the starting time and ending time of activity are given in the form of timestamps. We encode such information explicitly in the time field. Firstly, the time information needs to be normalized. We set the granularity of time to seconds, so the time of a day is $24 \times 60 \times 60 = 86,400$ in seconds. Thus, the normalized starting or ending time is obtained using $\frac{\text{time in seconds}}{86,400}$. Let x^t represent the activation vector of the time field:

$$x^t = (x_s^t, x_e^t, \bar{x}_s^t, \bar{x}_e^t), \quad (6)$$

where x_s^t and x_e^t denote the normalized starting and ending time, respectively, \bar{x}_s^t and \bar{x}_e^t denote the corresponding complement values.

Place vector

The place refers to the spatial information associated with the ADL. Activities take place in different locations in smart homes, such as living room, bedroom, bathroom, etc. The activation vector x^p of the place field is represented as fol-

lows:

$$x^p = (x_1^p, x_2^p, \dots, x_r^p, \bar{x}_1^p, \bar{x}_2^p, \dots, \bar{x}_r^p), \quad (7)$$

where r is the total number of places in consideration, x_i^p is the i th place type, and \bar{x}_i^p is the corresponding complement value. It is obvious that at any time, $\sum x_i^p = 1$.

ADL vector

The ADL types depend on the daily activity data set used. Usually, ADL types include cooking, sleeping, eating, etc. Let x^a represent the activation vector of the ADL field:

$$x^a = (x_1^a, x_2^a, \dots, x_q^a, \bar{x}_1^a, \bar{x}_2^a, \dots, \bar{x}_q^a), \quad (8)$$

where q is the total number of ADL types, x_i^a is the i th ADL type, and \bar{x}_i^a is the corresponding complement value. It is obvious that at any time, $\sum x_i^a = 1$.

For event encoding, relevant fusion ART operations are applied to learn the node in the event layer. For event retrieval, the retrieval cue can comprise information from all three input fields or partial input fields, with or without noise. Based on the presented retrieval cue, STEM-ADL selects the winner node(s) with the highest choice value in F_2 and then reads the relevant information of the retrieved event(s). Multiple similar events may be retrieved when the cue is partial or noisy. Algorithm 1 presents the whole process of event encoding and retrieval.

Episode encoding and retrieval

Temporal order coding of events is a primary component of episodic memory. In the fusion ART model, the activation value of the winner node in the category field is usually set to 1. Therefore, in STADLART, the activated spatial-temporal ADL pattern nodes in the category layer have the same activation value, so the order of events cannot be encoded in its daily

Algorithm 1 Event encoding and retrieval

```

1: Set  $x^k$  in  $F_1$  w.r.t the input vector  $I^k$ 
2: Perform code activation to compute the choice values of all nodes in  $F_2$  (see (1))
3: Perform code competition to select the event node  $J$  with the highest choice value as the winner node in  $F_2$  (see (2))
4: if event encoding process then
5:   while mismatch reset occurs (see (3)) do
6:     Suppress the activation of the current winner node that mismatches
7:     Perform code competition again (see (2))
8:   end while
9:   Perform template learning (see (4))
10: else if event retrieval process then
11:   Read out the weight vectors  $w_j^k$  to  $F_1$  (see (5))
12: end if

```

Algorithm 2 Episode Encoding and Retrieval

```

1: for all sequential events in a day do
2:   Select the winner node  $J$  in  $F_2$  (detailed in Algorithm 1)
3:   Set  $y_J \leftarrow 1$ 
4:   for all the previously activated events nodes in  $F_2$  do
5:      $y_j^{(new)} \leftarrow y_j^{(old)}(1 - \tau)$ 
6:   end for
7: end for
8: Given activation vector  $y$  formed in  $F_2$ 
9: Perform code activation to compute the choice values of all nodes in  $F_3$ 
10: Select the winner node  $J'$  with the highest choice value in  $F_3$  w.r.t  $y$ 
11: if episode encoding process then
12:   the weight vector  $w_{j'}^y$  in  $F_3$  is updated according to the learning rule
13:    $w_{j'}^{(new)} \leftarrow (1 - \beta^s)w_{j'}^{(old)} + \beta^s(y \wedge w_{j'}^{(old)})$ 
14: else if episode retrieval process then
15:   Read out the weight vector  $w_{j'}^y$  to  $F_2$ 
16: end if

```

routine layer. In STEM-ADL, we characterize the temporal order of events by updating the activation value of the activated event nodes and assume that the earlier an event occurs, the smaller the activation value is. Let $t_0, t_1, t_2, \dots, t_n$ denote the increasing timestamps, and y_{t_i} denote the event activation value in F_2 at time t_i , then $y_{t_0} < y_{t_1} < y_{t_2} < \dots < y_{t_n}$. Specifically, when a new node is activated in F_2 , the activation values of the previously activated nodes are decayed proportionally that $y_j^{(new)} = y_j^{(old)}(1 - \tau)$, where τ denotes the predefined decay coefficient and $\tau \in (0, 1)$.

Fields F_2 and F_3 in Fig. 2 form another fusion ART, corresponding to the input field of events and the category field of episodes, respectively. Field F_3 encodes an episode of sequential events in a day with the weight vector represented as $w_{j'}^y$. The parameters involved in episode learning include learning rate β^s , vigilance parameter ρ^s , contribution parameter γ^s , and choice parameter α^s in F_2 .

STEM-ADL can retrieve episodes from various types of cues. The episode retrieval cue can be a subset of any episodes starting from the beginning or any other time point in the day.

Once an episode is selected, the associated spatial-temporal ADL patterns can be reproduced from the selected episode node in F_3 to the event nodes in F_2 by performing the readout operation. Algorithm 2 presents the encoding and retrieval processes of the episode.

ADL retrieval and prediction

STEM-ADL can retrieve specific ADLs using the occupant's spatial and temporal preferences. Moreover, due to its episode learning capability, STEM-ADL can predict subsequent events based on the current event, predicting both the ADL type and starting time in one shot.

ADL retrieval ADL patterns have certain regularity in the temporal and spatial features. For example, an occupant usually has breakfast in the dining room at seven in the morning and leaves home at eight. So specific ADLs can be retrieved from spatial-temporal information.

STEM-ADL activates each spatial-temporal ADL pattern node in F_2 with the given time input x^t and the place input x^p . All event nodes matching x^t and x^p are selected to output the ADL information through a top-down readout operation. The detailed ADL retrieval procedure is presented in Algorithm 3.

Algorithm 3 ADL Retrieval using Spatial-Temporal Information

```

1: Set the time vector  $x^t$  and place vector  $x^p$  w.r.t the given input vectors  $I^t$  and  $I^p$ 
2: Activate each event node  $j$  in  $F_2$  by applying choice function
3:  $T_j \leftarrow \gamma^t \frac{|x^t \wedge w_j^t|}{\alpha^t + |w_j^t|} + \gamma^p \frac{|x^p \wedge w_j^p|}{\alpha^p + |w_j^p|}$  (with ADL information missing in  $x$ )
4: Select the event node  $J$  with the highest choice value in  $F_2$  (see (2))
5: Read out  $w_j^a$  associated with node  $J$  in  $F_2$  (see (5))

```

Subsequent event prediction Performance of ADLs has an intrinsic chronological order, for example meal preparation is typically followed by cooking, eating, and washing dishes. If the smart home could anticipate what the occupant will perform, it can provide better services in the form of activity reminders and recommendations. STEM-ADL can predict the subsequent events based on learned patterns of one's daily activities and the current event. The given current event maps to an event node J in F_2 by event encoding. The activation vector y of F_2 is updated to $y_J = 1$, and $y_j = 0, \forall j \neq J$, which is used as the cue for episode retrieval. The choice value of episode nodes in F_3 is calculated as follows:

$$T_{j'}' = \frac{|y \wedge w_{j'}^y|}{\alpha^s + |w_{j'}^y|}. \quad (9)$$

The episode node(s) with the highest choice value in F_3 is then selected. Because only one event is used as the cue

