A Unified Recommendation Framework for Data-driven, People-centric Smart Home Applications

Author:
Altulyan, May

Publication Date:
2022

DOI:
https://doi.org/10.26190/unsworks/24371

A Unified Recommendation Framework for Data-driven, People-centric Smart Home Applications

May Altulyan

A thesis in fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

February 2022

Thesis/Dissertation Sheet

Surname/Family Name	:	**May**
Given Name/s	:	**Altulyan**
Abbreviation for degree as give in the University calendar	:	**PhD**
Faculty	:	**Engineering**
School	:	**Computer Science and Engineering**
Thesis Title	:	**A Unified Recommendation Framework for Data-driven, People-centric Smart Home Applications**

Abstract 350 words maximum: (PLEASE TYPE)

With the rapid growth in the number of things that can be connected to the Internet, Recommendation Systems for the IoT (RSIoT) have become more significant in helping a variety of applications to meet user preferences, and such applications can be smart home, smart tourism, smart parking, m-health and so on. In this thesis, we propose a unified recommendation framework for data-driven, people-centric smart home applications. The framework involves three main stages: complex activity detection, constructing recommendations in timely manner, and insuring the data integrity.

First, we review the latest state-of-the-art recommendations methods and development of applications for recommender system in the IoT so, as to form an overview of the current research progress. Challenges of using IoT for recommendation systems are introduced and explained. A reference framework to compare the existing studies and guide future research and practices is provided.
In order to meet the requirements of complex activity detection that helps our system to understand what activity or activities our user is undertaking in relatively high level. We provide adequate resources to be fit for the recommender system. Furthermore, we consider two inherent challenges of RSIoT, that is, capturing dynamicity patterns of human activities and system update without a focus on user
feedback.
Based on these, we design a Reminder Care System (RCS) which harnesses the advantages of deep reinforcement learning (DQN) to further address these challenges. Then we utilize a contextual bandit approach for improving the quality of recommendations by considering the context as an input. We aim to address not only the two previous challenges of RSIoT but also to learn the best action in different scenarios and treat each state independently. Last but not least, we utilize a blockchain technology to ensure the safety of data storage in addition to decentralized feature.

In the last part, we discuss a few open issues and provide some insights for future directions.

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed ………………………………………………….............

Date …… 13-2-2022…………………………..............

INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in their thesis in lieu of a Chapter if:

- The student contributed greater than 50% of the content in the publication and is the "primary author", ie. the student was responsible primarily for the planning, execution and preparation of the work for publication
- The student has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis

Please indicate whether this thesis contains published material or not.

☐ *This thesis contains no publications, either published or submitted for publication (if this box is checked, you may delete all the material on page 2)*

☒ *Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement (if this box is checked, you may delete all the material on page 2)*

☐ *This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below*

CANDIDATE'S DECLARATION

I declare that:

- I have complied with the Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Name	**Signature**	**Date (dd/mm/yy)**
May Altulyan		13/02/2022

Abstract

With the rapid growth in the number of things that can be connected to the internet, Recommendation Systems for the IoT (RSIoT) have become more significant in helping a variety of applications to meet user preferences, and such applications can be smart home, smart tourism, smart parking, m-health and so on. In this thesis, we propose a unified recommendation framework for data-driven, people-centric smart home applications. The framework involves three main stages: complex activity detection, constructing recommendations in timely manner, and insuring the data integrity.

First, we review the latest state-of-the-art recommendations methods and development of applications for recommender system in the IoT so, as to form an overview of the current research progress. Challenges of using IoT for recommendation systems are introduced and explained. A reference framework to compare the existing studies and guide future research and practices is provided.

In order to meet the requirements of complex activity detection that helps our system to understand what activity or activities our user is undertaking in relatively high level. We provide adequate resources to be fit for the recommender system.

Furthermore, we consider two inherent challenges of RSIoT, that is, capturing dynamicity patterns of human activities and system update without a focus on user feedback. Based on these, we design a Reminder Care System (RCS) which harnesses the advantages of deep reinforcement learning (DQN) to further address these challenges.

Then we utilize a contextual bandit approach for improving the quality of recommendations by considering the context as an input. We aim to address not only the two previous challenges of RSIoT but also to learn the best action in different scenarios and treat each state independently.

Last but not least, we utilize a blockchain technology to ensure the safety of data storage in addition to decentralized feature.

In the last part, we discuss a few open issues and provide some insights for future directions.

Acknowledgements

My PhD journey has truly been a life-changing experience. It would have been difficult for me to cope with the challenges and demands of my PhD programme without the reliable support from my supervisors, colleagues, friends, and family.

First, I am deeply grateful to my supervisor Prof. Lina Yao for her indispensable support all through my PhD study. It is hard for me to describe her role during these years in few lines but with Lina, I found a professional, fun, enthusiastic, and lively environment to grow and develop as a researcher. She motivates me always to be the best version of myself. Lina is not just a supervisor but also a wonderful mentor who gives inspiration and brilliant advise to be taken for the rest of your life.

I would also like to thank my joint supervisor Prof. Salil Kanhere for his professional help which contributed significantly to the quality of my research output. I also would like to thank my secondary supervisor Dr. Chaoran for all his help and tutelage during this journey.

Many thanks too to my colleagues, Chuka Oham, Lei Bai, Xiang Zhang, Xuesong Wang, Uzma Mahroof, and Changwei Zou for their unreserved support and the pleasing moments we shared.

