Self-organizing neural networks for human activity modeling and analysis

Gao, Shan

2020

Shan, G. (2020). Self-organizing neural networks for human activity modeling and analysis. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/143281>

<https://doi.org/10.32657/10356/143281>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 20:29:49 SGT

NANYANG TECHNOLOGICAL UNIVERSITY

Self-Organizing Neural Networks for Human Activity Modeling and Analysis

A Thesis Submitted to
the Interdisciplinary Graduate School
of the Nanyang Technological University

by

Shan Gao

in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

July 15, 2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

15th July 2020

Date

Shan Gao

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

15 July 2020

Date

Prof. Tan Ah-Hwee

Authorship Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed journal / from papers accepted at conferences in which I am listed as an author.

The work of ADLART model described in Chapter 3 is published as **Shan Gao, Ah-Hwee Tan “User Activity Pattern Learning - A multi-memory Modeling Approach”, International Joint Conference on Neural Networks (IJCNN) 2014.**

The contribution of the co-authors are as follows:

- (i) Prof. Tan provided the initial project direction and co-designed the study.
- (ii) I prepared the manuscript drafts. The manuscript was revised by Prof. Tan.
- (iii) I conducted the experiments, collected the data, and interpreted the data.
- (iv) Prof. Tan provided advices on experiment design and analysis methods.

The work of RADLART model described in Chapter 3 is published as **Shan Gao, Di Wang, Ah-Hwee Tan “Progressive Sequence Matching for ADL Plan Recommendation”, IEEE/WIC/ACM Intelligent Agent Technology Conference (IAT) 2015.**

The contribution of the co-authors are as follows:

- (i) Prof. Tan provided the initial project direction and co-designed the study.

- (ii) I prepared the manuscript drafts. The manuscript was revised by Prof. Tan.
- (iii) Di Wang co-designed the wellness scores in four quotients for ADLs.
- (iv) I conducted the experiments, collected the data, and interpreted the data.
- (v) Prof. Tan provided advices on experiment design and analysis methods.

The work of STADLART model described in Chapter 4 is published as **Shan Gao, Ah-Hwee Tan, Rossi Setchi “Learning ADL Daily Routines with Spatiotemporal Neural Networks”, IEEE Transactions on Knowledge and Data Engineering 2019, DOI: 10.1109/TKDE.2019.2924623**

The contribution of the co-authors are as follows:

- (i) Prof. Tan provided the initial project direction and co-designed the study.
- (ii) I prepared the manuscript drafts. The manuscript was revised by Prof. Tan and Prof. Rossi Setchi.
- (iii) I conducted the experiments, collected the data, and interpreted the data.
- (iv) Prof. Tan provided advices on experiment design and analysis methods.

The work of ST-ART model described in Chapter 4 is published as **Shan Gao, Ah-Hwee Tan “An Autonomous Agent for Learning Spatiotemporal Models of Human Daily Activities”, Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2016: 1453-1454.**

The contribution of the co-authors are as follows:

- (i) Prof. Tan provided the initial project direction and co-designed the study.
- (ii) I prepared the manuscript drafts. The manuscript was revised by Prof. Tan.

(iii) I conducted the experiments, collected the data, and interpreted the data.

(iv) Prof. Tan provided advices on experiment design and analysis methods.

15th July 2020

Date

Shan Gao

Acknowledgments

This thesis would not have been possible without the guidance and the help of several individuals who kindly contributed their valuable assistance in the preparation and completion of this study.

First and foremost, I wish to render my heartfelt appreciation to my advisor, Professor Ah-Hwee Tan, who has supported me throughout this project with his patience and knowledge. Without his consistent support and supervision, the research would not have been so enriching and fulfilling.

My gratitude also goes to my co-advisor Assistant Professor Yun Ke Chang, co-advisor Associate Professor Lihui Chen, mentor Dr. Zhiqi Shen, and research group members: Dr. Wenwen Wang, Dr. Budhitama Subagdja, and Dr. Di Wang. Thanks for their excellent advice and contribution to make this research more complete and mature.

I would like to thank the Interdisciplinary Graduate School and NTU-UBC Research Center of Excellence in Active Living for the Elderly for their financial support.

Last but not least, I will thank my parents and my wife Yu Song. They are always there to support me and give me inner peace when I am under pressure.

Abstract

Activities of Daily Living (ADLs) refer to self-care activities performed by an individual in his/her residence on a daily basis. ADLs are widely used to measure the functional status of elderly people by health professionals. Given the importance of ADLs as indicators of wellness status, there has been active research in ADL detection and recognition in the recent years. However, whereas ADL recognition technologies are fairly mature, modeling ADL patterns remains a challenge and currently, there are limited works in this field. This thesis is aimed at developing computational methods to model and analyze ADL patterns, specifically daily ADL routines that a person frequently performs in a day as well as inter-ADL patterns encoding temporal relationship among individual ADLs.

This thesis firstly presents a multi-memory model designed to learn a person's ADL routines. By mimicking the human multi-memory systems, the model, named Activities of Daily Living Adaptive Resonance Theory (ADLART), encodes the user's activity routines through an episodic memory module and generalizes them into semantics ADL routines. Based on ADLART, another model, named Recommendation Activities of Daily Living Adaptive Resonance Theory (RADLART), is developed to recommend ADLs to users. By learning the association among users' profiles, daily ADL routines, and the wellness scores, RADLART provides ADL recommendations for a healthy lifestyle.

The second part of the thesis looks at encoding the spatiotemporal information of ADLs explicitly. A neural network model, named spatiotemporal ADL adaptive resonance theory (STADLART), is proposed which explicitly encodes the spatiotemporal

information of ADLs. Specifically, it utilises a spatiotemporal-ADL layer to encode spatiotemporally generalised ADLs. The results of the experiments show that STADLART could effectively learn the users' routines. Building upon STADLART, a more general model named spatiotemporal ART (ST-ART) is proposed, which extends STADLART with an added spatiotemporal layer encoding the user's preference of space and time combinations. ST-ART is applied to the ADL domain and has the potential to be used in other fields.

Beyond daily ADL routines, inter-ADL patterns, specifically, the causal relationship among ADLs, is studied in this work. A fusion ART model named hypothesis adaptive resonance theory (H-ART) is developed, which is capable of constructing hypotheses rapidly from a small number of observations. Hypotheses are constructed based on observations of repeated pairs of events. Observing that human may change behaviour patterns over time, H-ART also dynamically rejects hypothesis when new observations contradict them. Though illustrated in the ADL domain, H-ART can also be applied to other time series and stream data mining problems.

In summary, this thesis presents a family of neural network models for modeling and analysing different aspects of ADL patterns. These models are capable of handling real-time online data. This capability has been evaluated with synthetic and real-life data sets. Nevertheless, there are possible extensions which have not been worked through in this thesis and are left for future studies. Examples are long-term ADL monitoring, wherein the time horizon of analysis spans multiple years, and the association between health conditions and ADL patterns, which may help to unveil possible causes of illnesses.

Contents

Acknowledgements	i
Abstract	ii
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Background and Motivations	1
1.2 Research Problems and Methodology	4
1.3 Summary of Contributions	6
1.4 Organization of the Thesis	7
2 Literature Review	9
2.1 Human Activity Modeling	9
2.1.1 Action Detection	10
2.1.2 Activities of Daily Living Recognition	13
2.1.3 Activity Pattern Learning	14
2.2 Causal Learning	16
2.3 Adaptive Resonance Theory	17
2.3.1 Fusion ART	17
2.3.2 EM-ART	21

3	Multi-Memory Model of Daily ADL Routine	24
3.1	Introduction	24
3.2	ADLART Model	25
3.2.1	The ADLART Model Architecture	25
3.2.2	Episodic Memory Learning	28
3.2.3	Semantics Memory Learning	29
3.2.4	Evaluation	31
3.3	Recommendation ADLART: RADLART	35
3.3.1	The RADLART Model	35
3.3.2	Recommendation	40
3.3.3	Evaluation	43
3.4	Summary	47
4	Spatial Temporal Modeling of Daily ADL Routines	49
4.1	Introduction	49
4.2	STADLART Architecture	54
4.2.1	Encoding ADLs in STADLART	54
4.2.2	Encoding Daily ADL Routines	58
4.3	Learning and Retrieval of Daily Routines	59
4.3.1	Model Training	59
4.3.2	Daily Routine Readout	59
4.3.3	Spatiotemporal Information Retrieval	60
4.4	ADL Weight Normalization and Customization	61
4.4.1	ADL weight normalization	61
4.4.2	Customizable ADL weights	61

4.5	Experiments on Synthetic Data	62
4.5.1	Synthetic Data Generation	62
4.5.2	Experiments and Evaluation	63
4.6	Experiment on Real-life Data Set	66
4.6.1	CASAS Data Set and Preprocessong	66
4.6.2	Experiments on spatiotemporal ADL category generalization . . .	67
4.6.3	Experimentd on daily routine categories generation	68
4.6.4	Comparing STADLART with other algorithms	69
4.6.5	Experiment on feature weight normalization and customization . .	70
4.7	A General Spatiotemporal ART Model: ST-ART	72
4.7.1	Introduction	72
4.7.2	The ST-ART Model Input Fields	73
4.7.3	Encoding and Retrieval	73
4.7.4	Experiments	77
4.8	Summary	81

5 Sequential Causal Learning by real-time hypothesis formulation and testing 83

5.1	Introduction	83
5.2	Hypothesis adaptive resonance theory (H-ART) architecture	85
5.3	Hypothesis learning using H-ART	87
5.3.1	Current Episode Maintenance and Hypothesis Building	87
5.3.2	Null Hypothesis Rejection	88
5.3.3	Prediction of near future event	89
5.3.4	Interestingness learning	90

5.4	Experiments on Synthetic Data Set	91
5.4.1	Synthetic data generation	91
5.4.2	The vigilance parameter on F_2^2	92
5.4.3	The effect on the time window size	92
5.4.4	Prediction of future event	93
5.4.5	Experiment on Interestingness Learning	94
5.5	Experiments on Real-life Data Set	94
5.5.1	CASAS Data Set	94
5.5.2	Hypothesis Learning on Sensory Data	95
5.5.3	Hypothesis learning on ADL Data	96
5.6	Summary	97
6	Conclusion	98
6.1	Summary of Contributions	98
6.2	Future Work	100
6.2.1	Automatic Vigilance Value Assignment	100
6.2.2	Sophisticated Spatial Field Design	101
6.2.3	Imperfect Observations	101
	Author's Publication	102
	References	103

List of Figures

2.1	Action detection: hierarchical decision structure	11
2.2	Action detection: k-nearest neighbour approach	11
2.3	Action detection: HMM model	13
2.4	Action detection: motion primitives	14
2.5	ADL Recognition: kitchen ADL	15
2.6	ART: the generic fusion ART architecture	19
2.7	EM-ART: architecture.	21
2.8	EM-ART: $F_1 \rightarrow F_2$ Operation	22
2.9	EM-ART: $F_2 \rightarrow F_3$ Operation	23
3.1	ADLART: ADLART Architecture	26
3.2	ADLART: The episodic sub-model	26
3.3	ADLART: Experiment on vigilance parameter	33
3.4	RADLART: The RADLART architecture	35
3.5	RADLART: wellness scores improvement by different starting times . . .	45
3.6	RADLART: wellness scores improvement by different acceptance rates . .	46
4.1	STADLART: STADLART architecture	55
4.2	STADLART: synthetic routine template	63
4.3	STADLART: a sample of learned ADL routines	64

4.4	STADLART: the experiment on vigilance values	68
4.5	STADLART: two ADL routines learned by STADLART	69
4.6	STADLART: top two ADL routines learned by STADLART-N	71
4.7	ST-ART: The ST-ART architecture	72
4.8	ST-ART model: Experiment on different vigilance	78
5.1	H-ART: architecture.	85
5.2	H-ART: causal relationship between events in simulation	91

List of Tables

3.1	ADLART: day information input	27
3.2	ADLART: ADLs used in experiments	32
3.3	ADLART: simulation statistics of Monday	32
3.4	ADLART: accuracy in combined experiments	34
3.5	ADLART: semantic memory learning	34
3.6	RADLART: wellness scores in four quotients	39
3.7	RADLART: the major phases of the recommendation agent	40
3.8	RADLART: preferences by real elderly	47
4.1	STADLART: ADL types used by STADLART	56
4.2	STADLART: day information input	57
4.3	STADLART: spatial types	58
4.4	STADLART: assignment of ADL weight according to its importance . . .	62
4.5	STADLART: Experiment on different vigilance values	64
4.6	STADLART: the external index scores for the synthetic data set	65
4.7	STADLART: annotated ADL types in the CASAS data set	66
4.8	STADLART: the overall average silhouette width for the data sets	70
4.9	STADLART: Comparing STADLART and STADLART-N	70
4.10	STADLART: Comparing STADLART, STADLART-C, and STADLART-NC	71

4.11	ST-ART: breakfast categories created	79
4.12	ST-ART: top frequency ADL daily routine categories learnt	80
4.13	ST-ART: comparing the ST-ART and the ADLART	81
5.1	H-ART: experiment on different window sizes	92
5.2	H-ART: experiment on H-ART prediction	93
5.3	H-ART: experiment on interestingness learning	94
5.4	H-ART: annotated ADLs in the CASAS data set	95
5.5	H-ART: learnt hypothesis on sensory data	96
5.6	H-ART: learnt hypothesis on ADL data	96

Chapter 1

Introduction

1.1 Background and Motivations

Ageing of population is a common issue faced by countries all over the world. According to World Health Organization [1], the proportion of elderly people will be doubled from 11% in 2000 to 22% in 2050, and there will be about two billion people aged 60 and above by 2050. This trend will bring significant challenges to various countries in health care. Meanwhile, this is also a big opportunity in developing new economic models and bringing innovations into the elderly care industry. Research, e.g. [2], has shown elderly people have various problems ranging from sensory loss, reduced mobility, increased variance in moving time, and reduced speech capacity, to diminished information process capacity. These problems often result in the moving of the elderly people from homes to care institutions, adding much cost to both individuals and society. As such, many countries have invested in technology development for relieving the problems faced by elderly people, so that they can better enjoy their old age.

Living in care institutions for the elderly imposes a tremendous financial burden on elderly people and their families. Additionally, there are other disadvantages for elderly people staying in elderly care institutions. For example, compared to family members, the care centre staff may have a less emotional connection to elderly people [4], and it

takes time for elderly people to get used to the new environment in the care institution. They may need to change their daily routines, which have lasted a few decades. Moreover, elderly people living in care institutions are likely to develop negative feelings of losing freedom and independence, thinking that they have been abandoned by family members and society [5]. Because of these considerations, it is highly recommended that elderly people live in their own homes rather than moving to care institutions if it is not necessary [6]. Consequently, people have started to look upon technologies that could help in enabling elderly people to live independently at home as long as possible.

In recent years, hardware sensors are getting cheaper, widely available, and becoming standard accessories. Generally speaking, two types of sensors are used in people activity data gathering. The first type is ambient sensors, like IP cameras and sound recorder, which are typically fixed in indoor locations. It is obvious that this type of sensors could provide precise location information of the user. However, due to privacy issues, several places, like the living room, are unlikely to have a camera installed to monitor the users' behaviour. The other group of sensors are wearable sensors, such as accelerometer and gyroscope-based sensory devices. These sensors become available through smartphones, wearable wristbands, and smartwatches. Compared with cameras and sound recorders, wearable sensors are more user-friendly and make full-day monitoring possible. With the help of these sensory devices, there is a broad range of options in user monitoring and modeling.

Smart home [7] [8] is an idea that the house could be equipped with information and communication technologies in order to relieve the problems the elderly people have and assist them to stay in their own home comfortably over a long period. Various technologies and new concepts have been developed and made available in the smart home field over the years. Ambient intelligence (AmI) [9] is another concept associated

with smart homes that the intelligent agents working for the smart home better be more passive than active. For better usability, many aspects of human could be modeled, for instance, the static information modeling, the mental modeling using sophisticated psychological models, the physical movement modeling, and the activity modeling. The activity aspect is more directly linked to the health condition of users, which is the research focus of this thesis. Besides the physical condition, activity statistics also could reflect the mental status of the person to a certain extent. Generally, mental or emotional disordered people may have messy life routines which are different from their regular life routines. Everyone has his own daily and long-term activity habits. In this research, we mainly look at the users' ADL, which is widely used to measure the functional status of a person. Generally, there are two subcategories of ADLs performed by people, namely the basic ADLs for maintaining essential well-being, and the instrumental ADLs for living independently and respectably. Both categories could provide information on the health status of the users.

With the knowledge of users' ADL patterns like their daily routines and relationships among their ADLs, the smart home could provide personalized services to the users. The effective modeling of users' ADL patterns could help on various applications ranging from life assistance, health monitoring, and financial planning, to product recommendation. As an example application, based on the profile of the user, ADL pattern, and healthcare knowledge, ambient intelligence systems could provide activity recommendations for the user to perform healthier activities. Another potential application is abnormal activity detection. The knowledge of the user's ADL routines can be used to detect ADL patterns that are not performed before and generate a warning to the user, his relatives, or caregivers. The third service is long-term daily routine recommendation, where the system provides suggestions of ADL routines that best match his profile and health condition.

The fourth service is long-term health prediction. As elderly people usually have sensory loss over time, the system could predict the length of time that the elderly people can live independently before go to health care institutions or additional caregivers are needed. In order to make these services available, this research shall investigate the problem of modeling and analyzing activities of daily living and build models that could be used in this field.

1.2 Research Problems and Methodology

This thesis focuses on designing models to learn patterns of activities of daily living and applying these models to real-world applications.

First, this thesis explores the modeling of activity routines in memory models inspired by the human memory systems, including notably episodic memory and semantic memory [10]. Episodic memory receives and stores information about temporally dated events and temporal-spatial relations among these events. By using this form of memory, routines of activities are stored as episodes, and during recall, the sequence of activities can be retrieved. The statistics of such episodes can thus represent the habit of the user. On the other hand, semantic memory is the organized knowledge a person possesses about words and other verbal symbols and about the rules, formulas, and algorithms for the manipulation of these symbols. People tend to associate the routines of activities with semantic symbols, e.g., the day of the week. Similarly, by using semantic memory, a model could retrieve a typical activity routine of a particular day. To my best knowledge, there is no existing research on modeling human activities in human-inspired memory models.

Second, this thesis looks at the spatial and temporal modeling of activities. Human activities are associated with temporal-spatial information. This information reveals the habit of the user and activities performed in different time and location may have

different health impacts. In the existing literature, researchers [11] formulated human activity modeling as a spatiotemporal pattern-matching problem on top of the sequence of symbolic information produced by a sensor network. In this work, the proposed algorithm generates a transitional probability model between key ADLs to represent human activity patterns. Another group of researchers [12] applies Episode Discovery (ED) [13] to an ADL sequence to find regular patterns. These works are on the semantic form of activity patterns. There is no existing work on modeling ADL sequences in an episodic memory form which emphasizes the temporal-spatial relations among the activities.

Third, besides the activity routines, the inter-activity relationship is also of interest in this thesis. From observations of real-life human activity data sets [14], it is noticed that some activities always follow other activities. In causal learning, this scenario can be seen as effective causal relationships [15] that two events may not be actually in a cause-effect relationship, but in the application domain, they can be treated as in the cause-effect relationship. At the same time, some activities only follow other activities for some time, and after that, the user changed his habit. Based on these observations, the traditional Bayesian inference [16], which takes excessive training data, may not be suitable in this problem.

In summary, there are mainly two aspects of the thesis. In terms of research, bio-inspired models are designed and developed to bridge the gaps in human activity pattern learning, which have not been addressed in existing models. In terms of applications, the proposed models are extended to be used in real-life applications that could help elderly people in different aspects.

1.3 Summary of Contributions

The key contributions of this thesis are summarized as follows. First, for user activity pattern representation, this thesis proposes a multi-memory neural network model, named Activities of Daily Living Adaptive Resonance Theory (ADLART), utilizing an episodic memory neural network named EM-ART [17] as well as a semantic memory model based on fusion Adaptive Resonance Theory (fusion ART) [54]. In ADLART, both the episodic memory and the semantic memory of activities routines are learned from online input data samples. Empirically, the proposed model is validated by an extensive set of experiments using both synthetic and real-world data sets. As an application, a recommendation model named Recommendation Activities of Daily Living Adaptive Resonance Theory (RADLART) is developed based on ADLART for activity recommendation. The work of ADLART model was published as **“User Activity Pattern Learning - A multi-memory Modeling Approach”** [19] in the International Joint Conference on Neural Networks (IJCNN) 2014. The work of RADLART model was published as **“Progressive Sequence Matching for ADL Plan Recommendation”** [20] in the IEEE/WIC/ACM Intelligent Agent Technology Conference (IAT) 2015.