Algorithm 4 Subsequent event prediction process

- 1: Set activation vectors x^t , x^p , and x^a w.r.t the current event input vectors I^t , I^p , and I^a
- 2: Activate each event node j in F_2 by applying choice function (see (1))
- 3: Select the event node J with the highest choice value in F_2
- 4: Set $y_J \leftarrow 1$, $y_j \leftarrow 0$, $\forall j \neq J$
- 5: Activate each episode node j' in F_3 by applying choice function w.r.t y (see (9))
- 6: Select the episode node(s) J' with the highest choice value in F_3
- 7: Read out $w'_{J'}$ associated with episode node J' in F_3
- 8: Update $y_J \leftarrow w'_{J',J}$ in F_2
- 9: Compute the subsequent event node activation value (see (10))
- 10: All nodes j^* form the candidate event set $(j_1^*, j_2^*, \dots, j_b^*)$
- 11: Read out w_{j^*} associated with event node j^* in F_2
- 12: Obtain the tuple $(\hat{x}^t, \hat{x}^p, \hat{x}^a)$ in F_1
- 13: **if** exist different ADL types in the candidate events **then**
- 14: Select the candidate event with the highest ADL occurrence and the minimum time error
- 15: **else**
- 16: Select the candidate event with the minimum time error
- 17: **end if**

for episode retrieval, multiple episodes containing this event might be retrieved. In the selected episodes, the corresponding activation vector(s) y is then read out from F_3 to F_2 . The activation value y_J of the current event is updated to the readout value at the same index. As introduced in “[Episode encoding and retrieval](#)” section, the activation values of the previously activated events gradually decay over time. The activation value y_{j^*} of the subsequent event j^* can be computed by the following function:

$$y_{j^*} = \frac{y_J}{1 - \tau}. \quad (10)$$

All nodes j^* with an activation value equals to y_{j^*} form the candidate event set, denoted as $(j_1^*, j_2^*, \dots, j_b^*)$. Then, the event-related attributes associated with node j^* are read out from F_2 to F_1 to get the event tuple $(\hat{x}_l^t, \hat{x}_l^p, \hat{x}_l^a)$ for $l = 1, 2, \dots, b$. In episodic memory, prediction is made based on past experiences (i.e., candidate event set). The winner-takes-all strategy applies here as well. If an ADL always follows the current event, we predict it as the subsequent ADL. Let N_l denote the number of occurrences of tuple element \hat{x}_l^a , and the ADL type with the highest N_l value in the candidate set is identified as the prediction result. The winner of ADL types is indexed at L , where $L = \arg \max_l N_l$. In the candidate event set, two or more event nodes may have the same ADL but different starting times (depending on the degree of event generalization). Therefore, when the ADL of the subsequent event is identified, time error is used as another selection criterion to break the tie. Specifically, the predicted starting time should be the one with the minimum error between the starting time of the subsequent event and the ending time of the current event. Note that because we use complement coding and fuzzy AND operations, the output time vector \hat{x}_l^t

represents a generalized interval of the starting time (ending time). The starting time interval is denoted as $[\hat{x}_{l,s}^t, 1 - \hat{x}_{l,s}^t]$ rather than just a single timestamp. The error between the median of the starting time interval and the ending time of the current event is computed by the following function:

$$D_l = |\hat{x}_{l,sm}^t - x_e^t|, l \in L \quad (11)$$

where $\hat{x}_{l,sm}^t$ denotes the median of starting time interval that $\hat{x}_{l,sm}^t = \frac{1}{2}(\hat{x}_{l,s}^t + 1 - \hat{x}_{l,s}^t)$. The winner after competing the errors in time is indexed at L' , where $L' = \arg \min_l D_l$. Finally, the starting time of the subsequent event and the corresponding ADL type are predicted by reading out node L' . See Algorithm 4 for a detailed prediction procedure.

Complexity analysis

The complexity of STEM-ADL is analyzed in terms of both space and time. Let K denote the number of input field attributes, C denote the number of events, B denote the number of episodes, and suppose each episode contain g events on average.

Space complexity

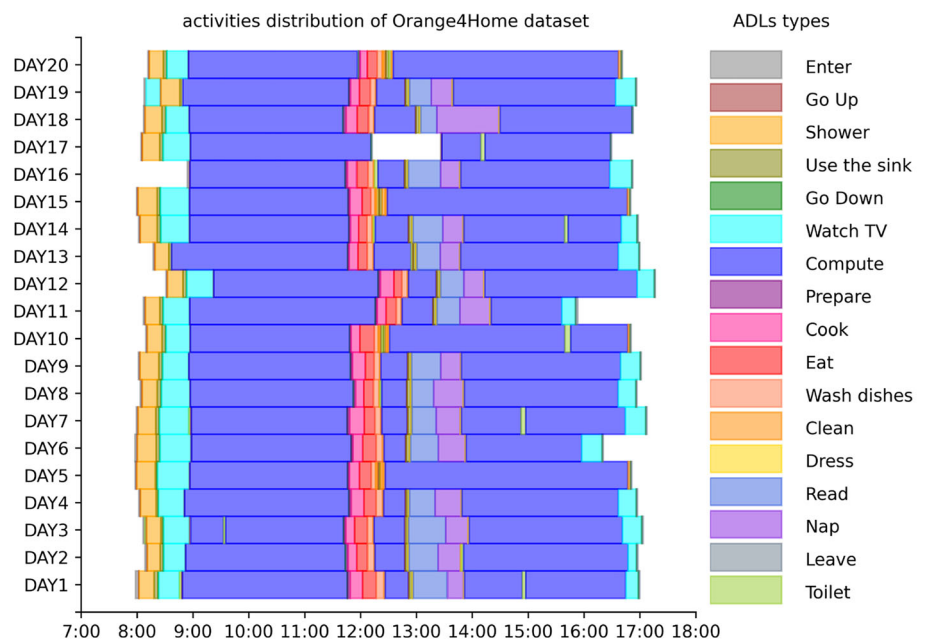
For event encoding, STEM-ADL requires C category nodes to represent the unique events in the F_2 layer and $2CK$ weights between F_1 and F_2 layers including the complements (see “[Event encoding and retrieval](#)” section). Similarly, for episode encoding, it requires B nodes in the F_3 layer and $2BC$ weights between F_2 and F_3 layers to encode B episodes. So the total space requirement without generalization is $B + C + 2BC + 2CK$. Because in most real-world scenarios, there are more number of episodes than the input field attributes, i.e., $B > K$, the space complexity of STEM-ADL is $O(BC)$.

Time complexity

In STEM-ADL, the time complexity of encoding an event is $O(CK)$ due to the resonance search process. Subsequently, STEM-ADL takes an $O(gCK)$ process on average to generate the activation for the sequence to be learned at the F_2 layer. Then the activated pattern in F_2 is matched with all patterns stored in the F_3 layer. When the matching is unsuccessful, it is learned as a new category node in the F_3 layer. The learning process of F_2 – F_3 takes $O(BC^2)$. Because $C > g$ and as discussed earlier $B > K$, $BC^2 > gCK$. Therefore, the time complexity for STEM-ADL to encode an episode is $O(BC^2)$.

In summary, both the space and time complexity of STEM-ADL are reasonably low, which is as expected that the Fusion-ART models have been shown as computationally efficient in various application domains [11, 27, 28].

Fig. 3 Visualization of activities in Orange4Home. The X-axis indicates the activity time (mainly 8 am–5 pm daily), the Y-axis shows the number of days recorded (20 days), and the legends on the right are the activity types (17 types)



Experimental setup

This section introduces the experimental setup, consisting of the data sets, data processing, parameter settings, and baseline models.

Data sets

Smart home-based ADL data sets involve significant sensor deployment and data collection costs, typically recording one or more occupants' activities over a considerably long period of time. In this work, we source for publicly available data sets to conduct experiments. As aforementioned in “Introduction” section that our work focuses on the high-level modeling of ADL patterns instead of low-level recognition of specific activities, we may only rely on those data sets comprising labeled ADL routines performed by one or more occupants in the smart home environment. However, due to various reasons, including privacy and ethical issues, not many collected data sets were made publicly available. Furthermore, among the limited number of publicly available ones comprising labeled ADL routines, only few are suitable to be used to evaluate the performance of our STEM-ADL and other baseline models. For example, in the Cairo data set [29], only a small part of the recorded ADLs are labeled with their performers in this multi-occupant data set. In addition, although ADL labels are given in the single occupant Kasteren Dataset [30], seven ADLs (namely leaving, beverage, sleeping, showering, breakfast, dinner, and toileting) are not rich enough to describe one's daily activities in real life, thus, not suitable for use in our experiments. In the end, we find the following two smart-home data sets suit-

able to be used to evaluate the performance of STEM-ADL and other baseline models. Both data sets have full annotations of ADLs, comprising the starting timestamp, ending timestamp, the location, and the associated activity.

Orange4Home The Orange4Home data set collected the ADLs of a single occupant for 20 successive work days (i.e., four weeks) [6] (see Fig. 3). The ADL types and the corresponding location and number of occurrence are shown in Table 1. The activities occurred in eight areas within the apartment: entrance, living room, kitchen, bedroom, office, bathroom, toilet, and staircase.

OrdonezB The OrdonezB data set collected 23 days of activities performed by a single occupant in his home [7] (see Fig. 4). The ADL types and the corresponding location and number of occurrence are shown in Table 2. The activities occurred in five areas: kitchen, living room, bathroom, bedroom, and entrance.

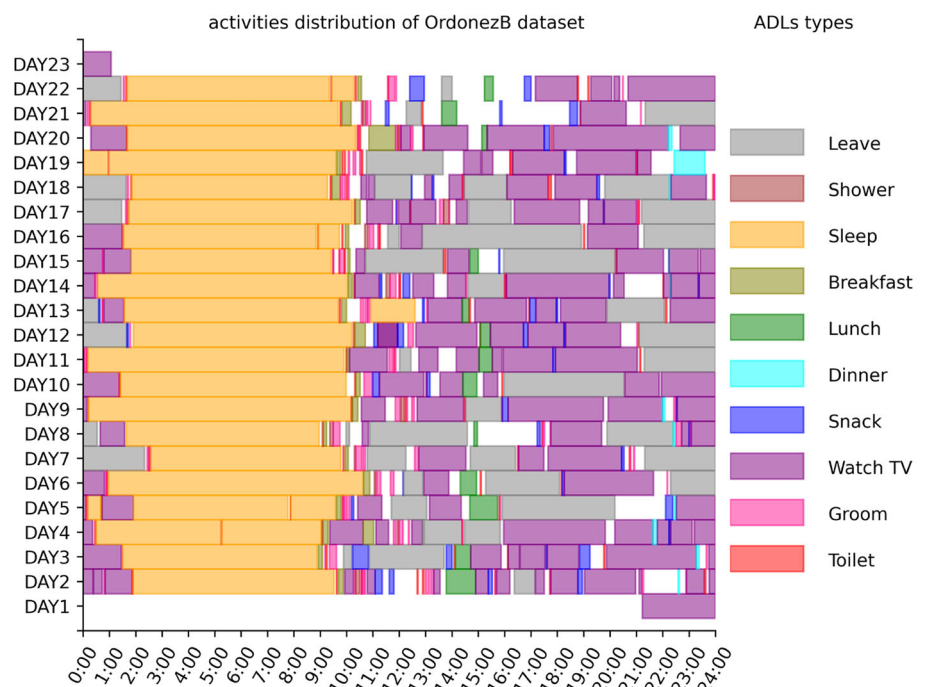
As shown in Tables 1 and 2, the number of ADL types captured in Orange4Home is much more than that in OrdonezB. Nonetheless, the ADL routines in Orange4Home are much more regular than those in OrdonezB (see Figs. 3 and 4), which leads to the difference in the prediction accuracy for all models between the two data sets (see “Experimental results” section).

Data processing

We encode a day consisting of 24h of activities as one episode. As an artifact, events in OrdonezB across midnight are split into two events, i.e., the first ends at 23:59:59, and the second begins at 00:00:00 on the following day. Based

Table 1 Location and number of occurrence of ADLs in Orange4Home

Place	Activity	Number	Place	Activity	Number
Bedroom	Dressing	30	Bathroom	Using the sink	38
	Reading	15		Using the toilet	9
	Napping	15		Showering	19
	Cleaning	3		Cleaning	4
Living Room	Eating	19	Kitchen	Preparing	19
	Cleaning	19		Cooking	19
	Watching TV	18		Washing the dishes	19
	Computing	15		Cleaning	4
Staircase	Going up	57	Office	Computing	46
	Going down	57		Watching TV	14
Entrance	Entering	21	Toilet	Cleaning	4
	Leaving	21		Using the toilet	8

Fig. 4 Visualization of activities in OrdonezB. The *X*-axis indicates the activity time (00:00–23:59), the *Y*-axis shows the number of days recorded (23 days), and the legends on the right are the activity types (10 types)**Table 2** Location and number of occurrence of ADLs in OrdonezB

Place	Activity	Number	Place	Activity	Number
Kitchen	Breakfast	22	Bathroom	Showering	11
	Lunch	13		Toileting	93
	Dinner	11		Grooming	113
	Snack	47	Living	Watching TV	116
Bedroom	Sleeping	29	Entrance	Leaving	38

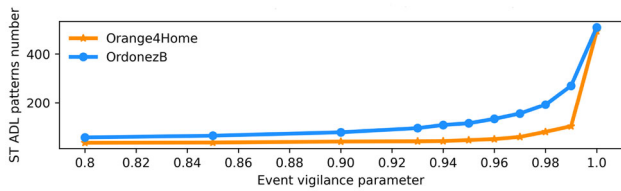


Fig. 5 Relationship between the number of ST ADL patterns and vigilance values

on such division, November 11, 2012 (Day 1) and December 3, 2012 (Day 23) both only have one event (see Fig. 4), so we remove these two days in all experiments. We only consider the prediction of events within the same episode and not across the episodes. The encoding of the input fields is as follows:

Time vector is obtained by normalizing the starting and ending timestamps of an ADL (see “Time vector” section). For example, in Orange4Home, Day 1, the activity “Computing” started at 08:48:41 and ended at 11:45:44. After normalization and complement coding, the corresponding time vector is (0.367, 0.490, 0.633, 0.510) (see (6)). The length of the time vector is 4.

Place vector represents the location where each ADL is performed. Referring to the previous example when introducing time vector, activity “Computing” occurred in the “Office.” The corresponding place vector is (0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1), where the sixth bit represents “Office.” The length of the place vector for Orange4Home is 16 (8 places with their complements) and 10 for OrdenezB (5 places with their complements).

ADL vector represents the recorded ADL types of the data set. Following the same example used before, the ADL vector is (0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1), where the seventh bit indicates the “Computing” activity. The length of the ADL vector for Orange4Home is 34, and 20 for OrdenezB.