Second, as an extension to ADLART, this thesis proposes an episodic-memory-based spatial-temporal model, named Spatiotemporal Activities of Daily Living Adaptive Resonance Theory (STADLART) to learn user activity patterns. Inherited from ADLART, STADLART is also a fusion-ART based neural network. Similar to ADLART, the performance of STADLART is validated using both synthetic data set and real-world data set. Normalization and customization algorithms are proposed to enhance the performance of STADLART further. In addition, a general model, named Spatiotemporal Adaptive Resonance Theory (ST-ART) is proposed to work on time series in other fields. The preliminary work of STADLART model was published as **“Learning routines of**

activities of daily living with spatiotemporal neural networks” [21] in the 5th International Conference on Ageless Aging (ICAA) 2017. The complete work of STADLART was published as **“Learning ADL Daily Routines with Spatiotemporal Neural Networks”** [22] in the IEEE Transactions on Knowledge and Data Engineering 2019. The work of ST-ART model was published as **“An Autonomous Agent for Learning Spatiotemporal Models of Human Daily Activities”** [23] in the proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems.

Third, to learn the causal relationship between activities, this thesis proposes an online causal learning neural network, named H-ART, utilizing the episodic memory model. H-ART detects event associations and continuously maintains a set of hypotheses from observations, where repeated observed associations are quickly learned. When users change their habit over time, H-ART could drop old knowledge and learn the new ones.

1.4 Organization of the Thesis

Chapter 2 reviews the literature in activity detection, activity pattern learning, causal learning, and fusion-ART.

In chapter 3, a multi-memory model ADLART is presented with an extension for recommendation applications.

In chapter 4, an episodic memory model emphasizing spatial-temporal features, named STADLART, is presented together with enhancement algorithms on normalization and activity weighting. A generic version that could be applied to other domains is presented. The proposed models are validated with synthetic data sets as well as real-world data sets.

In chapter 5, a causal learning model using ART neural network is presented. The proposed model is evaluated both from the theoretical and empirical perspectives.

Finally, chapter 6 concludes this thesis and discusses potential future work.

Chapter 2

Literature Review

This chapter reviews the literature on human activity modeling, adaptive resonance theory, and causal learning. The review on human activity modeling covers the topics of gesture level action detection, the ADL level activity recognition, and the inter-activity level pattern modeling. The inter-activity pattern modeling review provides state-of-the-art work in this field. For causal learning, Bayesian causal inference techniques and the notion of effective causal learning are briefly discussed. Bayesian causal inference is the most used technique in causal learning and it serves to provide a baseline to compare with the newly proposed causal learning mechanism.

Adaptive resonance theory is the foundation theory of the chosen neural network models to tackle the research questions. We firstly review the literature for the architecture and mechanism of a typical fusion ART. After that, we have a review of the episodic memory adaptive resonance theory (EM-ART), which is an episodic memory model utilizing the adaptive resonance theory.

2.1 Human Activity Modeling

To understand human activity patterns, researchers have proposed various models [24] of activity patterns ranging from short-time actions to lifelong habits. In this research,

we categorise the tasks of human activity modeling into three levels, namely the action detection, ADL recognition, and inter-activity pattern modeling. Actions are defined as gesture level behaviour that are performed over a short period of time. They are building blocks for higher-level behaviours. Whereas ADLs are functional behaviour that human perform to complete certain tasks, inter-activity pattern modeling goes one level higher, which describes the patterns among activities.

2.1.1 Action Detection

Action detection refers to the discovery of a gesture level human movement which are usually performed over a short period of time, typically within a few seconds. These movements could be treated as component blocks for ADLs. Actions include rising hand, lying down, sitting up, walking, walking upstairs, etc. There are currently two groups of literatures on action detection. The first group of literatures belongs to the field of computer vision, which use video data and computer vision techniques to abstract features from video clips [25] [24]. In these computer vision literatures, there are mainly two approaches, namely the space-time approach and the sequential approach. Space-time approach does not extract features on a frame-by-frame basis. They consider a video as a 3-D volume of pixel intensities and extend standard image features such as scale-space extrema, spatial filter responses to the 3-D case. The sequential approach either extracts a set of features from each frame for matching with some templates, or imposes a model (e.g. HMMs, LDSs) on the motion temporal dynamics. The second group of literatures is based on body-mounted sensors [26]. Most of the existing works of activity classification are different from each other by choice of sensors (inertial and body parameter), windowing techniques, feature generation, feature selection, activity set, and classification method. Figure 2.1 shows an example of a hierarchical decision approach,

and Figure 2.2 depicts a simple illustration of the k-nearest neighbour approach. Similar to the computer vision approach, the techniques in this group of literatures could be split into space-time methods and sequential methods.

The features extracted from video clips and features obtained from body-mounted sensors could potentially be put together to form a bigger feature set. Feature selection schemes can then be used to select features that have a higher discriminative power from the feature pool. After that, classification techniques are applied on these features.

Figure 2.1: Hierarchical decision structure, adopted from [26]

Bayes classifier and decision tree approach [27] are used to classify five types of pre-defined activities from a single accelerometer reading on the wrist. In this work, features are separated into three categories, namely time domain (e.g. standard deviation), fre-

Figure 2.2: K-nearest neighbour approach, adopted from [26]

quency domain (e.g. frequency-domain entropy), and spatial domain (e.g. orientation variation). After preprocessing the data, principal component analysis (PCA) is applied to reduce the size of the feature set. After that, a Bayesian classification function as shown below is used:

$$C(f_1, \dots, f_n) = \underset{c}{\operatorname{argmax}} p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c). \quad (2.1)$$

In the recent literatures, hidden Markov models (HMM) (Figure 2.3) are commonly used to deal with the activity recognition problem. In a statistical Markov model, it is assumed the system is modeled to be a Markov process with unobserved (hidden) states. In simpler Markov models, e.g., a Markov chain, the states are directly visible to observers, and therefore the state transition probabilities are the only parameters. However, in HMM, the state is not directly visible to the observers, whereas the output, which is dependent on the state, is visible to the observers. There is a probability distribution over the possible output tokens in each state. As a result, the sequence of tokens generated by an HMM provides information about the sequence of states. 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; a model is still referred to as a 'hidden' Markov model even if these parameters are known precisely. In motion detection works, the readings from sensors are the observed output which is dependent on internal states. The HMM parameter sequences is a human motion feature which could be used to classify different motions. For example, in the work [28], the researchers first use a filter algorithm to count the number of appearances within a time interval and then translate the raw data to state data using Equation 2.2 below.

$$\text{If } T_{sum} \geq T_{th} * T_{interval}, S_{ix} = 1; \text{If } T_{sum} < T_{th} * T_{interval}, S_{ix} = 0 \quad (2.2)$$

Figure 2.3: A HMM model

After this, the parameter values of the hidden Markov model are calculated, and finally, a forward algorithm is used to get the probability of the observation sequence.

However, there are some limitations in using HMM. One problem is that HMM is usually used in predefined domains where the system knows the type of activities, and HMM is hard to be used in an unsupervised learning environment, e.g, learning the universal set of human motions without prior knowledge. It is hard to predefine all possible gesture level movements by human and train a system using sensors to identify those movements. Some online incremental approaches are gaining more interests, e.g. [29].

Besides HMM, the Bag-of-Features (BoF)-based approach [30] is also used. In this work, they firstly perform feature extraction from motion sensor data and then construct a set of motion primitives. After that, they build activity models using histograms of these primitive symbols, and finally, they use a multi-class support vector machines (SVMs) with a linear kernel as the learning machine. An example of activity representation using BoF is shown in Figure 2.4.

2.1.2 Activities of Daily Living Recognition

As mentioned in the previous chapter, ADLs refers to the daily self-care activities performed by an individual in his residence, outdoor, or both. ADLs are usually used as a measurement of the functional status of a person. Compared to action detection, ADL recognition is a higher level of human activity modeling.

Figure 2.4: An example of activity representation using motion primitives, adopted from [30]

In ADL recognition, temporal clues, e.g., the sequence of actions, and spatial clues, e.g., the fixed sensors in the places of different ADLs, are used. Fixed cameras are widely used [31] [32] as the camera locations provide the spatial information and the captured images provide the data for image-based action recognition. Duong et al. [31] had research on ADL recognition and abnormality detection using switching hidden semi-Markov Model on camera captured user position. Their work is based on the assumption that a typical ADL, e.g. having breakfast, usually consists of a sequenced transition of positions, e.g. from the door, to the fridge, to the stove, to the table, and to the basin. Figure 2.5 shows a few ADLs that take place in the kitchen and their position routines.

Support vector machine [32] and ontological approach [33] are also used in ADL recognition. Neural networks [34] especially deep learning [35] techniques are applied to ADL recognition over the years.

2.1.3 Activity Pattern Learning

ADL Pattern Learning is the next level beyond ADL recognition. To the best of our knowledge, there are only a few works in this area.

The first group of researchers [36] [37] uses a topic model approach. For activity pattern learning, daily routines are pulled from ADLs similar to text processing techniques

of ADL recognition, their works also reveal the relationship among different ADLs; in other words, behaviour pattern learning.

2.2 Causal Learning

Causal learning is the process of learning the cause-effect relationships or determining the causal status among events. Causal learning can be viewed as a bottom-up process whereby events that share contingencies become causally related, or a top-down process whereby cause-effect relationships may be inferred from observation and empirically tested for its accuracy. The most popular method used in causal learning is Bayesian causal inference [16] [39] [40]. As an example of the typical Bayesian causal inference, we assume that two weather types could be observed, namely wind (W) and thunder (T). Suppose shortly after observing wind or thunder, there is a high probability of raining (R). The likelihoods are the probabilities of wind on the condition of a rain $P(W|R)$ or no rain $P(W|\neg R)$, and the probabilities of thunder following a rain $P(T|R)$ or no rain $P(T|\neg R)$. Consider the priori probabilities of windy and thunderly weather, $P(W)$ and $P(T)$, respectively, are given, the probabilities of, given a rain, how likely windy weather was earlier (i.e., the wind was the cause) can be calculated using the Bayes rule -

$$P(W|R) = P(R|W) * P(W)/P(R) \quad (2.4)$$

Or similarly, given no rain later, how likely it was thunder weather earlier. In this scenario, the causal inference is in a closed domain. The likelihoods $P(W|R)$, $P(W|\neg R)$, $P(T|R)$, and $P(T|\neg R)$ are all determined in an earlier learning process, and the variables are all hand-picked. The Bayesian inference is to infer which of the two earlier weather is more likely to give the later observation. This likelihood could not be learned at the

beginning, and thus, they cannot be learned quickly. Besides Bayesian inference, researchers [41] proposed a framework to autonomously determine the pair of action and the state of the world it brings about. This thesis follows the probably approximately correct (PAC) semantics that the learned hypothesis is accompanied by predictive guarantees during the testing phase. Due to a low rejection rate, this approach causes the system to learn new causal hypothesis very carefully or, that is to say very slowly. In addition, this work also tries to learn the causal hypothesis of given pairs of consecutive events. However, other events may be observed among causal events which makes this assumption invalid.

More recently, deep neural networks are applied in this field [42] [43]. One work [42] explains the predictions of deep neural networks using causal semantics to relate the output prediction of a network to concepts represented within it. Another work [43] draws a connection between causal inferences and latent variable models. To wrap up this section, this review shall mention the concept of effective causal learning [15], which argues that two events may not be cause and effect to each other, but they can always take place in order in the application domain. As such, these events can be seen as having an effective causal relationship that they can be applied practically, regardless of what the "true" causes behind.

2.3 Adaptive Resonance Theory

2.3.1 Fusion ART

Adaptive resonance theory (ART) is a theory that explains how the brain processes information [44]. ART provides the foundation to a group of neural network models capable of supervised and unsupervised learning. Problems such as pattern recognition and prediction are of interest in ART models. In an ART model, object identification and

recognition generally occur as a result of the interaction of observer expectations with sensory information. ART model can be considered as 'bottom-up' and 'top-down' at the same time in the sense that the observer expectations are "top-down" while the sensory information can be considered as "bottom-up". ART models learn knowledge in the form of memory templates or prototypes that are then compared with the actual features of an object when detected by sensors. The result of this comparison gives rise to a measure of category belongingness. If this difference between sensation and expectation does not exceed a set of thresholds, indicated by the 'vigilance parameters', the input object will be taken as a member of the expected class. Models of ART and their supervised learning variations have been used for various pattern analysis and recognition tasks.

While the original ART models are predominantly used for unsupervised learning, more complex ART architectures are capable of performing other machine learning modalities [45]. Elementary ART models includes ART1 [46], ART2 [47], fuzzy ART [48], distributed ART [49], Gaussian ART [50], and other variations. Topological architectures [51], hierarchical architectures [52], and data fusion architectures [18] [53] [54] are proposed based on elementary ART models. Within the family of data fusion ART architectures, there is a group of networks, known as fusion ART [18], formerly the multichannel adaptive resonance associative maps (multichannel ARAM) [55], that learns cognitive codes associating multimodal patterns across multiple input channels. Self-organizing is an important feature of fusion ART. When no learned node is matched, the network will use the uncommitted node to represent the new pattern.

In a generic multichannel fusion ART model, the category field is fully connected to all the input fields (see Figure 2.6). The dynamics of fusion ART are summarized as follows:

Input fields: Let F_1^k denote the input field that holds the input patterns of channel k .

Figure 2.6: The generic fusion ART architecture

Input vectors: Let $\mathbf{I}^k = (I_1^k, \bar{I}_1^k, \dots, I_n^k, \bar{I}_n^k)$ denote the input vector of channel k for $k = 1, \dots, n$, where $I_i^k \in [0, 1]$ are the input signal values and $\bar{I}_i^k = 1 - I_i^k$. Complement coding serves to normalize the magnitude of the input vectors and has been found effective in fuzzy ART systems when it comes to prevent the problem of code proliferation [48].

Category field: Let F_i , where $i > 1$, indicate the category field. In the standard multichannel ART, there is only one category field F_2 .

Activity vectors: Let $\mathbf{x}^k$ denote the activity vector for input field F_1^k , and $\mathbf{y} = (y_1, y_2, \dots, y_m)$ denote the activity vector of F_2 . Initially, $\mathbf{x}^k = \mathbf{I}^k$ for $k = 1, 2, \dots, n$.

Weight vectors: Let $\mathbf{w}_j^k$ denote the weight vector associated with the j th node in F_2 for learning the input patterns in F_1^k . Initially, F_2 contains only one uncommitted node with the weight vectors containing all 1s.

Parameters: Each field's dynamic is determined by the choice parameters $\alpha^k \geq 0$, learning rate parameters $\beta^k \in [0, 1]$, contribution parameters $\gamma^k \in [0, 1]$, and vigilance parameters $\rho^k \in [0, 1]$.

Code activation: Given the activity vectors $x^1, x^2, \dots, x^k$, for each F_2 node j , the choice function T_j is as follows:

$$T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|}, \quad (2.5)$$

where the fuzzy AND operator $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the norm $|\cdot|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code competition: The F_2 node with the highest choice function value is identified by the code competition process. The winner is indexed at J , where $T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}$. When a category choice is made at node J , $y_J = 1$ and $y_j = 0, \forall j \neq J$. This indicates a winner-take-all strategy.

Template matching: Upon code competition, the template-matching process takes place to check if resonance is occurring. For each channel k , the match function is given as follows:

$$m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k. \quad (2.6)$$

The template-matching value of the chosen node J is checked to see whether it meets the vigilance criterion. If any of the vigilance constraints is violated, a mismatch reset occurs by setting the choice function T_J to 0 for the duration of the input presentation. The search process will keep selecting other F_2 nodes until resonance occurs. If the uncommitted node in F_2 is identified as the winner, after learning it becomes committed, a new uncommitted node is created and added to F_2 .

Template learning: Once a node J is selected for learning, in each channel k , the weight vector is updated by the learning rule shown in the following:

$$\mathbf{w}_J^{k(new)} = (1 - \beta^k) \mathbf{w}_J^{k(old)} + \beta^k (\mathbf{x}^k \wedge \mathbf{w}_J^{k(old)}). \quad (2.7)$$

Figure 2.7: EM-ART architecture (adapted from [17]).

2.3.2 EM-ART

The Episodic Memory ART (EM-ART) [17] is an extension of fusion ART to model episodic memory. The proposed architecture of EM-ART is shown in Figure 2.7.

As EM-ART is an episodic memory inspired variation of fusion ART, it could learn episodic sequences of sensory input and encode the sequences into nodes in the higher layer. Experiments [17] show that EM-ART is able to achieve a high level of memory performance with good robustness while managing memory usage over time.

There are three layers F_1 , F_2 , and F_3 in EM-ART model. As a result, there are two main mechanisms linking them: $F_1 \rightarrow F_2$ and $F_2 \rightarrow F_3$.

The input data structure for the EM-ART is a fixed-length sequence of sensory pattern input: Let $\mathbf{E} = (\mathbf{S}^1, \mathbf{S}^2, \dots, \mathbf{S}^t)$ denote an episodic input that consists of a sequence of t instant states $\mathbf{S}$. Each state $\mathbf{S}$ consists of the sensory input vectors from all the k channels such that $\mathbf{S} = (I_1^k, I_2^k, \dots, I_n^k)$, where $I_i^k \in [0, 1]$ indicates the input i to channel k .

The operation for $F_1 \rightarrow F_2$ is similar to the standard fusion ART operation. It goes through code activation, code competition, template matching, and optionally template learning. For each input pattern in F_1 layer, either an existing code in F_2 layer is

activated, or new code is created to represent the new input pattern. The operation is shown in Figure 2.8.

Instead of using learned pattern codes as input, the EM-ART architecture uses the sequence of code activations in F_2 layer as the input for the second level of process $F_2 \rightarrow F_3$. To encode the sequence of activations in a single episodic input vector $\mathbf{E}$, the EM-ART firstly defines the time length t of each episodic, so that each episodic input vector $\mathbf{E}$ consists of t sequenced input states S^t . Each time a code in the F_2 layer is activated by a F_1 layer input, the system decreases the activity of all activated F_2 nodes, while the newly activated node has a value of 1. The $F_2 \rightarrow F_3$ mechanism goes through the standard fusion ART processes similar to that of $F_1 \rightarrow F_2$. This mechanism is shown in Figure 2.9.

In the original EM-ART [17], the activation value is decreased exponentially as $y_j^{(new)} = y_j^{(old)} (1 - \tau)$, where y_j is the activation value of the j th node in F_2 and $\tau \in (1, 0)$ is the decaying factor. In other words, it forgets rapidly at the beginning and slows down later on. This pattern of activation decay is similar to human forgetting pattern. One good feature of this decay algorithm is that when the system performs a code read-out operation, the sequence of states in F_2 could be easily retrieved by just looking at the order of the F_2 nodes' activation values. The F_2 node with the highest activation

Figure 2.8: EM-ART $F_1 \rightarrow F_2$ Operation

value therefore represents the most recent event in the episode. Due to the exponential decay feature, the decay is rapid at the beginning. As a result, the more recent code activations have a higher weighting in pattern comparison. For many applications, this feature could be valid as it is consistent to human-like forgetting pattern. However, it may not suit some other applications. Besides, the activation sequence vector cannot represent multiple occurrences of the same event in an episode. This issue is due to its encoding process, wherein the newest occurrence of one state will overwrite its previous occurrence record. There are several ways to work around this. The first possible way is to use another input channel to separate the possible repeated states. The other possible way is to create another code in F_2 layer dedicated to representing the second occurrence of a state.

Figure 2.9: EM-ART $F_2 \rightarrow F_3$ Operation

Chapter 3

Multi-Memory Model of Daily ADL Routine

This chapter introduces an ART-based multi-memory neural network model, ADLART, capable of learning the daily ADL routines. Experiments are conducted to analyze the learning patterns ADLART generated from test data. In the later part of this chapter, a recommendation model, RADLART, is further developed. Experiments show RADLART could provide meaningful recommendations to elderly people in their life.

3.1 Introduction

As mentioned in Chapter 1, this thesis starts from looking at the modeling of activity routines in memory forms inspired by the human biological memory system. The following design requirements are identified for the memory system to achieve.

First, the two basic elements in daily ADL routines, namely the individual ADLs and their sequences, are identified. To achieve efficient encoding of ADLs and their sequences, the data structure should firstly serve the function of distinguishing them from each other. The data structure should be able to identify key ADL differences between two ADL sequences. For example, the user had three meals in one day, but had no food in another day. Besides, the data structure should be able to give certain tolerance to

difference in different ADL sequences in order to classify them into clusters. For example, it should not be a major difference between a day having lunch at 12:10pm and another day at 12:15pm. The basic challenges in data structure design is to distinguish ADLs and their sequence, having key ADL difference identified while tolerating certain level of variations in ADL orders and happening time.

Second, in the daily routine retrieval, some key tasks to be achieved are identified. First task is ADL sequence formation that the system should be able to construct ADL sequence data structures from a series of ADL input with time-stamp. Second task is ADL sequence recognition that a stored sequence of ADL could be able to be identified in response to a incoming ADL sequence data. The recognition mechanism is preferred to be able to tolerate partial data with auto recall of missing parts. Lastly, the model should be able to retrieve all individual ADLs from recognized sequences.

Third, people have variations in ADL timing every day, but they usually have clear routines in different type of days. ADLs in working day should be more or less the same to each other, but very different from those in vacation days. The preferred model should be able to gradually learn semantic patterns of user's typical days, like holiday, working days, Fridays, and sick days. In a case that an input ADL sequence matches a different type of day, it could be a signal for a abnormal situation to be noticed. For example, if the people is sick in a working day, he may wake up and dress up slower than his normal working days.

3.2 ADLART Model

3.2.1 The ADLART Model Architecture

Given the design requirement mentioned in the previous section, a multi-memory model, ADLART (Activities of Daily Living ART), is proposed. ADLART is inspired from

human memory system, especially the episodic memory model, EM-ART, mentioned in the Chapter 2. As shown in Figure 3.1, the ADLART model contains a working memory component with two input fields, e.g. day information and ADL sequences. The working memory communicates with the episodic memory component to learn the episodes of ADL sequences. At the same time, the working memory also works with a semantic memory component for semantic extractions.

Inside the episodic memory sub-model (Figure 3.2), there are two input channels contributing to the process. The ADL sequence input field has activation vectors of ADL sequences of a day.

Figure 3.1: The ADLART architecture

$$\mathbf{S}^{\text{ADL}} = (I_1^{\text{ADL}}, I_2^{\text{ADL}}, \dots, I_n^{\text{ADL}}). \quad (3.1)$$

Figure 3.2: The episodic sub-model

where n is the total number of ADL categories and the value of $I_i^{ADL} \in [0, 1]$ represents the order of ADL sequence that a ADL with a smaller I_i^{ADL} value happens earlier than ones with a bigger value. The detailed algorithms to calculate I_i^{ADL} will be discussed later. By the way, the ADL recognition and activation process is beyond the scope of this architecture and it may be realized by using various ADL recognition algorithms, e.g. [31] [32].

At the same time, we have the day input field having vectors $\mathbf{S}^{\text{Day}}$ indicating the day information as tags.

$$\mathbf{S}^{\text{Day}} = (I_1^{\text{Day}}, I_2^{\text{Day}}, \dots, I_{11}^{\text{Day}}). \quad (3.2)$$

In this research, 11 types of days are defined and listed in Table 3.1.

Index	Meaning
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday
8	Public Holiday
9	Sick Day
10	Vacation
11	Abnormal Days

Table 3.1: Day Information Input

$\mathbf{S}^{\text{Day}}$ could include multiple day information. For example, a holiday is on Monday, and it just so happens that the user is sick on that day. In this case, the I_1^{Day} , I_8^{Day} , and I_9^{Day} values are 1s while others are 0s.

The output is an ADL day category representation $\mathbf{A}$ which stores instances of categories of $\mathbf{S}^{\text{ADL}}$ with tags of the day information. We could make many applications based on $\mathbf{A}$, or use it as an input for higher level analysis.

3.2.2 Episodic Memory Learning

The mechanics of the ADL sequence coding and routine learning is summarized in Algorithm 1 below.

Algorithm 1 ADL Sequence Coding and Routine Learning

- 1: **for** each ADL J in ADL sequence $\mathbf{A}$ **do**
 - 2: let node activation $I_J^{\text{ADL}} \leftarrow \frac{\text{ADL occurrence time}}{\text{Whole day time}}$
 - 3: **end for**
 - 4: **for** each ADL sequence vector $\mathbf{y}$ containing $\mathbf{S}^{\text{ADL}}$ and its corresponding Day Information vector $\mathbf{S}^{\text{Day}}$ **do**
 - 5: select a resonance node J' in L_4 based on sequence vector $\mathbf{y}$
 - 6: learn its associated weight vector as $\mathbf{w}_J'^{(new)} = \mathbf{y}$ if $\mathbf{A}$ is a novel ADL Sequence
 - 7: **end for**
-

To encode ADL action time information into the input data structure, we propose to use time of day as the activation strength of $I_i^{\text{ADL}} \in [0, 1]$. The time of day is calculated as $\frac{\text{ADL occurrence time}}{\text{Whole day time}}$. It represents time in minutes. A whole day has 1440 minutes. For example, if a person dresses up at 8:00am, the activation strength value for dressing up will be $8*60/1440 = 0.333$; if he has lunch at 12:25pm, the activation value for lunch will be $(12*60+25)/1440 = 0.517$. By doing it this way, the time of day is encoded into the ADL sequences. Small variations in ADL occurrence time will be reflected as a small difference in activation value, which will be tolerated in the template matching algorithm mentioned in the previous section. On the other hand, by using the activation strength assignment, it is very easy to retrieve the time of day and the sequence order for each individual ADLs. At the end of the day, the most recent ADLs will have a higher activation value, and the earliest ADLs in the morning will have smaller activation value.

This is consistent with the episodic memory forgetting trends. However, there is one issue in this setting that the ADLs happening around the before and after midnight will have very different values. The ADLs before midnight will have activation values near 1 while ones just after midnight will have activation values near 0. We will propose methods to compensate for this in future work.

3.2.3 Semantics Memory Learning

A good feature of ADLART model is that it could learn the semantics of the day models. As we propose two input channels for each daily routine category, the day information input could be treated as a tag for the routine models stored inside the routine category F_2 . If the day input vector is not fuzzy, there are few ways to implement semantic daily routine inside the routine category layer. The first and simplest way is to store strictly one category for each day type inside the F_2 layer. To achieve this, all we need to do is to prevent new category creation in F_2 when no node meets the vigilance. This is not a good way as it will limit the categories in F_2 , and hinder the ability of category learning. The second way is to tolerate different routine categories created for each day type, and use an average value (or weighted average for fuzzy days) of all these categories to represent the semantic day type. In addition, each time a category is fired, its ADL sequence and occurrence time could be updated with the new entry. We use a third way that creates a dedicated semantic component to store the semantic information.

$$\mathbf{S}^{Sem} = (A_1, A_2, \dots, A_{11}). \quad (3.3)$$

where A_n is the semantic routine (ADL sequence) stored in F_2 for day type n .

Each time when a routine category is activated, the semantic memory is generalized by a fraction of the new input. Mathematically, this is achieved by updating ADL sequence value for the same day type by

$$A_n^{new} = (1 - \beta_n^{Sem})A_n^{old} + \beta_n^{Sem}(\mathbf{x}^{ADL} \wedge A_n^{old}) \quad (3.4)$$

where the $\beta_n^{Sem} \in [0, 1]$ is the semantics update parameter for day type n .

The algorithm for semantics learning is summarised in Algorithm 2. For each input pattern, every category node in F_2 is activated by the ART choice function. After that, ADLART searches from the highest activation node to lower ones, testing which node fulfil the match function. If resonance occurs, it means one category matches the input pattern, this category is updated to generalize to the input pattern, and the semantics memory is also updated. However, if no resonance occurs, a new node will be created in F_2 to encode the input pattern.

Algorithm 2 Daily Routine activation and Semantics Learning

```

1: for each input pattern of ADL sequence as vector  $(\mathbf{S}^{ADL}, \mathbf{S}^{Day})$  in  $F_1$  do
2:   Activate every category  $j$  in  $F_2$  by choice function  $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|}$ 
3:   select category  $J$  such that  $T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}$ 
4:   set node activation  $y_J \leftarrow 1$ 
5:   while match function  $m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k$  (not in resonance)
6:     or  $J$  was selected previously do
7:       deselect and reset  $J$  by  $T_J \leftarrow 0, y_J \leftarrow 0$ 
8:       select another node  $J$  with  $T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}$ 
9:   end while
10:  if resonance occurs(Routine recognized) then
11:    Update episodic category(template learning)  $\mathbf{w}_J^{k(new)} = (1 - \beta^k)\mathbf{w}_J^{k(old)} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(old)})$ 
12:    Update semantic memory by  $A_n^{new} = (1 - \beta_n^{Sem})A_n^{old} + \beta_n^{Sem}(\mathbf{S}^{ADL} \wedge A_n^{old})$ 
13:  else(No matching/resonance)
14:    let  $J \leftarrow J^0$ , where  $J^0$  is a newly recruited uncommitted nodes in  $F_2$ 
15:    Learn  $(\mathbf{S}^{ADL}, \mathbf{S}^{Day})$  as a novel event with  $\mathbf{w}_J^{k(new)} = \mathbf{S}^k$ 
16:  end if
17: end for

```

The proposed ADLART model fulfils the design challenges described in the previous section that the individual ADL and ADL sequences are represented as single data entry

and data vectors. They have distinguished definitions from each other. The activation strength design makes sure the model is able to tolerate a certain level of data variations inside the same cluster, while the choice parameter α^k enables the ADLART to differentiate patterns. In F_1^{ADL} , the ADL sequences are formed. In the F_2 field, routine recognition takes place which compares the incoming entry with the stored categories. Also, it could recall the sequence of recognized ADLs. Last but not least, it maintains a semantic routine memory in the F_2 field as well.

3.2.4 Evaluation

To test the performance of ADLART, a simulation environment is designed and constructed. We first interviewed five people for their daily routine on different types of days. Based on their ADL time ranges, a programme is written to automatically generate data entries with some random variations. For example, a person has a standard morning routine, no social activities, and has an entertainment ADL in the evening for working days. At the same time, he has a delayed morning routine with both social and entertainment ADLs possibly ranging for the whole day in weekend and public holidays.

For the day category, as mentioned in the previous section, we have eleven types of days, including Monday to Friday, weekends, sick days, public holidays, and vacations. For the ADL category, in the experiment, we use a simplified version which consists of ten types of ADLs listed in Table 3.2.

In the first part of the experiment, we test the influence of different vigilance parameter values in creating new category codes for a given type of day. We use Monday as an example, of which the activities are listed in Table 3.3.

As we are looking at the ADL patterns, the day category is used as tags by setting the vigilance parameter ρ^{Date} to 0. We use the simulation programme to generate 100 samples

ID	ADL
1	Wake Up
2	Wash Up
3	Dress Up
4	Breakfast
5	Entertainment
6	Lunch
7	Social
8	Dinner
9	Housework
10	Sleep

Table 3.2: ADLs used in experiment

ADL	Time
Wake up	6:00am - 8:00am
Wash up	wake up + 15m $\pm$ 5m
Dress up	wake up + 30m $\pm$ 10m
Breakfast	8:00am-9:00am
Entertainment	9:30am-10:30am
Lunch	11:30am-13:30pm
Social	no
Dinner	17:00pm - 19:00pm
Housework	14:00pm - 20:00pm
Sleep	21:00pm-23:00pm

Table 3.3: Simulation Statistics: Monday

of Monday routine according to the variance mentioned in Table 3.3. The relationship between the vigilance parameter ρ^{ADL} and the number of categories created are plotted in figure 3.3. We could see that when ρ^{ADL} is smaller than 0.95 there are only one or two categories created. This is not a good setting as one category could not represent the variation and distribution of the day type. When ρ^{ADL} increases towards 1, the number of categories created accelerates towards the number of samples. When ρ^{ADL} is set to 0.98 to 0.99, there are about 10 categories created. We prefer this number of categories as they could reasonably represent the variations of the day type, and at the same time

Figure 3.3: Vigilance parameter vs. categories created in 100 samples

avoid overfitting. When ρ^{ADL} increases over 0.99, there are more than 40 categories created, in other words, a higher risk of overfitting.

In the next experiment, five types of days are used to show the performance of the ADLART model: Monday, Tuesday, sick day, vacation, and Sunday. We specially designed their relationships as follows: The Monday and Tuesday routines are similar to each other that they are basically weekdays. They have starting time differences in ADLs, but they have heavy overlap with each other. The Sunday, vacation, and sick Day are quite similar to each other that they have a delayed morning routine, vacation and sick day do not have breakfast, social, and housework. Besides, in sick days, the person sometimes does not dress up (stay on the bed). In the training phase, we use a simulation generator to generate 100 samples for each type of days. The vigilance parameter ρ^{ADL} is set at 0.99. The training sequence is Monday, Tuesday, Sunday, vacation, and sick days. After training, we use the same simulation generator generates another 100 samples for each type of days. We use the correctly recognized days over the 100 samples as the accuracy (recall). We run 10 times the experiment. The average result is summarized in Table 3.4. The result is as expected. Monday and Tuesday generators use similar sample distribution to generate data, and the model learns Monday first. As a result, Monday

has a higher accuracy, whereas the model sometimes mistakes Tuesday data for Monday. The Sunday, Sick day, and vacation are well distinguished from other days.

ADL	Recall Accuracy
Monday	98.4%
Tuesday	66.4%
Sunday	100%
Sick Day	98%
Vacation	100%

Table 3.4: Accuracy in combined experiments

After learning a semantic memory of the daily routine in ADLART, in this experiment, we generated 200 samples of Monday routine and 200 samples of sick day routine. After training the system, the content of the semantic memory is listed in Table 3.5. The results satisfy the design requirement that over the time the system has learnt semantic pattern of a typical day of the user, e.g. compare Table 3.3 and Table 3.5. At the same time, the most recent user activities contribute more on the semantics pattern, which is very helpful for user health trend monitoring.

ADL	Monday	sick day
Wake Up	7:14	11:52
Wash Up	7:29	12:51
Dress Up	7:46	
Breakfast	8:38	
Entertainment	10:4	
Lunch	12:45	12:52
Housework	17:49	
Dinner	18:07	18:20
Sleep	22:18	23:13

Table 3.5: Semantic memory learning

3.3 Recommendation ADLART: RADLART

3.3.1 The RADLART Model

The experiments in the previous section show the ADLART is capable of learning ADL routines in both semantic memory form as well as episodic memory form. It is natural to extend the ADLART model for applications to help users. In this section, a recommendation model, Recommendation agent using ADLART (RADLART), is introduced. RADLART extends ADLART episodic sub-model with a profile field and a wellness score field. The RADLART structure is shown in Figure 3.4. The ADLART's ADL sequence field is used to model the everyday ADL sequences. A wellness score field is added to handle the score assessment of ADLs; Moreover, the profile field is used to group elderly with similar profiles. This is necessary as elderly people generally prefer recommended ADL plans that come from similar peer's daily routine. The details of each field is discussed in the following subsections.

Figure 3.4: The RADLART structure

3.3.1.1 ADL Sequence Field

Similar to ADLART, the ADL sequence field of RADLART represents the ADL sequences of a day using an activation vector:

$$\mathbf{S}^A = (I_1^A, I_2^A, \dots, I_n^A). \quad (3.5)$$

where the value of $I_i^A \in [0, 1]$ indicates the order of ADL sequence within the day using $\frac{ADL\ occurrence\ time}{Whole\ day\ time}$, and n is the total number of ADL categories.

We identify a set of 30 ADLs organized into five board categories, namely rest and personal hygiene, meals, housework and exercises, learning, entertainment and social activities. As the activation vector S^A does not contain activity duration information, some ADL has two versions in our set of 30 ADLs, for example, "short Catnap" (catnap within 1 hr) and "long Catnap" (catnap more than 1 hr). Moreover, in our wellness score system, we give the same score for the same activity occurred in any time within a day which is not valid for some ADLs. To handle this, we treat some ADL as two different events if they happen in a different time, for example, having a meal after evening could be either "Dinner" (before 9:00 pm) or "Supper" (after 9:00 pm). And finally, for physical exercise, we use three ADLs to represent different levels of exercise. The full list of ADLs we used are:

Rest and personal hygiene category: {"Wake Up", "Wake Up Late", "Personal Hygiene", "Toilet", "Grooming", "Morning Shower", "Evening Shower", "Short Catnap", "Long Catnap"}.

Meal category: {"Breakfast", "Lunch", "Brunch", "Afternoon Tea", "Dinner", "Supper" }. As mentioned above, the classification of ADLs for meals is sensitive to the occurrence time.

Entertainment and social activities category: {"Outdoor Social", "Indoor Social", "Watch TV", "Board Game", "Video Game", "short use of computer", "Long use of computer" }. The "On computer " refers many activities on a computer including surfing the internet, reading articles, managing family photos, and etc. but except playing a computer game which is classified as "Video Game".

Housework and exercise category: {"Light Housework", "Heavy Housework", "Light Exercise", "Exercise", "Heavy Exercise" }. As mentioned above, we use three ADLs to represent different exercise levels.

Learning category: {"Reading", "Writing", "Working"}.

Inherent from the ADLART episodic sub-model, there are some limitations in this ADL representation. In this model, ADLs are represented with starting time only, and the duration information is not captured. Similarly, this model cannot handle interleave ADLs and multiple occurrences of ADLs.

3.3.1.2 Profile Field

The profile field is incorporated in view that the ADL plan recommended from people with similar profiles will be more acceptable. The profile field contains vector $\mathbf{S}^P$ indicating the age, gender, and fitness level of a person.

$$\mathbf{S}^P = (a, g, s). \quad (3.6)$$

To encode the age of elderly people, we use the formula $a = (Age - 50)/50$. By using this formula, we make the assumption that elderly people are 50 to 100 years old.

We use a boolean variable g to represent gender that male is encoded as 1 while a female is encoded as 0.

For fitness level, s has five discrete levels: 0.2, 0.4, 0.6, 0.8, and 1 (0 being least capable and 1 being most capable) to represent different strength levels.

3.3.1.3 Wellness Score Field

There are various ways to compute wellness score from a sequence of activities. In our approach, there is a default 50 score. Each activity brings a change of score in four wellness dimensions. The vector $\mathbf{S}^W$ indicates the change of wellness score in dimensions.

$$\mathbf{S}^{\mathbf{W}} = (I^P, I^C, I^E, I^S). \quad (3.7)$$

where I^P, I^C, I^E , and I^S are the wellness scores in the physical, cognitive, emotion, and social quotients respectively.

Table 3.6 lists all the ADLs that are used in all our experiments and their effects on the different quotients. Because the list of ADLs is not exhaustive in real life and their effects to the different quotients have not been analyzed quantitatively in the literature, Table 3.6 only represents our interpretation on how the listed ADLs (introduced in Section 3.3.1.1) can be assessed as the assignments of wellness scores. Although all the scores are assigned manually, we do generally follow the respective expert knowledge in different fields. Specifically, the physical scores are determined based on how ADLs affect the physical activeness of the elderly. For example, different intensity levels of exercise directly affect their physical well-being [56]. Moreover, housework is considered as a healthy activity for the elderly [56], but all activities associated with a relatively long time of stationary motion during day time (such as lying, sitting, and standing) affect the physical aspect negatively. The cognitive capability of the elderly can be generally assessed by whether they can remember the necessary information correctly and whether they can perform the ADL successfully. Therefore, we assign positive scores (relatively small numbers due to high frequency of occurrences) to the performed ADLs. It is hard to determine how most ADLs affect the emotional well-being of the elderly, unless the insights or the outcomes of the events are known. Among all the selected ADLs in the emotional quotient, we only assign positive scores to wake up in the morning according to [57] and shower in the morning (because on common sense, it is refreshing). Among all the listed ADLs, the two obvious social events are assigned with high positive scores

ADL	Physical	Cognitive	Emotional	Social
Wake Up 1	0	0	+10	0
Wake Up 2	0	-5	+5	0
Hygiene	+1	+5	+2	0
Toilet	0	+5	+2	0
Grooming	0	+5	+2	0
Morning Shower	+2	+5	+3	0
Evening Shower	+2	+5	+3	0
Short Catnap	-10	0	+5	0
Long Catnap 2	-20	0	+10	0
Breakfast	+2	+5	+1	0
Lunch	0	+5	+1	0
Brunch	0	+3	+1	0
Afternoon Tea	0	+3	+1	0
Dinner	0	+5	+1	0
Supper	-2	+3	+2	0
Watching TV	-3	+2	0	-2
Board Game	-3	+2	0	+10
Video Game	-3	+2	0	0
Short use of computer	-5	+2	+2	+2
Long use of computer	-10	+4	+5	+5
Outside Social	+5	0	+2	+15
Indoor Social	+2	0	+4	+15
Light Housework	+10	+3	-5	-2
Heavy Housework	+20	+5	-10	-4
Light Exercise	+10	0	+2	+2
Exercise	+20	0	-2	+4
Heavy Exercise	+30	0	-6	+2
Reading	-3	+5	+5	+3
Writing	-3	+10	+5	+5
Working	-5	+10	-10	+10

Table 3.6: Wellness scores in four quotients for ADLs

in the social quotient. "Board Game" is also assigned with a positive score, because in our definition, it refers to the traditional board game played head to head.