Parameter settings

Parameters ρ^k , one for each channel k in F_1 , regulate the generalization across events. In the template matching process (see (3)), vigilance parameters are used as threshold criteria to judge whether a resonance occurs. Figure 5 shows the total number of events generated with different vigilance values for the Orange4Home and OrdenezB data sets. When the vigilance value is set to 1, each activity instance is considered a distinct spatial-temporal ADL pattern. A reduction of 0.01 in the event-level vigilance value results in an approximately 80% and 50% decrease in the number of generalized ADL patterns for the Orange4Home and OrdenezB data sets, respectively (see Fig. 5). With vigilance values smaller than 1, STEM-ADL clusters similar events into the same event node

and generates the generalized pattern through template learning (see (4)). Table 3 shows the number of events and episodes generated with different vigilance combinations. Parameter ρ^s regulates the generalization across episodes. As shown in Table 3, the number of episodes for Orange4Home decreases as ρ^s drops from 1 to 0.9, indicating that the episodes in the Orange4Home are highly similar, while those in OrdenezB are significantly different (i.e., no generalization performed to merge episodes). To balance the specificity and generalization ability, we set $\rho^t = 0.99$ for Orange4Home and $\rho^t = 0.95$ for OrdenezB in F_1 , which corresponds to the variety of the two data sets (see Figs. 3 and 4). Because the place and ADL vectors are binary-valued, we set both ρ^p and ρ^a to 1, thus requiring an exact match on these two input fields.

The following parameter values of STEM-ADL are used in all the experiments conducted in this work. For event encoding between F_1 and F_2 , $k \in \{t, p, a\}$, choice parameters α^k are set to 0.001 to avoid NaN in the choice function (see (1)), contribution parameters $\gamma^k = 0.333$, which means the impact among the three input fields is equal, learning rate $\beta^k = 1$ for fast learning. For episode encoding between F_2 and F_3 , there is only one input channel, and the parameters are set as $\alpha^s = 0.001$, $\gamma^s = 1$, $\beta^s = 1$, and $\rho^s = 1$. The delay coefficient τ is set to 0.1. Except for the discussion of ρ^t , all the other parameters in STEM-ADL take the default values used in various fusion ART models [2, 11, 13]. Although STEM-ADL has several parameters, it is pretty straightforward to set their values.

Baseline models

For comprehensive performance comparisons, we select not only traditional machine learning approaches, such as Gaussian Naive Bayes (GNB) [31], Support Vector Machine (SVM) [32], and Decision Tree (DT) [33], but also deep learning approaches, such as Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU). We also compare it with STADLART and a self-organizing incremental neural network named SOINN+ [34]. Their input vectors are composed of the same features as STEM-ADL.

Compared to STEM-ADL, there is an extra day field in STADLART, and the day vector x^d is defined as follows:

$$x^d = (x_1^d, x_2^d, \dots, x_9^d, \bar{x}_1^d, \bar{x}_2^d, \dots, \bar{x}_9^d), \quad (12)$$

where x_1^d to x_7^d denote “Monday” to “Sunday,” x_8^d and x_9^d denote “Weekday” and “Weekend,” respectively.

LSTM controls the flow of information through three types of gates to perform memory functions. SOINN+ inherits ideas from SOINN [35] to execute associative memory tasks and is relatively more capable of handling noisy data

Table 3 The number of events and episodes generated with different vigilance values

ρ^t	ρ^s	Orange4Home (20 days)		OrdenezB (21 days)	
		Events No.	Episodes No.	Events No.	Episodes No.
1.00	1.00	492	20	508	21
1.00	0.90	492	20	508	21
0.99	1.00	102	19	269	21
0.99	0.90	102	14	269	21
0.95	1.00	45	19	114	21
0.95	0.90	45	9	114	21

streams. For RNN, LSTM, GRU, we set *neurons number* = 128, *epochs* = 300, *batch size* = 64, “MSE” for loss function, and “Adam” as the optimizer. We tune the parameters of all baseline models iteratively and report their best performance in the next section.

Experimental results

In this section, experimental results on event retrieval, ADL retrieval, and subsequent event prediction are evaluated in terms of accuracy and F1-score. In the event retrieval experiments, all events in the data sets are used for retrieval cues. In the ADL retrieval and subsequent event prediction experiments, the “leave-one-day-out” approach is adopted to verify the validity of the proposed model, which is K-fold cross validation, where K is equal to the number of days. The verification algorithm takes the activity instances of one day as the test set and the activity instances of the other days as the training set, calculates the average value and uses it for model evaluation.¹ All experiments were conducted on a computer equipped with an Intel i7-10750H 2.60GHz CPU and 16GB RAM.

Event retrieval

After ADL instances are encoded into the episodic memory model, we conduct event retrieval experiments using noisy and partial cues to evaluate the robustness of STEM-ADL. We use retrieval accuracy as the evaluation metric. If the event which generates the corresponding retrieval cue is within the retrieved event set, it is considered a successful retrieval. Otherwise, it is a failure. Therefore, retrieval accuracy is defined as the rate of the number of successful retrievals over the overall amount of retrieval cues used. For a fair comparison, the day vector in STADLART is set to all 1s (1s are used to denote do-not-care).

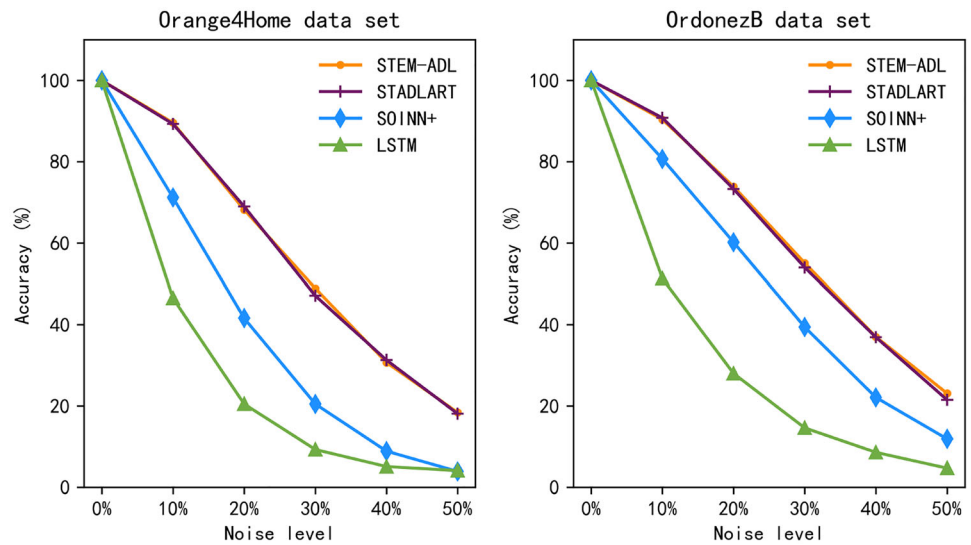
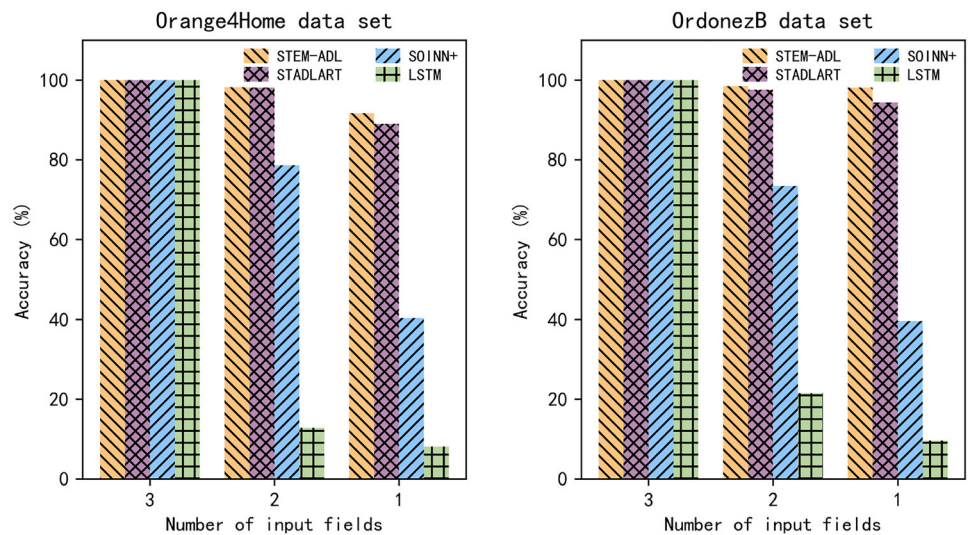
Noisy cue In this experiment, the presence of these noisy cues is simulated by injecting noise into the data sets. Specifically,