3.3.2 Recommendation

Using the three data fields, RADLART could learn the elderly's ADL sequence patterns with the profile and wellness scores from the training data. For each elderly, the agent keeps a record of his ADL routines. In a new day, the agent keeps searching its category knowledge, trying to select a number of daily ADL sequences who matches the recorded ADL sequence. In the matching process, we only look at ADLs happened up to the current time and scheduled ADLs in RADLART categories. Therefore, we name this process as partial matching. Selected ADL sequence categories will be ranked using wellness scores and the best candidate will be used for recommendation. The overall mechanism is shown in Table 3.7.

Phase	Details
Training Phase	System training
Sequence Selection	Select n ADL sequence categories that have beginning part closest to the input ADL sequence $\mathbf{S}^A$
Recommendation	Among the selected n ADL sequence categories, recommend the ADL sequence with maximum wellness ranking function $F(\mathbf{S}_i^W)$ value.

Table 3.7: The major phases of the recommendation agent

3.3.2.1 ADL sequence learning

In the first step of the recommendation process, the agent learns from training data and builds up the ADL daily routine categories. RADLART first classifies the input training samples into existing categories. If successful, the parameters of the existing category are

updated according to the new sample. If no existing category is identified, RADLART creates a new category for the input sample.

3.3.2.2 ADL sequence selection

ADLART can be activated at any time of the day. Given a set of ADLs that the elderly has performed, RADLART recommends a sequence of ADLs for the remaining part of the day with the objective of maximizing the overall daily wellness score. An intuitive way is to select and recommend single ADLs that have the highest wellness values. However, for a holistic recommendation, consideration must be made for the recommended ADLs to be based on the ADLs performed so far in the day. Based on the elderly's ADLs sequence that is already done earlier in the day, RADLART searches the knowledge base and selects n categories that, at this point of a day, have ADLs sequences closely similar to the ADL sequence already done. Subsequently, RADLART chooses one of the selected n categories for recommendations.

As summarized in Algorithm 3, the matching of partial ADL sequence is one of the core algorithms used in the ADL sequence field. Instead of choosing one node passing the match function with the highest activation value, RADLART selects n nodes as recommendation candidates. A masking technique is used to set the weight of irrelevant ADLs to 0 before matching. This technique will make RADLART focus on matching the given partial sequence. One additional feature of RADLART enabled by partial matching is the ability to take user scheduled future activities into recommendations. The activation value of the scheduled activities will be set to its scheduled time, and the RADLART system will recommend ADL plans matching these scheduled future activities as well.

Algorithm 3 Sequence Selection using Partial Matching

```

1: Initialize counter  $c = 0$ 
2: Calculate event activation value for current time  $w^t$ 
3: Masking every category  $j$  in  $F_2$  by  $w_j^k \leftarrow 0$  if  $w_j^k > w^t$ 
4: Activate every category  $j$  in  $F_2$  by choice function  $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + \mathbf{w}_j^k}$  according to
   input vector  $(\mathbf{S}^P, \mathbf{S}^A)$ 
5: while Counter  $c < n$  do
6:   select category  $J$  such that  $T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}$ 
7:   set node activation  $y_J \leftarrow 1$ 
8:   while match function  $m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} < \rho^k$ . (not in resonance) or  $J$  was selected
     previously do
9:     deselect and reset  $J$  by  $T_J \leftarrow 0, y_J \leftarrow 0$ 
10:    select another node  $J$  with  $T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}$ 
11:  end while
12:  if resonance occurs(Routine recognized) then
13:    Put  $J$  into selected category set
14:    Update counter by  $c++$ 
15:  end if
16: end while

```

3.3.2.3 Score ranking and recommendation

Most existing wellness assessment systems developed by the experts are based on the interview analyzes of surveys or questionnaires periodically [56]. However, to initiate more frequent interactions and to provide more real-time recommendations, RADLART is able to assess the elder user's wellness score at any time during the day based on the recognized activities [58].

The wellness scores in each individual quotient can be assigned according to the respective ADLs recognized or self-reported. The activity recommendation module uses the aggregated score to suggest a healthier or more active ADL sequence to the user. After selected n nearest categories inside the F_2 layer, the next step is to choose one from these n categories according to a ranking function, $F(n)$.

Note that for all-rounded well being, it is important to have a balanced set of scores

across all the four wellness quotients. In other words, a very low score in any of the quotients is not desirable for wellness recommendation. Compared with arithmetic mean, geometric mean favours balanced set of scores and penalize very low value in any of the four quotients. For this reason, in RADLART, we use the geometric mean formula:

$$\mathbf{F}(\mathbf{n}) = (I^P * I^E * I^S * I^C)^{\frac{1}{4}}. \quad (3.8)$$

to compute the ranking function.

3.3.3 Evaluation

3.3.3.1 Methodology

To evaluate the performance of RADLART, various simulations are designed and conducted. In the first set of experiments, the RADLART recommendations are compared with the users' typical daily routines. The second set of experiments simulates a more realistic case by taking the human factor into consideration that we assume users take recommendations with different acceptance rates and they will only follow the recommendation for a predefined period of time.

We construct a simulator to simulate the daily lives of different types of the elderly. The simulator first takes inputs from a set of elderly profiles which represent persons of different ages, genders, and body fitness levels. In the second step, the simulator generates daily routine templates based on the personal profiles with a certain level of randomness. These routine templates represent relatively stable ADL sequences on elderly's everyday life. Finally, daily ADL sequence instances are generated from the routine template. By following this three-step process, the simulator could generate numerous samples trying to replicate the real situations with a certain level of randomness.

3.3.3.2 RADLART Recommendation

In this set of experiments, we evaluate how well the RADLART agent recommends ADL plans by comparing against the baseline, which refers to the users' typical daily routine. Moreover, we want to see the effect of the recommendations at different times in a day. The inputs of the experiments are a set of simulated data samples of ten users and a typical daily routine of one selected user. In the experiment, at a particular point of day, the agent gives recommendation to the selected user, and the user follows all ADLs according to the recommended ADL plan. We then analyze whether there is any improvement in the wellness score.

In the training phase, we input ten profiles into the simulator to generate 100 daily routine templates and then generate 1000 ADL sequences from these routines. In the testing phase, we input a person's profile and a daily routine template spanning 10 ADL sequence samples. For each sample, we run the recommendation agent with four different settings: 1) no recommendation is taken, 2) taken recommendation (following all ADLs in the plan) at 11:00 am, 3) taken the recommendation at 3:00 pm, and 4) taken the recommendation at 6:00 pm. The results are shown in Fig. 3.5.

Generally speaking, the recommended ADL plan is expected to improve the user's overall wellness scores. In addition, we could observe in Fig. 3.5 that if the user takes the recommendation earlier in the day, RADLART gives recommendations that have more balanced wellness scores in the four quotients. This may suggest that the geometric mean approach of computing the overall wellness score (see Eq. (3.8)) provides a balanced recommendation across all the quotients.

3.3.3.3 Experiments with Different Acceptance Rates

In the second set of experiments, more realistic scenarios are simulated wherein the elderly may not always follow the RADLART recommendations. In other words, the

Figure 3.5: Wellness scores improvement by following the recommendations of the RADLART agent in different starting times.

elderly accept recommendations according to different acceptance rates. Moreover, if they have already taken some recommendations, they will switch back to their typical routines after some time. In that case, the agent will continue giving recommendations which hopefully will be accepted according to the acceptance rate.

In this experiment, we assume the elderly follow recommendations at different acceptance rates, namely 0% (baseline), 25%, 50%, 75%, and 100%. Moreover, the elderly will always follow the given recommendation for two consecutive hours. After which, the RADLART agent will provide recommendation again. For example, at 8:00 am, the agent gives the following ADL plan: having breakfast at 8:20 am, going light exercise at 8:50 am, and having lunch at 11:20 am. If the elder user accepts the recommendation, his or her routine will be having breakfast at 8:20 am and doing light exercise at 8:50 am as suggested. However he or she will switch back to the typical routine say “housework” at 11:00 am, and he or she will consider the lunch recommendation again at 11:20 am and make a decision on the next ADL according to the acceptance rate.

We conducted the experiments in our simulation environment using different acceptance rates. The result shows that in the designed experiment scenarios, the system helps

to improve the elderly’s wellness scores. The relationship between the average wellness scores and the different acceptance rates is given in Fig. 3.6. It is shown that users with higher acceptance rates benefit more from the RADLART recommendation agent.

Figure 3.6: Wellness scores improvement by following the recommendations of RADLART under different acceptance rates.

3.3.3.4 Experiments on User Preference

One important feature of RADLART is that all the recommendations are generated from the learnt knowledge base of the target population. More specifically, the profile field in RADLART ensures the recommended activity source shares a similar profile with the current user. This is expected to increase the ability of the user to perform the recommended activity sequence. Psychology literature shows that people are easily influenced by their preferred peers [59]. To evaluate how people might accept the recommendations provided by our RADLART agent, we evaluate its acceptance rate by people.

The experiment consists of two parts. In the first part, we interview subjects for their profiles and typical daily routines. Based on this information, we train the RADLART agent. In the second part, we ask the same subjects about their routines in a particular morning. Then we show them three choices of activity recommendations with the corresponding supporting reasons highlighted for the afternoon: 1) the activities from

the typical daily routine model, 2) the activities with the highest wellness scores, and 3) the activities recommended by RADLART (from people with similar profile). For each recommendation choice, we ask the subjects their preferences of the recommendations with the scale from 1 to 5 (where 5 means strongly willing to choose, and 1 means not interested at all).

We conducted the experiment on eight elder Singaporeans. The result is listed in the Table 3.3.3.4 that the RADLART recommended ADL plans receive higher preferences than the ADLs with the highest wellness scores, and is close to the preference level of people’s typical daily routines.

ADL sequence	preference mean	SD
Typical Routine	4.3	0.5
RADLART	3.9	0.4
Highest Score ADL	2.8	0.7

Table 3.8: Preferences by real elderly

3.4 Summary

This chapter has addressed a new problem of modeling of activity routines in human-inspired memory forms. A new multi-memory neural network model ADLART and its recommendation variation RADLART are proposed. Specifically, three learning algorithms for learning activity routines are presented. To evaluate the algorithms, an extensive set of experiments are conducted on simulation as well as collected real-world data sets. Promising empirical results validate the effectiveness of the proposed routine learning models. Despite the encouraging results, some limitations and open challenges of the current work remain. One issue is about the encoding of time, in the ADLART model, only the starting time of the ADLs are encoded while their duration and ending

time are not. It would be more attractive if one is able to design a mechanism to encode the entire temporal information.

Chapter 4

Spatial Temporal Modeling of Daily ADL Routines

In the previous chapter, a multi-memory model ADLART and its recommendation variation RADLART are presented. Although our experiments validated that ADLART and RADLART can quickly and effectively learn ADL routines, one major limitation of these models is their inability to encode complete temporal information as only the starting time of activities are encoded. This chapter introduces ART models with an additional spatiotemporal layer that enables these models to encode complete spatiotemporal information. These models are evaluated and compared with the ADLART models.

4.1 Introduction

As described in the previous chapter, ADLART and RADLART can quickly and effectively learn ADL routines. However, the encoding mechanism in both models can only encode partial temporal information, and there is no spatial information encoded which may be necessary in specific applications. In this chapter, an update on the ADL and ADL sequence definitions is introduced to provide emphasis on the spatiotemporal features. Formally, we define an **ADL sequence** S as a set of ADLs as follows:

$$\mathbf{S} = (\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n) \quad (4.1)$$

where each ADL A_i is represented by a tuple

$$\mathbf{A}_i = \langle a_i, s_i, e_i, l_i \rangle \text{ for } i = 1, \dots, n, \quad (4.2)$$

where a_i denotes the activity ID, s_i denotes the activity starting time, e_i denotes the activity ending time, l_i denotes the location of the ADL, and n denotes the total number of identified ADLs. Please note that the ADL sequence may not be an ordered set because each ADL has encoded its temporal information. The order of ADLs sequence can be reconstructed by using ADLs' temporal information. As shown in this definition, the set of ADLs, as well as their spatiotemporal information, are of great importance in ADL sequences. In real life, different applications have different sets of ADLs. The time when the ADLs occurred represents the order of the ADLs. The duration of an ADL also carries important information (e.g., wellness value). For example, exercising for 10 minutes is different from exercising for 2 hours.

An **ADL routine** refers to an ADL sequence that describes a person's ADLs in a typical day. A routine can be viewed as a template that captures certain regularities in the ADL order, occurrence time, and duration. A person may have different ADL routines for different days. ADL daily routines are important in the healthcare domain because they largely reflect a person's wellness. In real life, people give ADL routine advices to others for wellness purposes, for example, "go out for a walk after dinner" and "don't go to sleep too late". Beyond healthcare for elderly people, activity routine learning could also provide knowledge for related domains such as manufacturing and office scenarios. To define an ADL daily routine in a computational model, we view a

routine as the template of an ADL sequence that tolerates a certain level of variation in the ADL order, starting time, and duration.

However, modeling ADL daily routines on any given day poses many challenges. First, an ADL routine may contain varying number of ADLs, and the number of ADLs within a day largely depends on the context of the day, as well as other factors, such as the time of the year or weather. Second, defining the set of ADL-related features, for example the ADL types, temporal features, and spatial features, to be used in daily routine clustering is another problem. In particular, the ADL temporal information may contain the starting time, duration, and related day information, whereas the spatial information may contain the room information and location within the room. The selection and multimodal representation of features will affect the clustering performance. Furthermore, user-related information should be taken into consideration. This information provides another set of features that gives a better representation of a person's individualized connection between the spatiotemporal features of ADLs and his or her daily routine, and this understanding could provide better detection of activity patterns and recommendation services.

Daily routine applications are usually domain-specific such that only a certain set of ADLs are of interests. In some public ADL data sets [60], ADLs are usually collected (or annotated) from sensory inputs, and some ADLs (e.g., meal preparation) may be identified frequently throughout a day. In the extreme cases, a single ADL type could take more than half of the total number of ADLs. To reduce the weights of each individual occurrence of the same ADL types, it is necessary to normalize the weighting of each ADL. On the other hand, in different application domains, for example, in finding eating habits or measuring exercise patterns, various ADLs contribute differently and have different importance in the clustering of daily routines. Hence, an ADL weight assignment algorithm and a systematical weight assignment scheme are needed.

The ADLART proposed in Chapter 3 incorporates EM-ART (episodic memory adaptive resonance theory) [18] [17] to learn ADL sequence patterns. In EM-ART, the activities of events in an episode are decayed exponentially, so that the most recent events have higher weights in the code competition than the earlier events. This setting is not suitable for ADL daily routine learning because all ADLs in the same day should have the same level of importance, regardless of when they occur. In ADLART, an ADL daily routine is modeled as an episode of ADLs, using the normalized starting time over the day as the node activation values. The episode of an ADL daily routine is stored in both episodic memory and semantic memory. The semantic memory represents the routines for various days. However, there are several limitations of the ADLART model. First, because of the time representation of ADLs, ADLART only captures the starting time of ADLs, which loses the important ADL duration information. Secondly, ADLART does not incorporate the spatial information of ADLs.

Generally speaking, there are at least two main approaches to modeling activity temporal information in a neural network. In the first approach, temporal information, for example, activity sequences, are modeled as a time series. In recurrent neural networks (e.g., LSTM) and spiking neural networks [61] [62], temporal relationships are implicitly modeled as input iterations. On the other hand, time could be explicitly encoded as the activation of input nodes [19]. Moreover, if complement coding [48] is applied, the encoded time and the complement code could be used to learn a range of time in a continuous space. This approach provides the possibility for temporal features to be used in clustering activities.

For k-means and fuzzy k-means based clustering algorithms, feature weighting algorithms [63] [64] [65] [66] are proposed to automatically calculate a set of feature weights to maximize the specific internal cluster evaluation measures, for example, accuracy, F-1 score, and normalized mutual information.

For clustering algorithms that are based on the adaptive resonance theory (ART), fuzzy ARTMAP with adaptively weighted distances (FAMawd) [67] substitutes the regular L_1 -norm with a weighted L_1 -norm to measure the distances between categories and input patterns. The distance-related weights are a function of a category's shape, allowing for bias in the direction of a category's expansion during learning. In another work, FAMRFW [68] extended FAMawd with another new distance measure.

These measures make assumptions of the shapes of clusters in the sample space, and the feature weights assignment algorithms strive to find the weights to fit into these cluster shapes. However, for learning ADL daily routines, it is usually the expert knowledge in different application domains that provides the cluster shape assumptions. As features, different ADLs are weighted differently in different applications. For example, in food-related studies, meal-related ADLs are important, whereas in physical-wellness-related applications, the body-movement-related ADLs are of more interest.

A three-layer self-organizing neural network model called the Spatiotemporal ADL Adaptive Resonance Theory (STADLART), for learning ADL daily routines, is proposed. STADLART is built upon fusion ART which is a human-inspired neural network theory. Through the learning process that occurs across the three layers, STADLART is capable of learning spatiotemporally distinct ADLs and ADL daily routines by using encoded time, space, and activity information across the multimodal pattern channels. STADLART models ADL and ADL daily routines in two different layers, forming a deep neural network. The newly introduced spatiotemporal features enable STADLART to organize ADLs based on their spatiotemporal features. Also, STADLART incorporates an algorithm applying weight normalization according to the number of the same type of ADLs in a day. This algorithm is designed to perform weight customization for different application domains. To facilitate the use of this algorithm, a weight assignment scheme

and a set of newly designed cluster measurements are proposed to incorporate expert knowledge on daily routine clustering in different applications.

Empirical experiments on a synthetic data set and a public data set [60] were conducted to validate the performance of STADLART. The public CASAS data set is published by Washington State University [60] and contains long-term sensory data captured from several testbeds using real human participants. The CASAS data set is popular in the ambient intelligence field for studies on ADL-related problems. The experimental results show that STADLART, together with the ADL weight assignment algorithm, could cluster ADL daily routines according to different application requirements.

4.2 STADLART Architecture

To learn daily ADL routines, we propose a self-organizing neural network model called the Spatiotemporal ADL Adaptive Resonance Theory (STADLART), which is a three-layer fusion ART network, as shown in Figure 4.1. The first layer consists of the input fields that represent the information of the ADL type, time, day, and space. The second layer contains the spatiotemporal ADL field, wherein the category nodes encode the associations of ADL types and spatiotemporal information. The third layer contains the ADL routine field, wherein the ADL routine nodes encode the sequential combinations of the spatiotemporal ADLs. Across the three layers, the ADL patterns (in F_1 layer) are generalized into spatiotemporally distinguished ADLs (in F_2 layer) and then into ADL daily routines (in F_3 layer). A detailed description of the three layers is presented in the following subsections.

4.2.1 Encoding ADLs in STADLART

In STADLART (see Figure 4.1), the input fields in F_1 encode the ADL type, time, day, and spatial information.

Figure 4.1: STADLART model

4.2.1.1 ADL field

In different problem domains, the sets of ADLs used are different [69][70][71]. In STADLART, the set of ADLs are selected by considering the significance they have on wellbeing and their availability in the public data sets. The list of ADLs used in STADLART is summarized in Table 4.1.

The ADL field represents the type of the input ADL event. Let

$$\mathbf{x}^a = (x_1^a, \bar{x}_1^a, x_2^a, \bar{x}_2^a, \dots, x_8^a, \bar{x}_8^a) \quad (4.3)$$

denote the activity vector, where x_i^a indicates the ADL type, while $\bar{x}_i^a$ is its complement. Although in the current settings the observed activities are assumed to be totally certain (x_i^a to be 1 or 0), the model is capable of handling the fuzzy inputs $[0, 1]$ interval.