random noise is added to each attribute of the input field with varying probabilities. For the time field, the attributes are continuous, and the values are subjected to Gaussian noise with $N\%$ probability, where $N \in \{0, 10, 20, 30, 40, 50\}$. First, a random number r is selected, where $r \in [0, 100]$. If $r \leq N$, the corresponding time attribute is subjected to the addition of Gaussian noise with zero mean and a standard deviation of 0.01 (not applicable for complements, whose values are computed after the addition of noise), and its value is capped within the range of $[0, 1]$ for correctness. For the place and ADL fields, their attributes are binary-valued, and each bit of the attributes is toggled (i.e., $x' = 1 - x$) with a probability of $N\%$ (again, not applicable for complements).

Partial cue Partial cues are incomplete cues with missing attribute values. We randomly select one, two, or three attributes among the time, place, and ADL type fields to generate the partial cues. For LSTM and SOINN+, the missing attributes are filled with 0s. For example, when the partial cue contains only one attribute and the time vector is selected, the partial cue is $(x^t, 0, 0)$.

Because both types of retrieval cues involve randomness, the accuracy varies across different runs. Hence, we run each experiment ten times and report the average results. Figures 6 and 7 show the retrieval results using noisy and partial cues, respectively. When the noise level is 0 (i.e., exact retrieval), all models achieve 100% accuracy. When the noise level increases, the performance of all models declines. Overall, STEM-ADL and STADLART perform comparably well in event retrieval, and both outperform SOINN+ and LSTM, indicating that the fusion ART models are better at handling imperfect information. Similarly, all models achieve 100% accuracy with no missing attributes. With missing attribute values, STEM-ADL shows slightly better performance than STADLART, and both models outperform SOINN+ and LSTM. This is due to the fact that STEM-ADL adopts a multi-channel fusion mechanism (also adopted by STADLART), while SOINN+ is a single-channel learning model that takes all multi-source context information as a whole. When the number of attributes decreases to one or two, the performance of LSTM declines sharply, indicating that it might not handle partial cue retrieval tasks well.

¹ The source code will be provided if the article is accepted.

Fig. 6 Retrieval accuracy for noisy cues**Fig. 7** Retrieval accuracy for partial cues

ADL retrieval

We report the ADL retrieval accuracy of different models with the given spatial and temporal information in Table 4. Because the temporal information in STADLART contains the associated day information (see (12)), we provide both retrieve results with and without the day field. As shown in Table 4, STEM-ADL outperforms all other models in terms of accuracy and F1-score on both data sets. The underlying reasons for the high ADL retrieval accuracy achieved by STEM-ADL (91.2% for Orange4Home and 76.9% for OrdonezB) are mainly twofold. First, it encodes ADL patterns across all days for better generalization. Second, it holistically learns the association among the ADL and the corresponding spatial-temporal information. Specifically, STEM-ADL does not differentiate the day type and captures the generic ADL patterns across all days (see “ADL retrieval and prediction” section). Meanwhile, STADLART

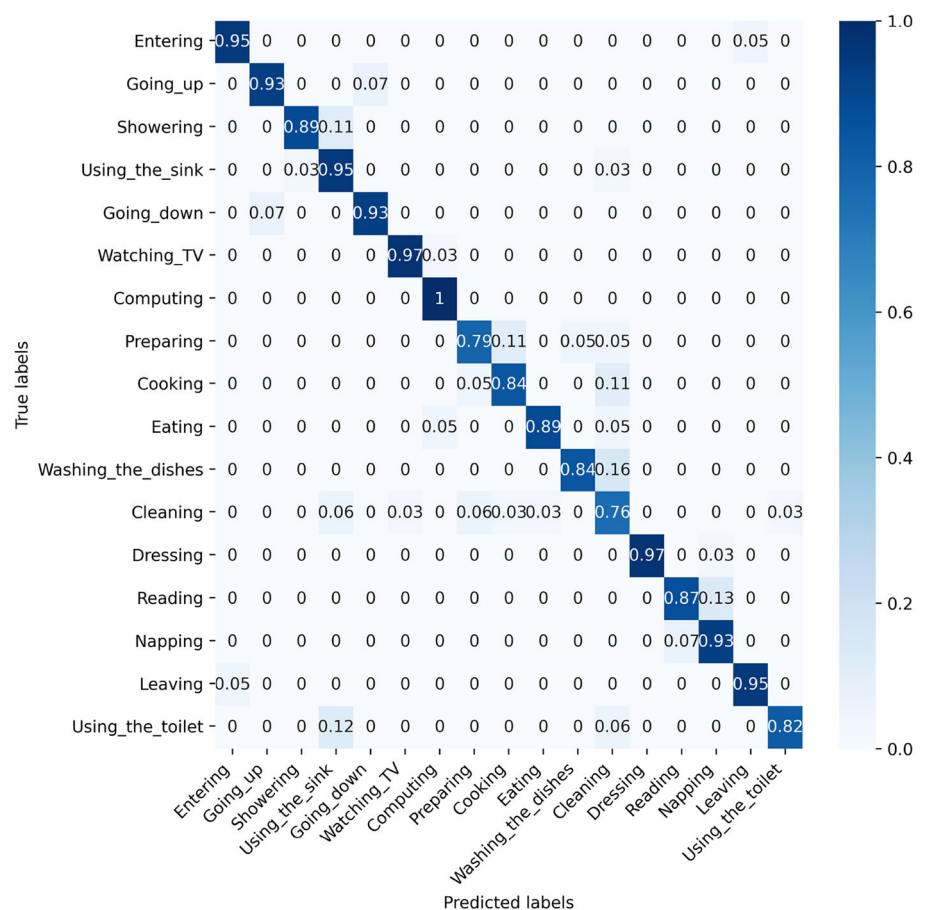
encodes the ADLs performed on different day types (see (12)) into disjoint patterns. This explains why STEM-ADL outperforms STADLART in the activity retrieval task. In contrast to other machine/deep learning models that learn an approximation function of inputs (time and place in this case) mapping to outputs (ADL), STEM-ADL adopts associative learning among all three input fields of F_1 . Therefore, STEM-ADL is naturally suitable for ADL retrieval tasks.

More details on the classification results of STEM-ADL are provided by presenting the confusion matrix. In Orange4Home, activities have high temporal and spatial stability. “Computing” is the most stable activity, followed by “Watching TV” and “Dressing.” “Cleaning” may occur in any room of the apartment and has the highest rate of incorrect retrieval (see Fig. 8), and “Toileting” is easily misjudged due to the randomness of its occurrence time. In OrdonezB, activities such as “Leaving,” “Sleeping,” and “Watching TV” can be correctly identified based on the

Table 4 Comparisons on ADL retrieval performance

Model	Orange4Home		OrdenezB	
	Accuracy	F1-score	Accuracy	F1-score
GNB	0.663±0.058	0.673±0.096	0.712±0.071	0.749±0.063
SVM	0.576±0.021	0.491±0.037	0.711±0.056	0.571±0.093
DT	0.888±0.121	0.874±0.149	0.746±0.091	0.723±0.114
RNN	0.686±0.049	0.640±0.077	0.712±0.060	0.650±0.095
LSTM	0.684±0.043	0.613±0.056	0.718±0.052	0.665±0.087
GRU	0.680±0.053	0.617±0.067	0.718±0.053	0.666±0.094
SOINN+	0.763±0.094	0.757±0.136	0.714±0.061	0.692±0.096
STADLART (with day)	0.785±0.129	0.723±0.170	0.515±0.080	0.500±0.114
STADLART (w/o day)	0.826±0.116	0.801±0.154	0.642±0.102	0.629±0.103
STEM-ADL	0.912±0.090	0.899±0.117	0.769±0.085	0.765±0.106

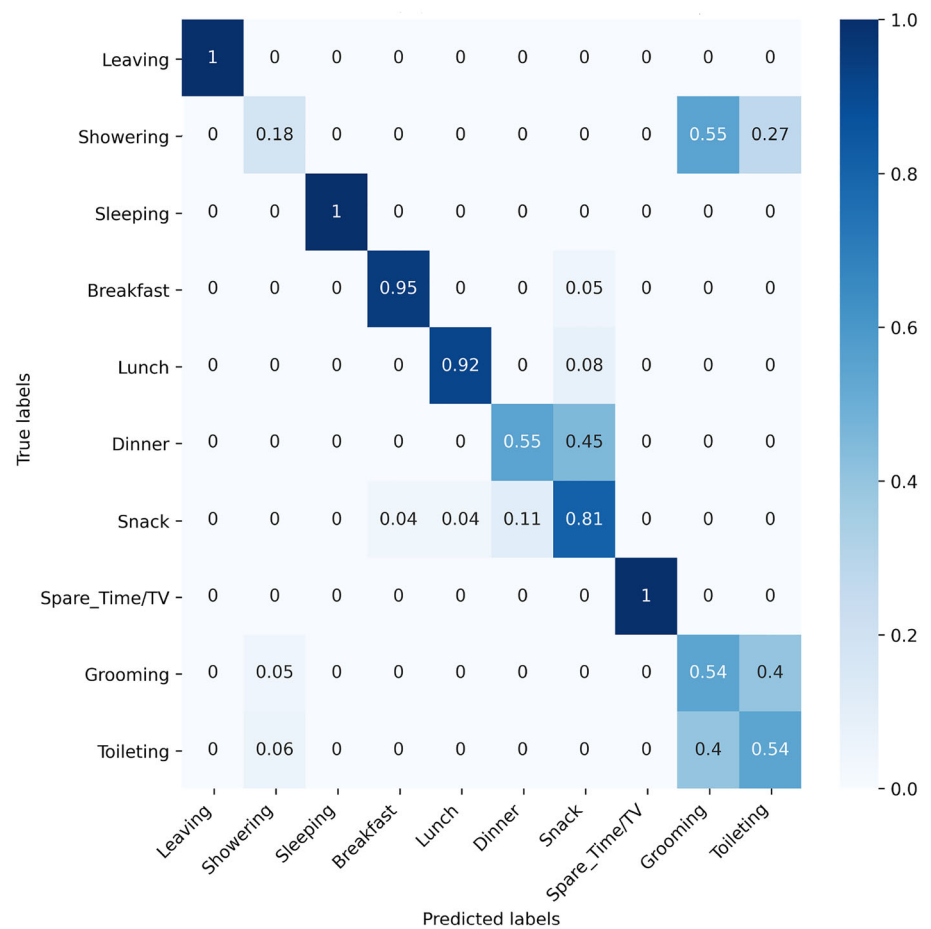
Bold values indicate the best performance

Fig. 8 Confusion matrix of ADL retrieval in Orange4Home

given spatial-temporal information. However, activities such as “Grooming,” “Toileting,” and “Showering” take place in the bathroom and can easily be confused with each other. Activity “Dinner,” which takes place over a wide range of time from 21:47 to 23:35, has no consistent temporal pattern and is usually confused with “Snack” (see Fig. 9).

Subsequent event prediction

As introduced in “ADL retrieval and prediction” section, the ADL and time vectors can be read out after selecting the winner event node in the prediction task.

Fig. 9 Confusion matrix of ADL retrieval in OrdonezB**Table 5** Comparisons on subsequent event prediction accuracy

Model	Orange4Home		OrdonezB	
	Accuracy	F1-score	Accuracy	F1-score
GNB	0.718±0.057	0.718±0.068	0.211±0.058	0.284±0.087
SVM	0.701±0.042	0.674±0.042	0.322±0.070	0.159±0.036
DT	0.886±0.063	0.894±0.047	0.359±0.068	0.311±0.088
RNN	0.883±0.095	0.887±0.096	0.342±0.081	0.216±0.053
LSTM	0.889±0.098	0.892±0.097	0.353±0.071	0.231±0.058
GRU	0.893±0.092	0.894±0.092	0.337±0.079	0.220±0.060
SOINN+	0.733±0.055	0.732±0.079	0.325±0.081	0.260±0.113
STADLART	0.884±0.077	0.884±0.093	0.320±0.061	0.288±0.081
STEM-ADL	0.920±0.059	0.921±0.058	0.381±0.070	0.333±0.098

Bold values indicate the best performance

ADL prediction The ADL prediction performance of various models on both data sets is shown in Table 5. The proposed STEM-ADL is superior to the existing models in terms of both accuracy and F1-score. Following the same example used in “Data processing” section, predict that the activity occurring after “Computing” is “Going down.” It can be interpreted by the behavior routine of the occupant to infer the subsequent activity. STEM-ADL encodes the context-

aware information of the activity and models the sequence of spatial-temporal activity patterns.

Starting time prediction In addition, we compare the predicted starting time of subsequent events, which is another important aspect of real-world applications. Because of the use of fuzzy AND operations and complement coding, STEM-ADL learns the generalized event nodes, representing aggregations of the same ADL occurring at close times.

Table 6 Subsequent event prediction accuracy with different vigilance values

Date Set	ρ^t	ρ^s	ADL		MAE (in minutes)	
			Accuracy	F1-score	Interval	Median
Orange4Home	0.95	1	0.919 ± 0.070	0.913 ± 0.076	3.60	15.84
	0.99	1	0.920 ± 0.059	0.921 ± 0.058	3.95	6.99
OrdonezB	0.95	1	0.381 ± 0.070	0.333 ± 0.098	24.62	42.78
	0.99	1	0.326 ± 0.077	0.289 ± 0.082	19.20	21.95

Bold values indicate the best performance

Table 7 Starting time prediction of the subsequent event compared across various models

Model	MAE (in minutes)	
	Orange4Home	OrdonezB
RNN (median error)	9.78	74.18
LSTM (median error)	<u>5.22</u>	69.90
GRU (median error)	5.28	69.57
STADLART (median error, see (14))	12.05	50.74
STADLART (interval error, see (13))	10.00	50.45
STEM-ADL (median error, see (14))	6.99	<u>42.78</u>
STEM-ADL (interval error, see (13))	3.95	24.62

The best interval error is highlighted in bold, and the best median error is underlined

Therefore, it can provide a time interval in which the occupant often performs a specific activity. We expect the event to occur within the predicted starting time interval. As with the activity prediction example, the predicted event starts within the intervals “11:41:07” and “11:52:21.”