Our work adopts the ADL set used and annotated in the CASAS data set. In problem domains where the input samples contain unrecognized activities “other activity”, our

Index	ADL	Index	ADL
1	Meal Preparation	5	Washing Dishes
2	Eating	6	Toilet
3	Working	7	Outside
4	Sleeping	8	Housekeeping

Table 4.1: ADL types used by STADLART

model will simply treat the “other activity” as a type of ADL and learn routines with “other activity” as part of the routine. However, as the “other activity” category is broad and unknown, it may not be very useful. Moreover, if later a new ADL type is identified, it is possible to create new nodes in the F_1^a field and create their links to the next layer. This process is similar to the template matching and node creation described in standard fusion ART and this will not affect the previously learnt associations of other nodes.

4.2.1.2 Time field

In STADLART, the time field F_1^t represents the starting time and duration of the activities performed by the user. In particular, F_1^t contains a vector with a normalized (over a day) starting time and the complement of the normalized ending time.

Let

$$\mathbf{x}^t = (x_1^t, \bar{x}_2^t) \quad (4.4)$$

denote the activity vector, where x_1^t represents the normalized starting time over a day, while $\bar{x}_2^t$ represents the complement of the normalized ending time. Based on fuzzy ART, this scheme that consists of a pair of complement-coded activity values (start time and complement of end time) is sufficient to encode a time interval.

4.2.1.3 Day field

The day field F_1^d contains the day type information, including the day of week and special days. Let $\mathbf{x}^d$ denote the activity vector of F_1^d . We have

$$\mathbf{x}^d = (x_1^d, \bar{x}_1^d, x_2^d, \bar{x}_2^d, \dots, x_{12}^d, \bar{x}_{12}^d), \quad (4.5)$$

where x_n^d indicates the activation value of the n th day type, while $\bar{x}_n^d$ is its complement.

In STADLART, we identify a total of 12 day types, as listed in Table 4.2.

Index	Day Type	Index	Day Type
1	Monday	7	Sunday
2	Tuesday	8	Weekday
3	Wednesday	9	Weekend
4	Thursday	10	Public Holiday
5	Friday	11	Sick Day
6	Saturday	12	Vacation

Table 4.2: Day information input

More than one element could be activated at the same time in the day information vector $\mathbf{x}^d$. For example, if a public holiday falls on Monday, the values of x_1^d , x_8^d , and x_{10}^d will be 1s while others will be 0s. Complement coding is then applied accordingly.

4.2.1.4 Spatial field

The ADL field F_1^s represents the spatial information of the input ADL event. Let $\mathbf{x}^s$ denote the activity vector in this field. We have

$$\mathbf{x}^s = (x_1^s, \bar{x}_1^s, x_2^s, \bar{x}_2^s, \dots, x_6^s, \bar{x}_6^s), \quad (4.6)$$

where x_n^s indicates the spatial information, for example, the room types or being outdoors, of the target ADL, while $\bar{x}_n^s$ is its complement. Referring to the literature on activity recognition, six room types (including outside) are considered in STADLART, as listed in Table 4.3. One example for $\mathbf{x}^s$ could be “having a shower in the washroom”, which is represented as an input vector of $\mathbf{x}^s = (0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1)$ (with the odd elements representing the complement of the even elements).

Index	Room
1	Living Room
2	Bedroom
3	Kitchen
4	Washroom
5	Reading Room
6	Outside

Table 4.3: Spatial types

4.2.2 Encoding Daily ADL Routines

An ADL daily routine refers to an ADL sequence that describes a person’s ADLs in a day. The self-organizing spatiotemporal ADL layer F_2 contains a single field that stores the spatiotemporal ADL categories associated with all the input information. Every basic ADL type may have multiple spatiotemporal ADL categories inside the F_2 layer. For example, “exercising in the morning in the living room” is a different spatiotemporal ADL from “exercising in the afternoon in the bedroom”. The current activity vector $\mathbf{y}$ is learned from the F_1 layer, and a winning node will be identified in $\mathbf{y}$. Let the performed activity vector $\mathbf{y}'$ denote the ADLs performed in a day, and we have

$$\mathbf{y}' = (y_1, \bar{y}_1, y_2, \bar{y}_2, \dots, y_m, \bar{y}_m), \quad (4.7)$$

where m is the number of spatiotemporal ADL categories in the F_2 layer, y_m indicates whether the m th spatiotemporal ADL category is performed in the day, and $\bar{y}_m$ is its complements. $\mathbf{y}'$ is used to learn the daily routine encoded in the next layer F_3 . The activities in $\mathbf{y}'$ form a day’s ADL sequence. In other words, $\mathbf{y}'$ has the size of all learned spatiotemporal ADLs, and the spatiotemporal ADLs performed in the current day have their activation value set to 1, while those not performed have their activation values set to 0.

The F_3 layer is the ADL daily routine layer. The nodes in the F_3 layer are learned from the ADL vector $\mathbf{y}'$ of the F_2 layer, and each node represents a unique ADL daily routine of the user. The activity vector of F_3 is denoted by $\mathbf{z}$. The detailed spatiotemporal information of each component of the ADLs could be retrieved by tracing through the STADLART architecture.

4.3 Learning and Retrieval of Daily Routines

4.3.1 Model Training

The STADLART neural network consists of three layers. The F_2 layer focuses on individual ADLs, while the F_3 layer combines the ADLs from the F_2 layer to form the daily ADL sequences. In the training phase, STADLART needs to go through two steps to start learning. First, STADLART learns through the F_1 layer to the F_2 layer for each ADL input. STADLART learns other individual ADLs in the same way until all ADLs in the day have been learned. Second, STADLART will learn the F_3 layer from the ADLs sequence. The training algorithm is summarized in Algorithm 4.

Algorithm 4 STADLART training process

Require: A sequence of ADLs, each in the form of $(\mathbf{x}^a, \mathbf{x}^t, \mathbf{x}^d, \mathbf{x}^s)$

Ensure: Learn an ADL routine

- 1: **for** each input ADL $(\mathbf{x}^a, \mathbf{x}^t, \mathbf{x}^d, \mathbf{x}^s)$ **do**
 - 2: Compute activity vector $\mathbf{y}$ in the F_2 layer using input patterns $(\mathbf{x}^a, \mathbf{x}^t, \mathbf{x}^d, \mathbf{x}^s)$
 - 3: The winning F_2 node learns a spatiotemporal ADL category
 - 4: Update the performed spatiotemporal ADL list $\mathbf{y}'$
 - 5: **end for**
 - 6: Update activity vector $\mathbf{z}$ in the F_3 layer using input patterns $\mathbf{y}'$
 - 7: The winning F_3 node learns an ADL routine category from $\mathbf{z}$
-

4.3.2 Daily Routine Readout

After training the STADLART model, each node in the F_3 layer encodes the learned daily ADL routines. By reading the activation values of spatiotemporal ADLs associated

with the daily routine categories, a list of routines can be generated from the F_2 layer. At the F_1 layer, the learned starting time, duration, and spatial information associated with the spatiotemporal ADL categories can be retrieved.

4.3.3 Spatiotemporal Information Retrieval

The spatiotemporal characteristic of STADLART allows the retrieval of a particular ADL by using the user's spatial and temporal preference. This is STADLART's capability for learning the ADL spatiotemporal patterns of the user. Specifically, for a particular ADL type input $\mathbf{x}^a$, STADLART will activate every category j in the F_2 layer using the choice function

$$T_j^2 = \sum_{k=1}^n \gamma_2^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha_2^k + |\mathbf{w}_j^k|}. \quad (4.8)$$

All the spatiotemporal ADL nodes in the F_2 layer that match the ADL type $\mathbf{x}^a$ will be selected. The spatial and temporal information associated with the selected categories will be retrieved and output. The algorithm is summarized in Algorithm 5.

Algorithm 5 Retrieve spatiotemporal information using ADL

Require: ADL vector $\mathbf{x}^a$ from F_1^a

Ensure: x^s , x^t , and x^d associated with ADL type $\mathbf{x}^a$

- 1: Read in the ADL vector $\mathbf{x}^a$ from F_1^a
 - 2: Activate every category j in F_2 by choice function $T_j^{F_2} = \sum_{k=1}^n \gamma_2^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha_2^k + |\mathbf{w}_j^k|}$
 - 3: **while** selecting a new category J and $T_J^{F_2} > 0$ **do**
 - 4: Readout the x^t , x^d , and x^s associated with J
 - 5: **end while**
-

4.4 ADL Weight Normalization and Customization

4.4.1 ADL weight normalization

With real-life ADL data, for example, [60], ADLs are usually collected (or annotated) based on sensory inputs. As such, certain ADLs (e.g., meal preparation) may occur frequently in a day. For example, a stove may be turned on and off seven times a day, and thus, seven meal preparation ADLs would be logged. To normalize the contribution of ADLs in a day, a weighting scheme is introduced and applied to the STADLART choice function (Formula (4.8)):

$$T_j^2 = \sum_{k=1}^n \gamma_2^k \frac{|\mathbf{n}(\mathbf{x}^k \wedge \mathbf{w}_j^k)|}{\alpha_2^k + |\mathbf{n}\mathbf{w}_j^k|}. \quad (4.9)$$

where the feature normalization vector $\mathbf{n}$ is defined by $n_i = 1/a_i$, and a_i is the count of spatiotemporal ADL type i in the day. The resultant algorithm is therefore named STADLART-N. With the introduction of $\mathbf{n}$, the frequently occurred ADLs will have less weight for each of their occurrences while the less frequent ADLs will have a higher weight for each occurrence. For example, if there are seven meal preparation ADLs identified within one day, after applying the normalization, each meal preparation ADL contributes one seventh compared with before.

4.4.2 Customizable ADL weights

Besides feature normalization according to the frequencies of ADLs, different ADLs play different roles in different application domains, bringing in different levels of importance. To emphasize the ADLs of interest in specific applications, similar to the feature weight normalization vector, a feature weight customization vector $\mathbf{c}$ is introduced to the choice function (Formula (4.8)) in the ADL daily routine learning algorithm

$$T_j^2 = \sum_{k=1}^n \gamma_2^k \frac{|\mathbf{nc}(\mathbf{x}^k \wedge \mathbf{w}_j^k)|}{\alpha_2^k + |\mathbf{ncw}_j^k|}. \quad (4.10)$$

where the weight customization vector $\mathbf{c}$ assigns weights to individual ADL i according to its importance. The new algorithm is accordingly named STADLART-NC. To formalize weight customization, a five-level scale is used to measure the importance of each ADL and assign corresponding weight values. As shown in Table 4.4, very important ADLs will have their weight set to 1 while other ADLs will have their weight below 1 according to their domain knowledge.

ADL importance	weight
Not Important	0.2
Less Important	0.4
Normal	0.6
Important	0.8
Very Important	1.0

Table 4.4: Assignment of ADL weight according to its importance

4.5 Experiments on Synthetic Data

4.5.1 Synthetic Data Generation

The performance of STADLART is first evaluated using a synthetic data set that is generated using a set of predefined ADL routines. The experiments on synthetic data serve to verify the veracity of the output when the training samples are generated from few templates and whether the model could generate clusters corresponding to the templates which are hidden from the model. To gain more confidence on veracity, real-world scenarios will be gradually added to the synthetic data making it more “real”. In the first experiment, five synthetic ADL routines were created, with each describing a specific type of day, namely the normal day, the wake up early day, the hungover day, the outing

day, and the housework day. About 2,000 training samples (days of ADLs) are generated. When generating a daily ADL sequence sample, the day is first randomly assigned a routine template, and then, all the ADL types are generated according to the assigned routine template, with each ADL having a 15-minute random variation in the starting time and duration.

Figure 4.2: Synthetic routine templates

A normal day routine, which is an example of a normal day, as shown in Figure 4.2, starts at 7:00. The person performs personal hygiene (Toilet) and has breakfast (Eating) before 8:00. Lunch preparation (MealP) is around 10:00, followed by having the lunch (Eating) before noon. The person prepares dinner (MealP) at 16:00, and dinner (Eating) starts at 18:00. The dishes are washed (WashDish) shortly after finishing dinner and finally the person goes to sleep (Sleep) at about 22:00. In the “wake up early” day routine, the person wakes up at about 6:00, and every ADL is performed about one hour earlier. In the “hungover” day routine, the person wakes up late, near noon, with no breakfast, and all other ADLs are postponed by around one hour. In the “outing day” routine and the “housework” day routine, there are outdoor ADL types and housework ADL types that distinguish these two types of days from other days.

4.5.2 Experiments and Evaluation

STADLART was trained throughout the experiments with slow learning by setting the ART learning parameter $\beta = 0.2$. Consequentially, a spatiotemporal ADL in a routine

that is not observed in the subsequent inputs of this routine will have its association strength decreased gradually.

In fusion ART networks, the vigilance parameters control the level of generalization. In this set, STADLART was run with the vigilance parameters of both F_1 and F_2 fields set from 0.8, 0.85, 0.9, to 0.95. The number of categories learned in the F_3 layer are listed in Table 4.5. With the vigilance values of 0.85 and 0.9, STADLART learned exactly five daily routine categories. Comparing the readouts of the F_3 layer with the synthetic data, it is clear that the categories in the F_3 layer represent exactly the routine templates in the synthetic data set.

Vigilance	Categories
0.80	4
0.85	5
0.90	5
0.95	17
1.00	220

Table 4.5: The number of daily routine categories generated using different vigilance values

Using the experiment with $\rho_2 = 0.85$ as an example, Figure 4.3 shows a learned daily routine category that corresponds to the normal day routine template.

Figure 4.3: A sample of learned ADL routines

The extended periods of ADLs, for example, eating, in the learned categories occur because of two main reasons. The first reason is that ADL generation has a 15-minute

random variation and thus has longer ADL periods. The second reason is that the generation of ADLs spans multiple days.

Because the templates for generating samples provide the label information, the STADLART output clusters can be evaluated with external evaluation indices, specifically accuracy, F-1 score, normalized mutual information (NMI), and the Jaccard index. Using these measures, STADLART is compared with our previous work ADLART, baseline algorithms (namely K-means), and the LSTM network (the implementation in `deeplearning4j` project [72]). Because of the limitation of ADLART, ADLs at different times of the day (e.g., morning, afternoon, and evening) are identified as different features. For example, meal preparation in the morning is considered a different feature from meal preparation in the afternoon. However, ADLs of the same type within the same part of day are only counted once. For K-means, each input sample is an ADL routine containing a fixed number of ADLs (if the ADLs are not performed in the day the fields will simply be 0s). Each cluster center represents one routine learned from several samples. The difference of element ADLs between the new set of ADLs and the cluster center is calculated as the distance, weighted in temporal and spatial differences.

Algorithm	Accuracy	F-1	NMI	Jaccard Index
STADLART	0.99 ± 0.002	0.94	0.92	0.91
ADLART	0.95 ± 0.008	0.88	0.88	0.85
K-means	0.94 ± 0.011	0.88	0.87	0.84
LSTM	0.99 ± 0.001	0.93	0.92	0.90

Table 4.6: The external index scores for the synthetic data set

As shown in Table 4.6, STADLART outperformed the ADLART model and baseline clustering algorithms significantly. The advantage of STADLART is largely due to the spatiotemporal features that could differentiate the same type of ADLs based on their temporal differences. At the same time, LSTM shows a similar level of performance with STADLART on the synthetic data set.

4.6 Experiment on Real-life Data Set

4.6.1 CASAS Data Set and Preprocessing

The CASAS data set consists of a total of 38 data sets that contain sensory inputs collected from well-equipped testbeds in several cities across the world. The 17th data set was chosen for the following experiment. The data set, collected from the testbed in Aruba, contains the sensory readings of an elderly female tenant over a period of 220 days. The data collected are primarily sensory readings with annotated ADLs as ground truths, which are taken as the inputs for STADLART. For the real-world data set, the results are checked against the original data set manually and verified whether the generated routines are representative.

ADL	Count
Meal_Preparation	1604
Eating	257
Working	171
Sleeping	400
Washing_Dishes	65
Bed_to_Toilet	157
Leave_Home	318
Housekeeping	33

Table 4.7: Annotated ADL types in the preprocessed CASAS data set

Data preprocessing was performed on the raw CASAS data set. First, the ADL type Enter_Home is combined with the ADL type Leave_Home. The duration of Leave_Home is hence the time span from Leave_Home to Enter_Home. Second, the ADL type “Resperate” has only six instances. Because it is not significant, this ADL type is removed. Third, there are 2,910 instances of the ADL type “relax” over the 220 days, with an occurrence count of more than 10 times per day, each spanning a very short duration. This ADL type has no significance in differentiating the daily routine and is thus also removed. In the CASAS data set, dates were given instead of days of the week. Therefore,

a conversion with reference to a calendar was performed. Because ADL are recognized from fixed sensors (stove, water tap, door, sofa, etc.) in the six rooms, the ADLs are associated with fixed spatial information. The final set of used ADL types is shown in Table 5.4, giving a total of 3,005 ADL samples.

With this preprocessed data set, experiments are conducted to test the generalization behavior of STADLART in terms of categories learned in different layers with different parameter configurations. Again, STADLART is compared with baseline algorithms, ADLART, and LSTM. Finally, the feature normalization and customization versions, STADLART-N and STADLART-C, are evaluated.

4.6.2 Experiments on spatiotemporal ADL category generalization

In this experiment, we look at the spatiotemporal ADL category layer. The ADL types associated with the temporal and spatial information are recognized and stored in this layer. We evaluate the generalization behavior and find the key ADL categories to represent a user’s ADL preference. We conduct this experiment in a fast learning setting, i.e., with a learning rate of 1.0. The contribution parameters are set to 0.4 for ADL type, 0.4 for temporal, and 0.2 for spatial. As discussed in the previous subsection, the vigilance parameter values for all the fields are set to 1.0, which means all nodes in the spatiotemporal ADL category represent an ADL type with a fixed temporal and spatial category. As a result, we generated 375 spatiotemporal ADL categories from the 3,005 input samples. In other words, each spatiotemporal ADL category represents about five inputs.

Among the 375 categories, the top 10 categories represent 33, 32, 23, 22, 22, 21, 20, 20, 19, and 19 inputs. At the same time, there are 100 categories that represent only one input and 62 categories that represent two inputs. The top categories mostly represent

Sleep, Meal_Preparing ADL, and Leave_Home. This is because these three ADLs appear in the most number of input instances.

4.6.3 Experimentd on daily routine categories generation

In this experiment, the learning process from the Spatiotemporal ADL category layer F_2 , to the ADL daily routine category layer F_3 , is evaluated, where F_2 contains the learned spatial temporal ADL categories that this person performed in his daily life. The last step is to learn the ADL daily routines that could be used to describe his personal behavior or lifestyle.

In the first experiment, fast learning is used with learning rate $\beta = 1.0$. The relationship between the choice of the vigilance parameter value of spatiotemporal ADL field and the number of daily routine categories generated is illustrated in Figure 4.4.

Figure 4.4: The number of daily routine categories generated with different vigilance parameter values

Subsequently the vigilance parameters ρ_2 and ρ_3 are chosen to be 0.9 and 0.2 for F_2 and F_3 , respectively. Because more generalization is preferred, ρ_3 for learning daily routines is set to a low value. Based on 220 samples, the STADLART model learned a total of 19 daily routines from the CASAS data set. To verify the veracity of the result, the top five routines are manually examined by randomly picking five days of that

routine and checking the raw data. The top three routines represent 25, 20, and 17 days, respectively. The common ADL types in the top two routines are shown in Figure 4.5. In the first routine, it can be seen that the elderly woman visited the toilet regularly in the morning between 4:00 and 6:00. However, in the second routine, on certain days, the elderly woman prepared meals throughout the day, and she may be preparing a big dinner with her visiting daughter. The results of the manual verification are consistent with the learnt routines. From the results, we could see that in reality, the activity routines are more diverse than the synthetic data, and there is not a single dominant daily routine over the 220 days. However, certain interesting ADL patterns can be observed.