For a fair comparison with the deep learning models that produce a single timestamp prediction, we compute and present two types of mean average error (MAE) for STEM-ADL and STADLART, namely the interval error and median error. For interval error, if the ground-truth starting time of an event falls within the time interval predicted by STEM-ADL (or STADLART), the amount of error is deemed as 0. Otherwise, the error is computed as the absolute difference between the ground-truth value and its nearest interval boundary. Specifically, let t_{gi} denote the ground-truth starting time of the i th subsequent event, and $t_i = [t_{s1i}, t_{s2i}]$ denote the predicted starting time interval of the i th subsequent event for STEM-ADL or STADLART. The interval MAE is computed as follows:

$$MAE_{interval} = \frac{1}{H} \sum_{i=1}^H \begin{cases} 0, & \text{if } t_{gi} \in t_i. \\ \min(|t_{gi} - t_{s1i}|, |t_{gi} - t_{s2i}|), & \text{if } t_{gi} \notin t_i. \end{cases} \quad (13)$$

where H denotes the overall number of subsequent events in the test set. For median error, the median timestamp of the predicted starting time interval is taken as the single prediction value and used for MAE computations as the other

models do. The median MAE is computed as follows:

$$MAE_{median} = \frac{1}{H} \sum_{i=1}^H |t_{gi} - \text{mean}(t_{s1i}, t_{s2i})|. \quad (14)$$

Note that for STEM-ADL and STADLART, only the generalized time interval is preserved after training. Both models do not keep track of the exact data samples learned during the generalization process for efficiency. Therefore, although we name MAE_{median} as “median error,” we make use of the mean of the interval boundaries (see (14)).

Although for the interval errors, if the predicted interval is large enough, the probability of obtaining zero error is high. However, the prediction of starting time should be considered in conjunction with the prediction accuracy of ADL. It is not true that the larger the prediction interval is, the better the result will be. Per the introductions in “Fusion ART” and “Parameter settings” sections, ρ^t and ρ^s regulate the generalization of events and episodes, respectively. The larger the ρ^t is, the higher specificity of the event is, on the contrary, the smaller the ρ^t is, the higher generalization of the event is. In the activity prediction task, spatial-temporal ADL patterns are learned from historical behaviors, and therefore, appropriate values are selected to balance specificity and generalization ability. Due to the difference between the two data sets, ρ^t takes diverse values to achieve reasonable generalization. Table 6 shows the prediction results with different vigilance values and the optimal ADL prediction results are consistent with the threshold of the optimal event generalization. ADL’s accuracy and time error are shown as not being

able to be optimized simultaneously because the time interval of activity becomes larger after generalization. Note that the results reported in the subsequent event prediction correspond to the best ADL accuracy.

The MAE errors of all models are reported in Table 7. Regarding interval error, STEM-ADL has the minimum prediction error for both data sets, outperforming all the others, including STADLART, which also computes the same type of interval error. Regarding the median error, STEM-ADL achieves the third best in Orange4Home (with minor differences to LSTM and GRU) and outperforms all the others in OrdenezB.

For STEM-ADL, ADL prediction is made by mining human behavior patterns, events generalization are needed. The time complexity is due to the generalization of computing winner nodes and the winner is subject to code activation and template matching. All event-related information can be retrieved together by following the bottom-up activation and top-down memory readout procedures (see Algorithms 1, 2 and 4, respectively). Using the vigilance parameter combination as introduced in “Parameter settings” section, the average model training time is 4.35 s for Orange4Home, 5.78 s for OrdenezB. For event prediction, it takes an average of 11.87 ms to predict subsequent ADL. On the contrary, when deep learning models handle the prediction task with time series, they generally need two networks to predict the ADL type (classification task) and the starting time (regression task), respectively. LSTM, for example, has an average training time of 23.86 s and prediction time of 29.33 ms, which are both significantly longer than those of STEM-ADL. Therefore, for subsequent event prediction, the deep learning models have no advantage on both accuracy and efficiency.

Conclusion

We propose a spatial-temporal episodic memory approach called STEM-ADL, that employs fusion ART modeling based on a self-organizing neural network for ADL learning and prediction. STEM-ADL explicitly considers ADL and related context information including starting time, ending time, and spatial information. ADL data streams are aggregated at the bottom layer to generate spatial-temporal ADL patterns represented by events to encode the temporal and spatial properties of ADLs, and a series of spatial-temporal ADL patterns are encoded as an episode. Compared with STADLART, STEM-ADL encodes the activated sequence of events in a gradient decay pattern, which enriches the context-aware knowledge and improves the overall performance. The data sets Orange4Home provided by Orange Labs and OrdenezB offered by UCI are applied to evaluate the effectiveness of the proposed STEM-ADL. STEM-ADL

generalizes an occupant’s ADL spatial and temporal patterns over several days and provides information for further behavior prediction. In all experiments, STEM-ADL was compared with STADLART and other baseline algorithms such as LSTM and an associative memory model. The results show that STEM-ADL is superior to the baseline algorithms in various aspects. STEM-ADL model has high robustness in encoding and recalling stored events using partial and noise search cues. In addition, STEM-ADL well retrieves ADLs from the given spatial-temporal information and further accurately predicts the subsequent event, which comprises ADL type and starting time. As discussed, STEM-ADL requires less time and space complexities.

By utilizing the STEM-ADL model, an intelligent system will know the typical activity patterns of the occupant. Based on this knowledge, smart home-based healthcare applications and other health-assistive services become possible. For example, the intelligent system can predict the occupant’s next activity to provide a lifestyle recommendation to improve people’s quality of life.

Due to the limitations of data sets, we are only able to study the behavior of a single occupant living in an intelligent environment. In the future, we intend to enlarge the capability of STEM-ADL to deal with more challenging scenarios, such as using ADLs collected from the same smart home to identify individuals living together when such real-world data sets are available.

Author Contributions Xinjing Song: Conceptualization, Methodology, Software, Investigation, Validation, Writing - original draft, Visualization. Di Wang: Conceptualization, Methodology, Software, Review & editing, Supervision. Chai Quek: Conceptualization, Methodology, Resources, Review & editing, Supervision. Ah-Hwee Tan: Conceptualization, Methodology, Software, Review & editing, Funding acquisition, Supervision. Yanjiang Wang: Conceptualization, Funding acquisition, Supervision. All authors have read and approved the final manuscript.

Funding This research is supported by the National Natural Science Foundation of China (Grant No. 62072468), the China Scholarship Council (201906450051), Fundamental Research Funds for the Central Universities (18CX06059A), the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No. AISG2-RP-2020-019), and the SMU-A*STAR Joint Lab in Social and Human-Centered Computing (Grant No. SAJL-2022-HAS001).