Figure 4.5: Two ADL routines learned by STADLART

4.6.4 Comparing STADLART with other algorithms

Because the routines are not labeled in the CASAS data set, the results from STADLART are compared with the ADLART and the baseline algorithms using an intercluster measure. Specifically, the overall average silhouette width [73] is used, which indicates the quality of the underlying structure of the clusters: a higher value indicates a stronger structure. For ADLART and other clustering algorithms, the CASAS data set is preprocessed in a similar way as the synthetic data set, wherein ADLs in different parts of day are treated as different features. The evaluation results from clustering the spatiotemporal ADLs and ADL routines are shown in Table 4.8. The results show that the learned

clusters from the CASAS data set are much less structural than the clusters learned from the synthetic data. Nevertheless, for both data sets STADLART outperforms ADLART and the baseline K-means clustering algorithm. Perhaps due to the small sample size (220 in the CASAS data set), deep neural network LSTM does not significantly out-perform the baseline algorithms.

Algorithm	Synthetic-ASW	CASAS-ASW
STADLART	0.93	0.32
ADLART	0.90	0.31
K-means	0.90	0.30
LSTM	0.93	0.30

Table 4.8: The overall average silhouette width for the data sets

4.6.5 Experiment on feature weight normalization and customization

In this subsection, the original STADLART is first compared with its feature weight normalization variation, STADLART-N based on the CASAS data set. Similar to the previous experiments, fast learning is used here with a learning rate of $\beta = 1.0$. Vigilance parameters ρ_2 and ρ_3 are set to be 0.9 and 0.2 for F_2 and F_3 , respectively.

Algorithm	Number of Routines	Size of Top Three Routines
STADLART	19	25, 20, 17
STADLART-N	18	31, 28, 25

Table 4.9: Comparing STADLART and STADLART-N using the CASAS data set

As shown in Table 4.9, with feature weight normalization, STADLART-N learns a similar number of routines to STADLART on the CASAS data set. However, the top routines now represent more entries in the data set. By reading out the associations at the F_2 and F_1 layers, the top two routines are plotted in Fig. 4.6. Compared with the

top routines generated by ADLART (Fig. 4.5), which contain mostly meal_preparation ADL, STADLART-N generated more meaningful routines from the data set.

Figure 4.6: Top two ADL routines learned by STADLART-N

In the second experiment, two scenarios involving meal and outing behavior are evaluated, with an emphasis on eating and outing ADL respectively. According to the ADL weight assignment guideline in Table 4.4 and suppose we would like to look into meal analysis, the weight of eating is set to “very important” (i.e., 1.0), the weight of meal preparation is set to “normal” (i.e., 0.6), and all other ADLs are set to be “not important” (i.e., 0.2). Similarly, in the outing behavior analysis experiment, the outing ADL is set to “very important” (i.e., 1.0), while all other ADLs are set to be “not important” (i.e., 0.2). The experiment results are summarized in Table 4.10. From the results, it is seen that with feature customization, STADLART-C and STADLART-NC could cluster daily ADL sequences into routines while focusing on key ADLs.

Experiment	Algorithm	Number of Routines
Meal	STADLART	19
Meal	STADLART-C	10
Meal	STADLART-NC	8
Outing	STADLART	19
Outing	STADLART-C	7
Outing	STADLART-NC	6

Table 4.10: Comparing STADLART, STADLART-C, and STADLART-NC on the CASAS data set

4.7 A General Spatiotemporal ART Model: ST-ART

4.7.1 Introduction

In the previous section, the STADLART model is presented to learn ADL routines through layers of neural networks. Inside the STADLART model, the ADL type, time, day information, and spatial information are taken as input to generalize spatiotemporal ADLs, which is later used to form ADL routines. By using this design, the time, day information, and spatial information are generalized in the same level to an optimized performance for the ADL domain. To extend the STADLART model to other domains, a generalized model, named spatiotemporal adaptive resonance theory (ST-ART) is proposed in this section. The architecture of the ST-ART model is shown in Figure 4.7.

Figure 4.7: The ST-ART architecture

The main difference between the ST-ART and the STADLART is that the ST-ART model has one additional layer which generalizes spatial and temporal information before the generalization of the spatiotemporal events.

4.7.2 The ST-ART Model Input Fields

The ST-ART model has four layers with seven fields. Six fields are same as the fields in the STADLART model, and the repeated descriptions are skipped in this section. The newly introduced fields is the temporal field F_2^t . The temporal field generalizes the information in both time field and the day information field. The nodes in the temporal field represent combinations of the input fields, for example, Wednesday afternoon 3:00pm to 4:00pm or weekend morning 10:00am to 10:30am. With this specific temporal information, the model can learn and retrieve specific time slots that are important to the user.

4.7.3 Encoding and Retrieval

4.7.3.1 Learning

In the learning phase, the ST-ART first learns through F_1 layer to F_3 layer for each ADL event input, after a whole day's ADLs had been received, the ST-ART learns F_4 layer from F_3 layer. The learning algorithm for F_1 layer to F_3 layer is summarized in Algorithm 6, and the learning algorithm for F_3 layer to F_4 layer is summarized in Algorithm 7.

4.7.3.2 Recognize ADL Sequence

Similar to ADLART model, ST-ART model is able to recognize a learnt node in F_4 layer using a ADL sequence input from F_3 , the algorithm is summarized in Algorithm 8.

4.7.3.3 Retrieve Spatial Temporal information

Based on the special spatial temporal feature, ST-ART is able to recognize an ADL node in F_3 by using a ADL type input from F_2^a . After that, ST-ART is able to retrieve related spatial and temporal information of the queried ADL type. The algorithm is summarized in Algorithm 9.

Algorithm 6 ADL event learning

for each temporal input vector $(\mathbf{x}^t, \mathbf{x}^d, \mathbf{y}^a, \mathbf{y}^s)$ **do**
 Activate every category j in F_2^t by choice function $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|}$
 select category J such that $T_J = \max\{T_j : \text{for all } F_2^t \text{ node } j\}$
 set node activation $y_J \leftarrow 1$
while match function $m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k$ (not in resonance)
 or J was selected previously **do**
 deselect and reset J by $T_J \leftarrow 0, y_J \leftarrow 0$
 select another node J with $T_J = \max\{T_j : \text{for all } F_2^t \text{ node } j\}$
end while
 calculate $\mathbf{y}^t \leftarrow \frac{\text{ADL occurrence time}}{\text{Whole day time}}$
if resonance occurs(Temporal Pattern Matched) **then**
 $\mathbf{w}_J^{k(\text{new})} = (1 - \beta^k) \mathbf{w}_J^{k(\text{old})} + \beta^k (\mathbf{x}^k \wedge \mathbf{w}_J^{k(\text{old})})$
else(No matching/resonance)
 let $J \leftarrow J^0$, where J^0 is a newly recruited uncommitted nodes in F_2
 Learn $(\mathbf{y}^t)$ in F_2^T as a novel temporal node with $\mathbf{w}_J^{k(\text{new})} = \mathbf{x}^k$
end if
 Activate every category j in F_3 by choice function $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{y}^k \wedge \mathbf{v}_j^k|}{\alpha^k + |\mathbf{v}_j^k|}$
 select category J such that $T_J = \max\{T_j : \text{for all } F_3 \text{ node } j\}$
 set node activation $y_J \leftarrow 1$
while match function $m_J^k = \frac{|\mathbf{y}^k \wedge \mathbf{v}_J^k|}{|\mathbf{y}^k|} \geq \rho^k$ (not in resonance)
 or J was selected previously **do**
 deselect and reset J by $T_J \leftarrow 0, y_J \leftarrow 0$
 select another node J with $T_J = \max\{T_j : \text{for all } F_3 \text{ node } j\}$
end while
if resonance occurs(ADL recognized) **then**
 $\mathbf{v}_J^{k(\text{new})} = (1 - \beta^k) \mathbf{v}_J^{k(\text{old})} + \beta^k (\mathbf{y}^k \wedge \mathbf{v}_J^{k(\text{old})})$
else(No matching/resonance)
 let $J \leftarrow J^0$, where J^0 is a newly recruited uncommitted nodes in F_2
 Learn $(\mathbf{y}^a, \mathbf{y}^t, \mathbf{y}^s)$ in F_3 as a novel ADL category with $\mathbf{v}_J^{k(\text{new})} = \mathbf{y}^k$
end if
end for

Algorithm 7 ADL sequence learning

for each ADL sequence input vector ($\mathbf{Z}$) in F_3 **do**
 Activate every category j in F_4 by choice function $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{z}^k \wedge \mathbf{u}_j^k|}{\alpha^k + |\mathbf{u}_j^k|}$
 select category J such that $T_J = \max\{T_j : \text{for all } F_4 \text{ node } j\}$
 set node activation $y_J \leftarrow 1$
 while match function $m_J^k = \frac{|\mathbf{z}^k \wedge \mathbf{u}_J^k|}{|\mathbf{z}^k|} \geq \rho^k$ (not in resonance)
 or J was selected previously **do**
 deselect and reset J by $T_J \leftarrow 0, y_J \leftarrow 0$
 select another node J with $T_J = \max\{T_j : \text{for all } F_4 \text{ node } j\}$
 end while
 if resonance occurs(ADL sequence recognized) **then**
 $\mathbf{u}_J^{k(new)} = (1 - \beta^k)\mathbf{u}_J^{k(old)} + \beta^k(\mathbf{z}^k \wedge \mathbf{u}_J^{k(old)})$
 else(No matching/resonance)
 let $J \leftarrow J^0$, where J^0 is a newly recruited uncommitted nodes in F_4
 Learn ($\mathbf{Z}$) in F_4 as a novel ADL category with $\mathbf{u}_J^{k(new)} = \mathbf{z}^k$
 end if
end for

Algorithm 8 ADL Sequence Activation and Readout

for each ADL sequence input vector ($\mathbf{Z}$) in F_3 **do**
 Activate every category j in F_4 by choice function $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{z}^k \wedge \mathbf{u}_j^k|}{\alpha^k + |\mathbf{u}_j^k|}$
 select category J such that $T_J = \max\{T_j : \text{for all } F_4 \text{ node } j\}$
 set node activation $y_J \leftarrow 1$
 while match function $m_J^k = \frac{|\mathbf{z}^k \wedge \mathbf{u}_J^k|}{|\mathbf{z}^k|} \geq \rho^k$ (not in resonance)
 or J was selected previously **do**
 deselect and reset J by $T_J \leftarrow 0, y_J \leftarrow 0$
 select another node J with $T_J = \max\{T_j : \text{for all } F_4 \text{ node } j\}$
 end while
 if resonance occurs(ADL sequence recognized) **then**
 readout category J
 else(No matching/resonance)
 display no ADL sequence found message
 end if
end for

The ST-ART can also be used to identify a particular type of ADL by providing related spatial and temporal information. In other words, given F_1^t , F_1^d , and F_2^s , ST-ART is able to identify corresponding category in F_2^a . The algorithm is summarised in

10

Algorithm 9 Retrieve Spatial Temporal Information Using ADL Type

```

read in the ADL type input ( $\mathbf{y}^a$ ) from  $F_2^a$ 
Activate every category  $j$  in  $F_3$  by choice function  $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{y}^k \wedge \mathbf{v}_j^k|}{\alpha^k + |\mathbf{v}_j^k|}$ 
while select a new category  $J$  such that  $T_J > 0$  do
    Readout the  $y^t$  and  $y^s$  associated with  $J$ 
    Readout the  $x^t$  and  $x^d$  associated with  $y^t$ 
    deselect and reset  $J$  by  $T_J \leftarrow 0, y_J \leftarrow 0$ 
end while
Organize the readouts and prepare for display
    
```

Algorithm 10 Identify ADL Type Using Spatial Temporal Information

```

Read in the temporal input ( $\mathbf{x}^t, \mathbf{x}^d$ ) in  $F_1$ 
Activate every category  $j$  in  $F_2^t$  by choice function  $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|}$ 
select category  $J$  such that  $T_J = \max\{T_j : \text{for all } F_2^t \text{ node } j\}$ 
set node activation  $y_J \leftarrow 1$ 
while match function  $m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k$  (not in resonance)
    or  $J$  was selected previously do
        deselect and reset  $J$  by  $T_J \leftarrow 0, y_J \leftarrow 0$ 
        select another node  $J$  with  $T_J = \max\{T_j : \text{for all } F_2^t \text{ node } j\}$ 
    end while
if resonance occurs(Temporal Pattern Matched) then
    readout the  $\mathbf{y}^t$  associated with  $J$ 
else(No matching/resonance)
    display no ADL with this temporal information found
end if
Read spatial temporal vector ( $\mathbf{y}^t, \mathbf{y}^s$ ) in  $F_2$ 
Activate every category  $j$  in  $F_3$  by choice function  $T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{y}^k \wedge \mathbf{v}_j^k|}{\alpha^k + |\mathbf{v}_j^k|}$ 
select category  $J$  such that  $T_J = \max\{T_j : \text{for all } F_3 \text{ node } j\}$ 
Readout the ADL type category in  $F_2^a$  associated with  $J$ 
    
```

4.7.4 Experiments

To evaluate the performance of ST-ART, a simulation environment is designed to conduct various experiments. The simulator generates a year’s daily ADL sequence data samples from one user’s profile. In the first set of experiments, we evaluate the F_2 layer temporal model generalization property by comparing the number of generated temporal categories using different vigilant parameter values. In the second set of experiments, we look at the F_3 layer. Compare the number of spatiotemporal ADL categories by using different vigilant parameters in the F_2 layer. The key ADL categories will be identified to describe the user’s lifestyle preference. Finally, in the last set of experiments, we look at comparison between ST-ART and ADLART in temporal classification performance.

4.7.4.1 Experiments with Different Vigilance Parameter Values

The vigilance parameter ρ plays an important role in fusion-ART models’ generalization performance. Generally speaking, a higher value of ρ towards 1 will result in a larger number of categories learnt. In the first experiment, we look at the effect of the different vigilance parameter values in the time field, e.g., $\mathbf{x}^t$, toward the categories generated in the temporal field. For the training data, we use the generated one-year daily ADL sequences which contains a list of ADL with the associated temporal information and spatial information. The time field of ADLs has randomization in the starting time and duration.

We use “having breakfast” as the target ADL. In our setting, having breakfast has a starting time between 7:00 and 8:30 in the morning, and the duration is ranging from 15 minutes to 1 hour and 15 minutes. There are total 35 (7 times 5) possible starting time and duration combinations. The vigilance parameter values, ρ^{F1} , ranging from 0.8 to 1.0 are evaluated against the same set of training data. The experiment result is

Figure 4.8: Categories generated with different ρ^{F1} values

plotted in Fig. 4.8. We could see from the figure that when the ρ^{F1} goes beyond 0.97, there is a slightly increasing number of categories learnt by the model. In the subsequent experiments, 0.93, which created around 15 categories in this experiment, is used as the default vigilance parameter value for $\mathbf{x}^t$.

4.7.4.2 Experiments on the ADL category generalization

In the previous experiment, we evaluated the generalization behaviour of ST-ART by using different vigilance parameter values in the F_1 layer. In this experiment, we look at the F_3 layer, which is the spatiotemporal ADL category layer. The ADL types associating with temporal and spatial information are recognized and stored in this layer. We evaluate the generalization behaviour and try to find the key ADL categories to represent a user’s ADL preference.

Similar to the previous experiment, we use the generated one-year samples as the training data, and the target ADL is “having breakfast” with the starting time and duration described in the previous experiment. For the day information, besides the day of week information, twenty random days are identified as sick day while ten other random days are identified as public holidays. In the normal days and public holidays, the user prefers to have breakfast in the kitchen and sometimes in the living room, while in the sick days he prefers to have breakfast in the bedroom.

After training the model, there are 20 breakfast categories learnt. The top 10 categories with their normalized frequencies, times, key day types, and key locations are listed in the Table 4.11.

Index	Freq.	Time	Day Type	Location
1	0.44	7:15-8:15	Weekday	Kitchen
2	0.20	7:00-8:30	Weekend	Kitchen
3	0.09	7:00-8:15	Weekday	Living room
4	0.03	7:15-8:30	Sick day	Bedroom
5	0.03	7:00-8:15	Weekend	Living room
6	0.02	7:00-7:15	Monday	Kitchen
7	0.02	7:45-8:30	Wednesday	Living room
8	0.02	8:00-8:15	Public Holiday	Kitchen
9	0.02	7:30-8:30	Sick day	Kitchen
10	0.02	8:15-8:30	Weekend	Living room

Table 4.11: Breakfast categories created by ST-ART

The result shows that 1) the ST-ART model could generalize basic user’s behaviour patterns for weekdays and weekends, 2) the ST-ART model could identify exceptional occasions, e.g., user has breakfast in the bedroom in sick day, and 3) some small categories, including particular day’s preference in certain time intervals, are identified.

4.7.4.3 Experiment on daily routine categories generation

In this experiment, we look at the learning process from F_3 , ADL category layer to F_4 , ADL daily routine category layer. F_3 contains the learnt spatial temporal ADL categories that a person performed in his daily life. The last step is to learn ADL daily sequences or ADL daily routines that could be used to describe his personal behaviour or lifestyle.

In this experiment, we use the simulation data similar to previous experiments. After training a year’s simulation data, the F_3 layer has learnt big variations of spatial temporal ADL categories. As we want to have some generalization of personal daily routine

patterns, we try to find a low vigilance parameter value, for generating roughly 50 or less categories from 365 samples, for the learning process from F_3 layer to F_4 layer.

After training the model with vigilance parameter value of 0.8, the ST-ART has learnt 46 ADL daily routine categories. The top frequency categories are listed in Table 4.12.

Index	Count	Key Description
1	64	Normal Weekday
2	49	Normal Weekend
3	41	Exercise on Weekday
4	28	Going out for social on Weekend
5	19	Have no breakfast

Table 4.12: Top frequency ADL daily routine categories learnt by ST-ART

4.7.4.4 Comparing ST-ART and ADLART

In this experiment, we compare ST-ART with the ADLART model. ADLART is a two-layer neural network which learns ADL routines from a day type and a sequence of ADLs. First, we generate a set of 300 daily ADL sequence samples with spatiotemporal information for each ADLs using the simulator. In the training phase, 240 samples are used to train both the ST-ART and the ADLART models. In the testing phase, the other 60 samples are used to test the accuracy of recognizing a correct day type. As the ADLART model only takes the starting time of ADLs, the spatial information of the testing samples are hidden. The accuracy is calculated on the correct classification of weekday and weekend versus the testing sample size. The results are summarized in Table 4.13.

With a close look at the models, we could see that the ADLART takes in the ADL sequence as a whole without duration and day type preference information for individual ADLs. However, the ST-ART associates detailed spatiotemporal information with each ADLs. In the sample data, some ADLs, e.g., social and house work, are only performed

Model	Accuracy
ST-ART	0.87
ADLART	0.78

Table 4.13: The accuracy of the ST-ART model and the ADLART model on classifying weekdays and weekends

during the weekend, and ST-ART could utilize this temporal information of individual ADLs for a better classification between weekdays and weekend.

4.8 Summary

In this chapter, a spatiotemporal fusion ART neural network model named STADLART has been presented to learn human daily activity routines. In contrast to the early work of ADLART, STADLART takes into consideration the ADL start time, duration, and spatial information explicitly. Experiments conducted based on a synthetic data set show that STADLART could learn ADL routines which are consistent with the ADL templates used to generate synthetic data. For the CASAS data set, STADLART manages to generalize some interesting routines of a person’s life across days and provide information for further investigation on the person’s behavior. In both experiments, STADLART was compared with ADLART and baseline clustering algorithms such as nearest neighbour and K-means. The results show that STADLART outperformed the baseline clustering algorithms in various aspects. Moreover, a STADLART variation named STADLART-NC was presented to normalize and customize ADL weights for different ADLs in daily routines. The ADL weight normalization successfully reduces the influence of ADL frequency while the ADL weight customization promotes ADLs of interest from expert knowledge. A guideline for ADL weight assignment on ADL weighting customization is also provided for ease of use. Experiments on the CASAS data set further demonstrate

that STADLART-NC could learn more meaningful daily routines with normalized and customized ADL weighting for different application configurations.

By utilizing the STADLART models, an intelligent system will have the knowledge of the typical life routines of the user. Based on this knowledge, various applications are made possible. First of all, the intelligent system can detect abnormalities of the user and notify caregivers or his relatives through messages. The intelligent system can also make predictions of the users, next activities, give recommendations of activities, or provide other activity advices. The user activity routines also provide samples for long-term analysis, for example, activity routine change pattern over the years for certain age group.

Going forward, some limitations of STADLART have to be addressed. For example, the normalization formula makes use of the frequency of the same ADLs performed in the day, making STADLART not suitable for online learning as and when partial ADLs are collected. An online adaptive normalization method would be more desirable to enhance the feature weighting algorithm. Another limitation of STADLART is the lack of automatic explanation capability. Currently, the learned routines and activity patterns are interpreted manually. An automatic algorithm for translating learned patterns to meaningful symbolic representation will be an important direction for our future research.

Besides STADLART, another model ST-ART, which an addition layer of generalization on temporal field, is proposed to extend spatiotemporal learning to other event domains. Experiments show ST-ART is able to learn event patterns effectively. However, as one additional layer of generalization is introduced, the ST-ART requires a larger sample size than STADLART in certain scenarios.

Chapter 5

Sequential Causal Learning by real-time hypothesis formulation and testing

In the previous two chapters, we have introduced ART models that could learn ADL daily routines patterns. In this chapter, we will look at the causal relationship between ADLs. Different from traditional Bayesian-based approaches, in this chapter, we propose an ART-based neural network that could associate events from observing their occurrence sequences. Hypotheses will be proposed within the algorithm and be rejected if future observations contradict. Experiments have been conducted to evaluate its performance.

5.1 Introduction

Through evolution, human beings have developed advanced intelligence that is capable of learning various knowledge from the environment. The first type of learning is unsupervised learning that human beings can group outside world knowledge into clusters. This learning capability enables human to generalize observations and retrieve knowledge efficiently. The second type of learning is supervised learning that human beings can associate a new observation with learned categories. This capability enables human to extract category features to understand current observation. The third type of learning

is reinforcement learning [74], that human beings can learn action sequences to obtain optimal rewards. For example, after several failures of hunting herbivore animals, the human can gradually learn strategies to hunt them. The fourth type of learning is causal learning that human being can build up associations between events. By observing the preceding event, human beings can predict subsequent events and make preparation accordingly. For example, if a forest is on fire, animals will soon run out. Human beings can have a good chance of hunting. One important difference between reinforcement learning and causal learning is that the cost of reinforcement learning is very high that failures in learning may lead to fatal consequences. However, causal hypotheses are build up by using few observations which largely reduces the risks.

Bayesian reasoning and learning [75] has been used for causal learning in closed domains. In Bayesian learning, the probabilities of causal relationships between variables, or likelihoods, are known, which are obtained from statistical data, with hand-selection of relevant parameters. Then, together with a priori probabilities of the variables, the more likely causal relationship is identified. Similar to the issue of reinforcement, the Bayesian approach largely relies on statistical data, and this approach is more likely to be used by human to validate learned causal hypothesis in the long term. There is still a gap to learn hypothesis fast from the very small sample size.

On the other hand, in an open domain, multiple events are happening concurrently that intelligent systems may not be able to capture all of them and construct all possible associations. Because of this, attention is a precious resource for learning. The events related to life and death have a high value of interest. After learning the association between other events and these events of interest, those related events will be assigned some interest value and be listed on the attention list. For example, hunting herbivore animals solve food problem and thus hunting-related events, for example, forest on fire

will be leaned to attract attention. This attention allocation is another gap in causal learning in open domains.

In the psychology literature, when observing an event, people assume that things preceding the event cause it, and things following the event are effects of it [76] [77]. Inferences are made bi-directionally, and temporal cues demonstrate causality [78]. In other words, episodic memory plays an important role in associating events causal relationship. However, learning causal hypotheses quickly from the temporally related event from small sample size will probably lead to a high ratio of a false hypothesis. The rejection of false causal hypothesis is another important aspect of causal learning.

5.2 Hypothesis adaptive resonance theory (H-ART) architecture

Figure 5.1: H-ART architecture.

In this chapter, we propose an agent utilizing a three-layer hypothesis learning model H-ART (Hypothesis Adaptive Resonance Theory) shown in Figure 5.1, which is inspired by the psychological research that human being learns hypothesis using temporal cues, i.e., episodic memory. The H-ART model consists of three layers.

The first layer is the input layer having two input channels, namely the observed event information field and an interestingness field. The encoding of the observed event information is application dependant, and the input state S can be generally represented as

$$S_1^1 = (I_1^1, I_2^1, \dots, I_n^1) \quad (5.1)$$

where I_n^1 are features of state S . In a typical setting with spatiotemporal information of the events, the event information vector I_n contains the activity type, the temporal information, and the spatial information. The temporal information may contain time information and period (e.g., the day of the week) information. If the time is encoded using a normalized value over a day, the time will be represented continuously in H-ART. The spatial information may contain vectors representing locations and landmarks.

The interestingness field is used to indicate the interestingness associated with the input event. The interestingness vector contains a single element. In a typical input scenario, the interestingness value is either ‘1’ or ‘0’, with ‘1’ indicates the event is of interest and ‘0’ indicates no interestingness. However, throughout the learning process in H-ART, those events have no initial interestingness but hypothetically associated with events of interest will be assigned some interestingness value, and this field could be used to read out learned interestingness from H-ART.

The second layer contains a learned event field which stores the current episode, and a current event field which shares the same size of the event field with only the current event activated. Similar to EM-ART, the learn event field contains the learned event categories from the input, and the activation value indicates the sequence of events that the current event has an activation value of 1 while the older events are having activation value depreciated over time. Different from EM-ART which applied a gradient

depreciation, in H-ART a liner depreciation is applied over a fixed time depending on the application. If the application is to learn the hypothesis of events over a day, the decay rate is recommended to set to over 24 hours or 48 hours. If the application is about learning immediate cause and effect, the decay rate could be set to a smaller value. In other words, this decay rate is the parameter to determine how much time is allowed to pass to consider two events are causally related.

The current event vector S_2^2 is derived from S_2^1 and only contains the current event activated while other events elements have 0 or no activation value. This vector is used to represent the event of interest and associate the episode before it in the episodic memory.

The third layer is the episodic memory layer which contains a field associating events of interest and a generalized episode preceding it.

5.3 Hypothesis learning using H-ART

5.3.1 Current Episode Maintenance and Hypothesis Building

H-ART is an online architecture which takes input from an event stream. By using the F_2^1 learned event field, H-ART maintains a current episode. For every new input, the activated node in the F_2^1 layer will be assigned an activation value of 1, and at the same time, other previously observed events, i.e. having some activation value, will have their activation value decayed linearly. The algorithm is summarized in Algorithm 11.

With the proper maintenance of the current episode, when an event of interest is observed, the preceding episode which maintained the current episode will be associated with an event of interest and learned as a node at the F_3 layer. In a simple configuration, the vigilance parameters ρ for the current event F_2^2 is set to 1, and for the current episode, F_2^1 is set to 0 that each event of interest will be associated with only one episode. However, in a complex configuration, if the vigilance ρ for the current episode F_2^1 is not 0 which

Algorithm 11 H-ART Current Episode Maintenance

```

1: for each incoming event in the episode  $S$  do
2:   select a resonance node  $J$  in  $F_2^1$  based on input  $\mathbf{I}^1$  in  $F_1^1$ 
3:   let node activation  $y_j \leftarrow 1$  (or a predefined maximum value)
4:   for each previously selected node  $i$  in  $F_2$  do
5:     decay its activation by  $y_i^{(new)} = y_i^{(old)} - \frac{\Delta t}{T}$ , or 0 if  $y_i^{(old)} \leq 0$  where  $\Delta t$  is the time
       between current event and the previous event and  $T$  is the predefined observation
       window.
6:   end for
7: end for

```

means potentially several episodes clusters will be associated with the same event, i.e., multiple potential causes may lead to this event of interest.

When the event of interest is observed more than once, the new incoming current episode will be generalized with the episode stored in F_3 layer. If fast learning is applied, or $\beta = 1$, commonly shared events in the episodic memory will remain while other events will be cancelled out. Moreover, if compliment coding is applied to the episodic memory, the generalized remaining event activation and its complement will indicate a time interval meaning the preceding event is observed within a time interval before the event of interest is observed. A hypothesis is build using a threshold, say $n = 3$, that events have been observed n time before the event of interest is considered a potential cause. The more times the observations consistent with the hypothesis, the more confident that the preceding event is the cause of the event of interest.

The algorithm of hypothesis learning is summarized in Algorithm 12.

5.3.2 Null Hypothesis Rejection

When a new observed preceding episode of an event of interest does not contain the previously assumed cause event, the cause event activation will be automatically cancelled out during episode learning in Algorithm 12. However, there is no point to maintain an empty episode meaning nothing confirmed to cause the event of interest, and this empty

Algorithm 12 H-ART Hypothesis Building

```

1: for each incoming event in the episode  $S$  do
2:   Maintain the current episode using Algorithm 11
3:   if interestingness value  $i_1 \geq 0$  in input vector  $I_1^2$  then
4:     select the resonance node  $J$  in  $F_3$  based on current episode and the current
       event with  $\rho_2^1 = 0$  and  $\rho_2^2 = 1$ 
5:     learn its weight vector by  $w_j^{(new)} = x \wedge w_j^{(old)}$ , i.e.,  $\beta = 1$ 
6:     check the remaining events in the episode
7:     update the observation counter  $n$ 
8:     confirm a hypothesis when  $n$  passes a threshold e.g.,  $n \geq 2$ 
9:   end if
10: end for

```

node prevented learning new hypothesis. For these reasons, H-ART will drop nodes in F_3 layer when it represents an empty preceding episode of the event of interest. By dropping the node, it brings back the potential to reconstruct a hypothesis of the event of interest later when the environment changed.

5.3.3 Prediction of near future event

With the H-ART architecture and mechanisms described above, the agent could make predictions of the environment based on the hypothesis build using episodic memory. When a new event comes in, besides, to maintain the current episode, the agent also checks the F_3 learned episodic associations finding a potential prediction of an event of interest. The algorithm is summarized in Algorithm 13.

In real life, people may use single observations to build a hypothesis and make predictions for the significant event of interest. For example, if someone died during unusual weather, next time this rare weather comes people may feel terrible luck, and someone may die. This prediction is not rational, but it is a part of human nature. H-ART simulates this part of human hypothesis building. To avoid making pure random associations, H-ART has a threshold to the hypotheses that the hypothesis needs to be

Algorithm 13 H-ART Prediction Event of Interest

- 1: **for** each incoming event in the episode S **do**
 - 2: form a new vector S'_2 that follows the structure of S_2 of F_2^1
 - 3: select a resonance node in F_2^1 based on input I^1 in F_1^1
 - 4: Assign the activation value of current event in S'_2 to be 1 while other events to be 0
 - 5: If there is a resonance node J in F_3 based on input S'_2 with count n passes the threshold
 - 6: Readout the F_2^2 weighting associated with J
 - 7: Output the prediction
 - 8: **end for**
-

observed enough number of times for the agent to predict critical events based on prior knowledge confidently.

5.3.4 Interestingness learning

Attention allocation is an essential feature of H-ART. In Algorithm 12, only current episodes before the event of interest are learned. The hypothesis cause of the event of interest should inherit a certain level of interestingness. In H-ART, this is done by assigning a portion of the target event's interestingness to its hypothesis causes. A small portion of interestingness is shared when an event is in the episode preceding the event of interest, for example, 0.1 of the interestingness value. This is consistent with real-life experience that any event was happening just before a life and death event, even we consciously knowing it is not a cause of the result, we still give great attention when we encounter those events again, and episodic memory associated with it comes back at the same time. At the time when a hypothesis is learned, a large portion of the interestingness value will be assigned to the hypothesis cause, for example, 0.8 of the interestingness value. If we know something is highly related to some important events, its cause and effect will surely raise a high level of attention.

Figure 5.2: Causal Relationship Between Events in Simulation

5.4 Experiments on Synthetic Data Set

5.4.1 Synthetic data generation

To test H-ART, a simulation environment has been designed to generate a synthetic data stream. Inside the simulation environment, there are twenty types of events generated with a timestamp attached, namely event $A, B, C, \dots, T$. The time density of events is set to, on average, five minutes an event. Among all the events, seven events are generated following causal roles, as shown in Figure 5.2. Each arrow represents an independent cause and effect pair. The time between the cause and the effect events is set to between twenty minutes and an hour. The remaining thirteen events are generated randomly. More particularly, the causal relationship between A and B is generated periodically such that with a probability of 20% they will follow the causal relationship for the next 100 hours after the 100 hours expired. The generation will make a decision on the probability whether they will follow the causal relationship for another 100 hours. This is used to simulate temporally built causal behaviour.

5.4.2 The vigilance parameter on F_2^2

The first experiment is using the default configuration described in Section 5.3, and time window set to 1 hour, vigilance for F_2^2 set to 0, and all events are of interest. After 200 hours, e.g., about 2400 events, in simulation time. H-ART learned hypothesis of $G \rightarrow E$ and $D \rightarrow C$. By tracing back episodic memory operations, it is shown the cause E is cancelled with cause F in an episodic memory learning phase, and H-ART dropped the node. In the second trial, the vigilance ρ_2^2 for F_2^2 is set to 0.8 that allowing multiple episodes associated with same effect event. After 200 hours in simulation, H-ART now could learn the $G \rightarrow E$, $D \rightarrow C$, $E \rightarrow D$, and $F \rightarrow D$. However, if the vigilance parameter ρ_2^2 for F_2^2 is set too low, i.e., 0.4 or lower, the random events before the effect could not generalize and almost every event in the simulation is learned as a cause hypothesis of the effect.

5.4.3 The effect on the time window size

In the second experiment, H-ART model is tested with two time window settings: 1 hour and 4 hours. Vigilance for F_2^2 is set to 0.8 to enable learning multiple causes of events. After running H-ART 200 hours of simulation time, around 2400 events are generated, and by reading out the final F_3 layer nodes, the learned hypotheses are summarized in Table 5.4.3.

1 hour window		4 hours window	
Event	Cause	Event	Cause
A		A	
C	D	C	DEG or DF
D	E or F	D	EG or F
E	G	E	G

Table 5.1: Experiment on different window sizes

From the table we could see by using a larger time window, H-ART could learn hypotheses containing more events including the “cause of the cause”. However, in reality, the “cause of the cause” may not be a strong hypothesis for the final effect as each cause and effect pair is independent.

5.4.4 Prediction of future event

If a hypothesis could be observed enough times, i.e., pass a threshold n , H-ART considers it as an effective causal role that it may not be valid but in practice can be used for prediction. In this experiment, two thresholds: $n = 2$ and $n = 5$ have been tested to see how many predictions are made by H-ART and how accurate it could predict based on the simulation configuration. While other settings remain the same, ρ_2^2 is set to 0.8, the window size is set to 2 hours, and all events are of interest. The experiment is running on 500 hours of simulation time or about 6000 events. The result is shown in Table 5.4.4.

	$n = 2$	$n = 5$
Number of Predictions	1842	828
Accuracy	45%	91%

Table 5.2: Experiment on H-ART Prediction

With a smaller confidence threshold, H-ART will be more aggressive to make predictions of future events, while with a higher confidence threshold, H-ART will play safe. However, it is not the higher the confidence threshold is set, the better results in all applications, that the choice of confidence threshold largely depends on the application and the nature of the problem. In some scenarios, the error rate is tolerated, and the number of successful prediction is appreciated. In those cases, a smaller threshold is preferred. Moreover, for the temporal causal relationship $B \rightarrow A$ in the simulation, it only persists for a limited period and making predictions too hesitated will waste part of its presenting period.

5.4.5 Experiment on Interestingness Learning

In this experiment, the interestingness learning feature of H-ART is evaluated. The interestingness learning feature is used to identify potentially important features that have a link to the events of interest. In the default setting, the events in the episode preceding the event of interest will have a 0.1 share of its interestingness, and the events in the hypothesis of the event of interest will have a 0.8 share of its interestingness. The events A and C are set with the events of interest with an initial interestingness value one while others 0. $\rho_2^2 = 0.8$, and the window size is 2 hours. As hypothesis is learned through a confidence threshold, $n = 2$ and $n = 5$ are tested with the default interestingness learning policy. As shown in Table 5.4.5, with a smaller confidence threshold, H-ART is more aggressive in making a hypothesis, and this brings high interestingness to more event, which later is rejected as a cause. However, with a higher confidence threshold, H-ART will differentiate the events with interestingness from those random events in the stream.

	$n = 2$	$n = 5$
Average Interestingness	0.62	0.33
Standard Deviation	0.11	0.21

Table 5.3: Experiment on Interestingness Learning

5.5 Experiments on Real-life Data Set

5.5.1 CASAS Data Set

Empirical experiments on the public CASAS data set [60] were conducted to validate the performance of the hypothesis prediction Agent. The 17th data set was chosen for the following experiment. The data set, collected from the testbed in Aruba, contains

the sensory readings of an elderly female tenant over a period of 220 days. The data collected are primarily sensory readings with annotated ADLs as ground truths.

The experiments are conducted at two levels. In the first level, the 1.7 million sensory readings from 26 sensors are used to detect a potential causal relationship between events. In the second level, the annotated 6468 ADLs shown in Table 5.4 are used for testing.

ADL	Count
Meal_Preparation	1606
Relax	2910
Eating	257
Working	171
Sleeping	401
Washing_Dishes	65
Bed_to_Toilet	157
Enter_Home	431
Leave_Home	431
Housekeeping	33
Resperate	6

Table 5.4: Annotated ADLs in the CASAS data set

Some of the sensors are binary sensors that they only indicate the on and off states while some other sensors are scalar sensors that they have a reading. Considering the feature size of the 26 sensors, the different readings of the same sensor are encoded to different events in the H-ART. For example, T001 (temperature) reading 24 is a different event from T001 reading 30.

5.5.2 Hypothesis Learning on Sensory Data

In the first experiment, the interestingness feature is disabled by giving all events the interestingness value of one. From the 1.7 million sensory readings, the agent managed to construct 8123 hypothesis, which is observed more than twice during the observation process. Within the 8123 hypothesis, 5301 of them are dropped later due to contradiction to

new observations. The remaining 2822 hypothesis are considered learnt hypothesis and within them 131 hypotheses that observed more than five times. These 131 hypotheses observed more than five times are considered the strong hypothesis. The result is summarized in Table 5.5.

ADL	Count
Total Hypothesis	8123
Dropped Hypothesis	5310
Learnt Hypothesis	2822
Strong Hypothesis	131

Table 5.5: Leant Hypothesis on Sensory Data

Looking at the 131 strong hypotheses, it can be seen that they are mainly the routines of certain ADL. For example, cooking consists of open the kitchen door, turn on the kitchen light, and turn on the stove, and the temperature in the kitchen raises.

5.5.3 Hypothesis learning on ADL Data

In the second experiment, the annotated ADLs are used for hypothesis learning. After observing the 6468 ADLs at one go, the agent constructed 212 hypotheses during the process. Within them, 140 hypotheses are dropped, and only 9 of them are strong hypotheses which have five or more observations. The result is summarized in Table 5.6.

ADL	Count
Total Hypothesis	212
Dropped Hypothesis	140
Learnt Hypothesis	72
Strong Hypothesis	9

Table 5.6: Leant Hypothesis on ADL Data

Having a close look at the dropped hypothesis, especially those dropped with more than three observations, we can see the monitored user has a change of behaviour across

the days. In early months the user sometimes went out in the noon and cooked after that. This behaviour was dropped later on, and the user adopted a new behaviour to cook several times in the afternoon and eat at home. And later this behaviour is dropped as well. The experiment shows with the help of H-ART, the agent is also able to learn the temporal behaviour patterns, which only last a period of time and disappear.

5.6 Summary

In this chapter, we have proposed an agent with an online hypothesis learning neural network named Hypothesis Adaptive Resonance Theory, or H-ART. H-ART can learn hypotheses quickly from an online data stream. Through the experiments on a synthetic data set and a real-life data set, the performance of H-ART is evaluated.

There are still several issues in the current H-ART architecture. Firstly, there are several configuration choices of H-ART, and all will affect the performance. Choosing a proper set of configuration for different applications is a challenge. In future work, an automatic parameter adjustment mechanism is preferred. Secondly, in the current design, the interestingness value is not fully used in the H-ART. The interestingness value could be linked to the confidence threshold, that in the case of very unfavoured events, i.e., death, the confidence threshold will be set low, for example, $n = 2$, to avoid the hypothesis cause of such events. On the other hand, for favoured events, H-ART can be more patient by having a higher confidence threshold to make predictions reliable. The third issue is the encoding of the events. In a typical setting, the input vector has the size of the events, and only the observed event has an activation value of 1. This is the same in the F_2^1 field. Therefore, the H-ART only recognizes the most recent observed event in the time window, and the information of the previous observations are lost. Different event encoding schemes may be proposed to avoid this situation.