Data Availability The data sets analyzed in the current study are available from the following links <https://amiquel4home.inria.fr/orange4home/> and <https://archive.ics.uci.edu/ml/machine-learning-databases/00271/>.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adap-

tation, 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

- Wadley VG, Okonkwo O, Crowe M, Ross-Meadows LA (2008) Mild cognitive impairment and everyday function: evidence of reduced speed in performing instrumental activities of daily living. *Am J Geriatr Psychiatry* 16(5):416–424. <https://doi.org/10.1097/01.jgp.0000310780.04465.13>
- Gao S, Tan A-H, Setchi R (2019) Learning ADL daily routines with spatiotemporal neural networks. *IEEE Trans Knowl Data Eng* 33(1):143–153. <https://doi.org/10.1109/tkde.2019.2924623>
- Xu Z, Wang G, Guo X (2023) Event-driven daily activity recognition with enhanced emergent modeling. *Pattern Recognit* 135:109149. <https://doi.org/10.1016/j.patcog.2022.109149>
- Tan A-H, Carpenter GA, Grossberg S (2007) Intelligence through interaction: Towards a unified theory for learning. In: *International Symposium on Neural Networks*, pp. 1094–1103. https://doi.org/10.1007/978-3-540-72383-7_128
- Tan A-H, Subagdja B, Wang D, Meng L (2019) Self-organizing neural networks for universal learning and multimodal memory encoding. *Neural Netw* 120:58–73. <https://doi.org/10.1016/j.neunet.2019.08.020>
- Cumin J, Lefebvre G, Ramparany F, Crowley JL (2017) A dataset of routine daily activities in an instrumented home. In: *International Conference on Ubiquitous Computing and Ambient Intelligence*, pp. 413–425. https://doi.org/10.1007/978-3-319-67585-5_43
- Ordóñez F, De Toledo P, Sanchis A (2013) Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors* 13(5):5460–5477. <https://doi.org/10.3390/s130505460>
- Mueller ST, Shiffrin RM (2006) REM II: a model of the developmental co-evolution of episodic memory and semantic knowledge. In: *International Conference on Learning and Development (ICDL)*
- Nuxoll AM, Laird JE (2007) Extending cognitive architecture with episodic memory. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1560–1564
- Carpenter GA, Grossberg S, Rosen DB (1991) Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Netw* 4(6):759–771. [https://doi.org/10.1016/0893-6080\(91\)90056-b](https://doi.org/10.1016/0893-6080(91)90056-b)
- Chang P-H, Tan A-H (2017) Encoding and recall of spatio-temporal episodic memory in real time. In: *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1490–1496. <https://doi.org/10.24963/ijcai.2017/206>
- Yang C-Y, Gamborino E, Fu L-C, Chang Y-L (2022) A brain-inspired self-organizing episodic memory model for a memory assistance robot. *IEEE Trans Cogn Dev Syst* 14(2):617–628. <https://doi.org/10.1109/TCDS.2021.3061659>
- Wang D, Tan A-H, Miao C (2016) Modeling autobiographical memory in human-like autonomous agents. In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 845–853. https://ink.library.smu.edu.sg/sis_research/6277
- Wang D, Tan A-H, Miao C, Moustafa AA (2019) Modelling autobiographical memory loss across life span. *Proc AAAI Conf Artif Intell* 33(01):1368–1375. <https://doi.org/10.1609/aaai.v33i01.33011368>
- Maitre J, Bouchard K, Bertuglia C, Gaboury S (2021) Recognizing activities of daily living from UWB radars and deep learning. *Expert Syst Appl* 164:113994. <https://doi.org/10.1016/j.eswa.2020.113994>
- Han C, Zhang L, Tang Y, Huang W, Min F, He J (2022) Human activity recognition using wearable sensors by heterogeneous convolutional neural networks. *Expert Syst Appl* 198:116764. <https://doi.org/10.1016/j.eswa.2022.116764>
- Dang LM, Min K, Wang H, Piran MJ, Lee CH, Moon H (2020) Sensor-based and vision-based human activity recognition: a comprehensive survey. *Pattern Recogn* 108:107561. <https://doi.org/10.1016/j.patcog.2020.107561>
- Meng L, Miao C, Leung C (2017) Towards online and personalized daily activity recognition, habit modeling, and anomaly detection for the solitary elderly through unobtrusive sensing. *Multimed Tools Appl* 76:10779–10799. <https://doi.org/10.1007/s11042-016-3267-8>
- Wu S, Rendall JB, Smith MJ, Zhu S, Xu J, Wang H, Yang Q, Qin P (2017) Survey on prediction algorithms in smart homes. *IEEE Internet Things J* 4(3):636–644. <https://doi.org/10.1109/JIOT.2017.2668061>
- Alam MR, Reaz MBI, Ali MM (2011) SPEED: An inhabitant activity prediction algorithm for smart homes. *IEEE Trans Syst Man Cybern A Syst Hum* 42(4):985–990. <https://doi.org/10.1109/tsmca.2011.2173568>
- Nazerfard E, Cook DJ (2015) CRAFFT: an activity prediction model based on Bayesian networks. *J Ambient Intell Humaniz Comput* 6(2):193–205. <https://doi.org/10.1007/s12652-014-0219-x>
- Cumin J, Lefebvre G, Ramparany F, Crowley JL (2020) PSINES: activity and availability prediction for adaptive ambient intelligence. *ACM Trans Auton Adapt Syst* 15(1):1–12. <https://doi.org/10.1145/3424344>
- Jain A, Singh A, Koppula HS, Soh S, Saxena A (2016) Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3118–3125. <https://doi.org/10.1109/icra.2016.7487478>
- Du Y, Lim Y, Tan Y (2019) A novel human activity recognition and prediction in smart home based on interaction. *Sensors* 19(20):4474. <https://doi.org/10.3390/s19204474>
- Fatima I, Fahim M, Lee Y-K, Lee S (2013) A unified framework for activity recognition-based behavior analysis and action prediction in smart homes. *Sensors* 13(2):2682–2699. <https://doi.org/10.3390/s130202682>
- Wang D, Tan A-H (2014) Creating autonomous adaptive agents in a real-time first-person shooter computer game. *IEEE Trans Comput Intell AI Games* 7(2):123–138. <https://doi.org/10.1109/tciaig.2014.2336702>
- Meng L, Tan A-H, Miao C (2019) Saliency-aware adaptive resonance theory for large-scale sparse data clustering. *Neural Netw* 120:143–157. <https://doi.org/10.1016/j.neunet.2019.09.014>
- Meng L, Tan A-H, Wunsch DC (2015) Adaptive scaling of cluster boundaries for large-scale social media data clustering. *IEEE Trans Neural Netw Learn Syst* 27(12):2656–2669. <https://doi.org/10.1109/tnnls.2015.2498625>
- Cook DJ (2012) Learning setting-generalized activity models for smart spaces. *IEEE Intell Syst* 27:32–38. <https://doi.org/10.1109/mis.2010.112>
- Kasteren TV, Noulas AK, Englebienne G, Kröse BJA (2008) Accurate activity recognition in a home setting. In: *the 10th International*

- Conference on Ubiquitous Computing (UbiComp), pp. 1–9. <https://doi.org/10.1145/1409635.1409637>
31. Liu L, Wang S, Su G, Huang Z-G, Liu M (2017) Towards complex activity recognition using a Bayesian network-based probabilistic generative framework. *Pattern Recognit* 68:295–309. <https://doi.org/10.1016/j.patcog.2017.02.028>
 32. Fleury A, Vacher M, Noury N (2010) Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. *IEEE Trans Inf Technol Biomed* 14(2):274–283. <https://doi.org/10.1109/titb.2009.2037317>
 33. Logan B, Healey J, Philipose M, Tapia EM, Intille S (2007) A long-term evaluation of sensing modalities for activity recognition. In: *International Conference on Ubiquitous Computing*, pp. 483–500. https://doi.org/10.1007/978-3-540-74853-3_28. Springer
 34. Wiwatcharakoses C, Berrar D (2020) SOINN+, a self-organizing incremental neural network for unsupervised learning from noisy data streams. *Expert Syst Appl* 143:113069. <https://doi.org/10.1016/j.eswa.2019.113069>
 35. Shen F, Osamu H (2006) An incremental network for on-line unsupervised classification and topology learning. *Neural Netw* 19(1):90–106. <https://doi.org/10.1016/j.neunet.2005.04.006>

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