Chapter 6

Conclusion

6.1 Summary of Contributions

ADLs are commonly used in the healthcare domain to measure the functional status of individual person. Given the importance of ADLs as wellness indicators, research in ADL detection and recognition has been very active in recent years. However, there have been limited studies on ADL pattern modeling and analysis. This dissertation has identified the following challenges and provided modeling attempts to address these three challenges.

The first challenge is to model the activity routines in memory models. To our best knowledge, there is no existing literature that touched this particular problem. This dissertation proposes the human-inspired multi-memory neural network, ADLART, utilizing an episodic memory neural network. In ADLART, both episodic memory and semantic memory of activities routines are learned and maintained. An application of ADLART named RADLART is developed to provide recommendations to elderly people. Extensive experiments are conducted on both synthetic and real-world data sets. The results prove the performance of these models.

The second challenge is to improve modeling by addressing spatiotemporal dimensions. This dissertation proposed a neural network named STADLART to address this

challenge. The STADLART model introduces a spatiotemporal ADL layer in addition to the original ADLART making STADLART able to generalize activities according to spatial and temporal features before generalizing the ADL routines. Additionally, to address the observation that some activities are more frequently performed than others, normalization and customization algorithms are introduced. Besides STADLART, which focus on the ADL domain, a generalized model named ST-ART is proposed providing an additional temporal field to generalize temporal information before the spatiotemporal events. Experiments validate the effectiveness of these models.

The third challenge is finding causal relationships among activities for prediction with the consideration that human behaviour may change over time. While many existing works are using a statistical approach, this thesis uses a neural network approach which performs hypothesis testing on episodic memory. Based on the effective causal relationship theory, an online causal learning neural network named H-ART is proposed. H-ART detects event associations and continuously maintains a set of hypotheses from observations on repeated associations. Considering users may change their habits, H-ART continuously checks the learnt hypotheses against new observations and drops hypotheses when contradicts are found. Experiments validate the model, and the results are promising.

The contributions of this dissertation can also be viewed as technique and application aspects. In terms of technique contributions, this dissertation has proposed novel models learning different ADL information, including ADL routines, ADL type, temporal information, spatial information, and generalized ADL groups. There is no known work in learning ADL routines. In this work, ADL routines are generalized by comparing users' ADLs sequences. In addition, in this dissertation, ADL routines are modeled in human-inspired memory forms, which provides efficient encoding and retrieving. Comparing to

deep models, the interpretation of learnt models in this work are largely human-readable. Furthermore, to overcome the difficulty of encoding temporal information, the dissertation proposes encoding schemes of the time, day of the week, as well as generalized cross-day time slots. This dissertation also provided a modeling approach for causal relationship and hypothesis testing. The algorithm can record hypothesis after observing several occurrence and can reject hypothesis when contradiction observed.

For potential applications, all the proposed models and algorithms are experimented with both synthetic and real-world data sets. The promising results show the capability of future applications of the models in real-world. Different from other deep neural networks, the fusion ART based neural networks are capable of working with online data and small sample size. The RADLART is proposed for ADL recommendation application. Application issues are also addressed for the STADLART by introducing the ADL weight normalization and customization algorithms. Finally, these models can work with scenarios which have little or no training data. For example, there is no training phase needed for the H-ART to start observing event associations and learning hypotheses.

6.2 Future Work

Although this thesis has proposed models to address the problems in human activity modeling, there are still limitations of the techniques which deserve future research.

6.2.1 Automatic Vigilance Value Assignment

The proposed neural network models are all fusion ART based models. For these models, the vigilance parameter is vital in controlling the generalization behaviour in each layer. In the current practice, a range of vigilance parameter values are tested empirically for

each layer, and the best-performing vigilance parameter value is chosen for subsequent experiments. It would be helpful to develop algorithms to automatically assign vigilance values to different layers.

6.2.2 Sophisticated Spatial Field Design

The STADLART and ST-ART models have sophisticated modeling on temporal information which covers detailed temporal aspects including starting time, ending time, day of the week, and cross-day time slots. However, the current modeling of spatial information relatively remains elementary. Specifically, the spatial information currently only represents the room ID while other information, e.g., the neighbouring information between rooms, are not captured. Some activities may be captured in a bigger area while other activities may be captured in its sub-areas. A neural representation of the spatial containing relationship is very interesting to be added in future extensions.

6.2.3 Imperfect Observations

The proposed models currently work on perfect observations based on the assumption that all events must be captured. For ADLART and STADLART, if some ADLs are not captured, the models are still able to identify a neighbouring cluster in certain extent. However, for the H-ART, if an event in a known hypothesis is not captured, the hypothesis will contradict with the current observation and will be deleted from the knowledge base. The perfect observation assumption may not be valid in all applications. For the causal learning model to work in an imperfect observation setting, some probability-based algorithms may be needed. This is another interesting direction to be looked at.

Author's Publications

- (i) Shan Gao, Ah-Hwee Tan “User Activity Pattern Learning - A multi-memory Modeling Approach”, International Joint Conference on Neural Networks (IJCNN) 2014.
- (ii) Shan Gao, Di Wang, Ah-Hwee Tan “Progressive Sequence Matching for ADL Plan Recommendation”, IEEE/WIC/ACM Intelligent Agent Technology Conference (IAT) 2015.
- (iii) Shan Gao, Ah-Hwee Tan “An Autonomous Agent for Learning Spatiotemporal Models of Human Daily Activities”, Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2016: 1453-1454.
- (iv) Shan Gao, Ah-Hwee Tan “Learning routines of activities of daily living with spatiotemporal neural networks”, The 5th International Conference on Ageless Aging (ICAA) 2017.
- (v) Shan Gao, Ah-Hwee Tan, Rossi Setchi “Learning ADL Daily Routines with Spatiotemporal Neural Networks”, IEEE Transactions on Knowledge and Data Engineering 2019, DOI: 10.1109/TKDE.2019.2924623

References

- [1] WHO, “10 facts on ageing and the life course,” Apr. 2012.
- [2] F. J. S. Salces, M. Baskett, D. Llewellyn-Jones, and D. England, “Ambient interfaces for elderly people at home,” in *Ambient Intelligence in Everyday Life*, pp. 256–284, Springer, 2006.
- [3] D. Low, *Behavioural economics and policy design: Examples from Singapore*. World Scientific, 2011.
- [4] H. Hasson and J. E. Arnetz, “Nursing staff competence, work strain, stress and satisfaction in elderly care: a comparison of home-based care and nursing homes,” *Journal of clinical nursing*, vol. 17, no. 4, pp. 468–481, 2008.
- [5] H. K. Butcher, P. A. Holkup, M. Park, and M. Maas, “Thematic analysis of the experience of making a decision to place a family member with alzheimer’s disease in a special care unit,” *Research in Nursing & Health*, vol. 24, no. 6, pp. 470–480, 2001.
- [6] T. Fahey, A. A. Montgomery, J. Barnes, and J. Protheroe, “Quality of care for elderly residents in nursing homes and elderly people living at home: controlled observational study,” *Bmj*, vol. 326, no. 7389, p. 580, 2003.
- [7] R. Harper, *Inside the smart home*. Springer Science & Business Media, 2006.

REFERENCES

- [8] L. Jiang, D.-Y. Liu, and B. Yang, “Smart home research,” in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, vol. 2, pp. 659–663, IEEE, 2004.
- [9] E. Aarts and R. Wichert, “Ambient intelligence,” in *Technology guide*, pp. 244–249, Springer, 2009.
- [10] E. Tulving *et al.*, “Episodic and semantic memory,” *Organization of memory*, vol. 1, pp. 381–403, 1972.
- [11] D. Lymberopoulos, A. Bamis, and A. Savvides, “Extracting spatiotemporal human activity patterns in assisted living using a home sensor network,” *Universal Access in the Information Society*, vol. 10, no. 2, pp. 125–138, 2011.
- [12] J. Soulas, P. Lenca, and A. Thépaut, “Monitoring the habits of elderly people through data mining from home automation devices data,” in *Portuguese Conference on Artificial Intelligence*, pp. 343–354, Springer, 2013.
- [13] E. Heierman, M. Youngblood, and D. J. Cook, “Mining temporal sequences to discover interesting patterns,” in *KDD Workshop on mining temporal and sequential data*, pp. 104–111, Citeseer, 2004.
- [14] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas, “Collecting and disseminating smart home sensor data in the casas project,” in *Proceedings of the CHI workshop on developing shared home behavior datasets to advance HCI and ubiquitous computing research*, pp. 1–7, 2009.
- [15] S.-B. Ho, “On effective causal learning,” in *International Conference on Artificial General Intelligence*, pp. 43–52, Springer, 2014.

REFERENCES

- [16] J. Pearl, *Causality*. Cambridge university press, 2009.
- [17] W. Wang, B. Subagdja, A.-H. Tan, and J. A. Starzyk, “Neural modeling of episodic memory: Encoding, retrieval, and forgetting,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, 2012.
- [18] A.-H. Tan, G. A. Carpenter, and S. Grossberg, “Intelligence through interaction: Towards a unified theory for learning,” in *Advances in Neural Networks–ISNN 2007*, pp. 1094–1103, Springer, 2007.
- [19] S. Gao and A.-H. Tan, “User daily activity pattern learning: A multi-memory modeling approach,” in *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 1542–1548, IEEE, 2014.
- [20] S. Gao, D. Wang, A.-H. Tan, and C. Miao, “Progressive sequence matching for ADL plan recommendation,” in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 2, pp. 360–367, IEEE, 2015.
- [21] G. Shan and A.-H. Tan, “Learning routines of activities of daily living with spatiotemporal neural networks,” in *The 5th International Conference on Ageless Aging(ICAAG)*, 2017.
- [22] S. Gao, A.-H. Tan, and R. Setchi, “Learning ADL daily routines with spatiotemporal neural networks,” *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [23] S. Gao and A.-H. Tan, “An autonomous agent for learning spatiotemporal models of human daily activities,” in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 1453–1454, 2016.

REFERENCES

- [24] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” *IEEE Transactions on Circuits and Systems for Video technology*, vol. 18, no. 11, p. 1473, 2008.
- [25] J. Aggarwal and M. S. Ryoo, “Human activity analysis: A review,” *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.
- [26] S. J. Preece, J. Y. Goulermas, L. P. Kenney, D. Howard, K. Meijer, and R. Crompton, “Activity identification using body-mounted sensors—a review of classification techniques,” *Physiological measurement*, vol. 30, no. 4, p. R1, 2009.
- [27] X. Long, B. Yin, and R. M. Aarts, “Single-accelerometer-based daily physical activity classification,” in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 6107–6110, IEEE, 2009.
- [28] G. Yin and D. Bruckner, “Daily activity model for ambient assisted living,” in *Technological Innovation for Sustainability*, pp. 197–204, Springer, 2011.
- [29] L. Li and Z. B. Zabinsky, “Incorporating uncertainty into a supplier selection problem,” *International Journal of Production Economics*, vol. 134, no. 2, pp. 344–356, 2011.
- [30] M. Zhang and A. A. Sawchuk, “A bag-of-features-based framework for human activity representation and recognition,” in *Proceedings of the 2011 international workshop on Situation activity & goal awareness*, pp. 51–56, ACM, 2011.
- [31] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, “Activity recognition and abnormality detection with the switching hidden semi-markov model,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, vol. 1, pp. 838–845, IEEE, 2005.

REFERENCES

- [32] A. Fleury, M. Vacher, and N. Noury, “Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.
- [33] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, “Ontology-enabled activity learning and model evolution in smart homes,” in *International Conference on Ubiquitous Intelligence and Computing*, pp. 67–82, Springer, 2010.
- [34] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [35] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.
- [36] T. Huynh, M. Fritz, and B. Schiele, “Discovery of activity patterns using topic models,” in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 10–19, ACM, 2008.
- [37] E. Kim, S. Helal, and D. Cook, “Human activity recognition and pattern discovery,” *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48–53, 2010.
- [38] U. Blanke and B. Schiele, “Daily routine recognition through activity spotting,” in *Location and Context Awareness*, pp. 192–206, Springer, 2009.
- [39] P. Spirtes, C. N. Glymour, R. Scheines, D. Heckerman, C. Meek, G. Cooper, and T. Richardson, *Causation, prediction, and search*. MIT press, 2000.

REFERENCES

- [40] H. Lu, A. Yuille, M. Lijieholm, P. W. Cheng, and K. J. Holyoak, “Modeling causal learning using bayesian generic priors on generative and preventive powers,” 2011.
- [41] L. Michael, “Causal learnability,” in *22nd International Joint Conference on Artificial Intelligence*, pp. 1014–1020, 2011.
- [42] M. Harradon, J. Druce, and B. Ruttenberg, “Causal learning and explanation of deep neural networks via autoencoded activations,” *arXiv preprint arXiv:1802.00541*, 2018.
- [43] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, “Causal effect inference with deep latent-variable models,” in *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.
- [44] G. A. Carpenter and S. Grossberg, *Adaptive resonance theory*. Springer, 2010.
- [45] L. E. B. da Silva, I. Elnabarawy, and D. C. Wunsch II, “A survey of adaptive resonance theory neural network models for engineering applications,” *Neural Networks*, vol. 120, pp. 167–203, 2019.
- [46] G. A. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” *Computer vision, graphics, and image processing*, vol. 37, no. 1, pp. 54–115, 1987.
- [47] G. A. Carpenter and S. Grossberg, “Art 2: Self-organization of stable category recognition codes for analog input patterns,” *Applied optics*, vol. 26, no. 23, pp. 4919–4930, 1987.
- [48] G. A. Carpenter, S. Grossberg, and D. B. Rosen, “Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system,” *Neural networks*, vol. 4, no. 6, pp. 759–771, 1991.

REFERENCES

- [49] G. A. Carpenter, *Distributed activation, search, and learning by ART and ARTMAP neural networks*. Boston University, Center for Adaptive Systems and Department of Cognitive ..., 1996.
- [50] J. R. Williamson, “Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps,” *Neural networks*, vol. 9, no. 5, pp. 881–897, 1996.
- [51] H. Isawa, M. Tomita, H. Matsushita, and Y. Nishio, “Fuzzy adaptive resonance theory with group learning and its applications,” in *Proc. of International Symposium on Nonlinear Theory and its Applications*, pp. 292–295, 2007.
- [52] L. Massey, “Discovery of hierarchical thematic structure in text collections with adaptive resonance theory,” *Neural Computing and Applications*, vol. 18, no. 3, pp. 261–273, 2009.
- [53] T.-H. Teng, A.-H. Tan, and J. M. Zurada, “Self-organizing neural networks integrating domain knowledge and reinforcement learning,” *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 889–902, 2014.
- [54] A.-H. Tan, B. Subagdja, D. Wang, and L. Meng, “Self-organizing neural networks for universal learning and multimodal memory encoding,” *Neural Networks*, vol. 120, pp. 58–73, 2019.
- [55] A.-H. Tan and H.-S. V. Soon, “Concept hierarchy memory model: A neural architecture for conceptual knowledge representation, learning, and commonsense reasoning,” *International Journal of Neural Systems*, vol. 7, no. 3, p. 305, 1996.

REFERENCES

- [56] A. L. Stewart, K. M. Mills, A. C. King, W. L. Haskell, D. Gillis, and P. L. Ritter, “Champs physical activity questionnaire for older adults: outcomes for interventions,” *Medicine & Science in Sports & Exercise*, 2001.
- [57] C. S. McCrae, J. P. McNAMARA, M. A. Rowe, J. M. Dzierzewski, J. Dirk, M. Marsiske, and J. G. Craggs, “Sleep and affect in older adults: using multilevel modeling to examine daily associations,” *Journal of sleep research*, vol. 17, no. 1, pp. 42–53, 2008.
- [58] D. Cook and M. Schmitter-Edgecombe, “Assessing the quality of activities in a smart environment,” *Methods of information in medicine*, vol. 48, no. 5, p. 480, 2009.
- [59] R. B. Cialdini, “Influence: The psychology of persuasion,” 1993.
- [60] D. J. Cook, “Learning setting-generalized activity models for smart spaces,” *IEEE intelligent systems*, vol. 2010, no. 99, p. 1, 2010.
- [61] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [62] J. Hu, H. Tang, K. C. Tan, and H. Li, “How the brain formulates memory: A spatio-temporal model research frontier,” *IEEE Computational Intelligence Magazine*, vol. 11, no. 2, pp. 56–68, 2016.
- [63] D. S. Modha and W. S. Spangler, “Feature weighting in k-means clustering,” *Machine learning*, vol. 52, no. 3, pp. 217–237, 2003.
- [64] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, “Automated variable weighting in k-means type clustering,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 657–668, 2005.

REFERENCES

- [65] C.-C. Hung, S. Kulkarni, and B.-C. Kuo, "A new weighted fuzzy c-means clustering algorithm for remotely sensed image classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 543–553, 2010.
- [66] J. Li, X. Gao, and L. Jiao, "A new feature weighted fuzzy clustering algorithm," in *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pp. 412–420, Springer, 2005.
- [67] D. Charalampidis, G. C. Anagnostopoulos, M. Georgiopoulos, and T. Kasparis, "Fuzzy ART and fuzzy ARTMAP with adaptively weighted distances," in *Applications and Science of Computational Intelligence V*, vol. 4739, pp. 86–97, International Society for Optics and Photonics, 2002.
- [68] R. Andonie, L. M. Sasu, and A. Cațaron, "A novel fuzzy ARTMAP architecture with adaptive featureweights based on onicescu's informational energy," *International Journal of Computers Communications & Control*, vol. 4, no. 2, pp. 104–117, 2009.
- [69] S. Roley, J. DeLany, C. Barrows, S. Brownrigg, D. Honaker, D. Sava, V. Talley, K. Voelkerding, D. Amini, E. Smith, *et al.*, "Occupational therapy practice framework: Domain & practice," *The American journal of occupational therapy.: official publication of the American Occupational Therapy Association*, vol. 62, no. 6, p. 625, 2008.
- [70] M. Lawton and E. Brody, "Assessment of older people: Self-maintaining and instrumental activities of daily living.," *The Gerontologist*, vol. 9, no. 3, pp. 179–186, 1969.
- [71] R. Bucks, D. Ashworth, G. Wilcock, and K. Siegfried, "Assessment of activities of daily living in dementia: Development of the bristol activities of daily living scale," *Age and Ageing*, vol. 25, no. 2, pp. 113–120, 1996.

REFERENCES

- [72] “Deep learning for java.” <https://deeplearning4j.org/lstm.html>. Accessed: 2018-05-01.
- [73] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.
- [74] R. S. Sutton, “Reinforcement learning: Past, present and future,” in *Asia-Pacific Conference on Simulated Evolution and Learning*, pp. 195–197, Springer, 1998.
- [75] J. Pearl, *Causality: models, reasoning and inference*, vol. 29. Springer, 2000.
- [76] P. W. Cheng and L. R. Novick, “A probabilistic contrast model of causal induction,” *Journal of personality and social psychology*, vol. 58, no. 4, p. 545, 1990.
- [77] S. Sloman, *Causal models: How people think about the world and its alternatives*. Oxford University Press, 2005.
- [78] D. A. Lagnado and S. A. Sloman, “Time as a guide to cause,” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 32, no. 3, p. 451, 2006.
- [79] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine,” in *Ambient Assisted Living and Home Care*, pp. 216–223, Springer, 2012.
- [80] R. B. Saltman, H. F. Dubois, and M. Chawla, “The impact of aging on long-term care in europe and some potential policy responses,” *International Journal of Health Services*, vol. 36, no. 4, pp. 719–746, 2006.
- [81] M. A. Sanz-Bobi, D. Contreras, and Á. Sánchez, “Multi-agent systems orientated to assist with daily activities in the homes of elderly and disabled people,” in *Human-Computer Interaction: The Agency Perspective*, pp. 131–167, Springer, 2012.

REFERENCES

- [82] J. C. Augusto, H. Nakashima, and H. Aghajan, “Ambient intelligence and smart environments: A state of the art,” in *Handbook of Ambient Intelligence and Smart Environments*, pp. 3–31, Springer, 2010.
- [83] S. Bahadori, A. Cesta, L. Iocchi, G. Leone, D. Nardi, F. Pecora, R. Rasconi, and L. Scozzafava, “Towards ambient intelligence for the domestic care of the elderly,” in *Ambient Intelligence*, pp. 15–38, Springer, 2005.
- [84] L. Zaad and S. B. Allouch, “The influence of control on the acceptance of ambient intelligence by elderly people: an explorative study,” in *Ambient Intelligence*, pp. 58–74, Springer, 2008.