Thankfulness to my three little kids who surround me with their love especially my little boy Aziz who gives me ambition and motivation to continue. Thanks to my parents too for believing in me, for their continuous care and support most especially while writing this thesis.

Furthermore, I would like to dedicate this dissertation to Prof. Amal Zaki who passed away since the beginning of my journey. Her passing was quite difficult to take however, she reposed in me an infinity of love and courage that will remain with me forever.

Finally, I extend my gratitude to the Prince Sattam bin Abdulaziz University in

Saudi Arabia along with the Saudi Arabian Cultural Mission in Australia for their support during the period of my scholarship, which helped me achieve my goals toward a successful career.

Abbreviations

IoT Internet of Things

RSIoT Recommender System of the Internet of Things

RS Recommender System

CF Collaborative Filtering

PNN Probabilistic Neural Networks

PDF Probability Density Function

MOGS Member Organization-based Group Similarity

OBCF Object-Based Collaborative Filtering Answering

UBCF User-Based Collaborative Filtering

POI Points of Interest

CB Content Based

(SEQREQ Sequence-based Recommendation

CONFREQ) Recommendations for Configurators

DIAGREQ Recommending Diagnoses

HB Hybrid Based

DS Digital Signage

CSL Cognitive Specification Language

KB Knowledge Based

WoO Web of Objects

VO Virtual Objects

CVO Composite Virtual Objects

(CAMS Context-Aware Management System

CoaaS Context-as-a-Service

CSDL Context Service Description Language

CSM Context Service Matchmaking

HSCT Hierarchical Social Contextual Tree

SIoT social IoT

S/I-M Service/Interest Matching

S/CT-P) Service/Connection Time-Prediction

MAS Multi Agent System

MRS Mobility Recommender System

MQE Mobility Query Engine

RCP Route Calculation Planner

ML Machine Learning

AI Artificial Intelligence

OSR Optimal State-based Recommender

SGD Stochastic Gradient Descent

ALS Alternating Least Square

RBF Radial Basis Function

DS Dempster-Shafer

UPR Universal Profiling and Recommendation

AHP Analytic Hierarchy Process

SVM Support Vector Machine

fp-growth Frequent Pattern Growth

DL Deep Learning

MLP Multi Layer Perceptron

AE Autoencoder

CNN Convolutional Neural Network

RNN Recurrent Neural Network

RBM Restricted Boltzmann Machine

LinUCB Upper Confidence Bounds algorithm

NFC Near Field Communication

RFID Radio Frequency Identification

MQTT Message Queue Telemetry Transport

CoAP Constrained Application Protocol

6LowPAN Low Power Wireless Personal Area Network

HMM Hidden Markov Model

CRF Conditional Random Field

NMF Non-Negative Matrix Factorization

MAB Multi Armed Bandit

QoS Quality Of Service

BC Blockchain

AD Alzheimer's disease

RCS Reminder Care System

ADLs Activities of Daily Living

HAR Human Activity Recognition

IMU Inertial Measurement Units

OWL Ontology Web Language

DRL Deep Reinforcement Learning

DQN Deep Q-Network

DEER Deep Recommender System

MSE Mean Squared Error

Contents

1 Introduction **1**

1.1 Overview of Recommender System and Internet of Things 3

1.2 Challenges in RSIoT 4

1.3 Conclusion 8

1.4 Dissertation Organization 8

2 Recommender system for the internet of Things Techniques and Applications: Literature Review **11**

2.1 Introduction 12

2.2 Recommender Techniques and Systems for the IoT: State of the Art . 13

2.2.1 Collaborative filtering approach 14

2.2.2 Content-based approach 21

2.2.3 Hybrid approach 23

2.2.4 Knowledge-based approach 25

2.2.5 Context-aware Recommendation 27

2.2.6 Social IoT-based Recommendation Technique 30

2.2.7 Multi-Agent Algorithms 32

2.2.8 Recommendations with Graph Database Model 36

2.2.9 Recommendations with Machine learning 38

2.2.10 Recommendations with Deep Learning 40

2.2.11 Recommendations with Reinforcement Learning 42

2.3 Applications of RSIoT 43

2.3.1 Smart m-health 43

2.3.2 Smart Car-parking 45

2.3.3 Smart Tourism 48

2.3.4 Personal Recommender System Applications 49

2.3.5 Social Recommendations 52

2.3.6 Smart Homes 53

2.3.7 Smart Marketing 55

2.4 Analysis and Findings 55

2.5 A Unified Recommender System Framework for the IoT 59

2.5.1 Data acquisition platform 59

2.5.2 Data processing for rich information 61

2.5.3 Event generator and Rule composer 62

2.5.4 Accurate recommendations production 64

2.5.5 Privacy and Security in RSIoT 65

2.6 Conclusion 73

3 Reminder Care System (RCS): An Activity-aware Cross-device Recommendation System 74

3.1 Introduction 75

3.2 Related work 77

3.2.1 Smart home applications 77

3.2.2 Deep learning for human activity recognition 78

3.3 Reminder Recommender Care System 80

3.3.1 Correlation Analysis of Devices 80

3.3.2 Rule-Based Orchestration 85

3.3.3 Activity-Triggered Reminder Recommendation 86

3.4 Evaluation 88

3.4.1 Dataset 89

3.4.2 Elementary Activity Recognition 91

3.4.3 Complexity Activity Recognition 91

3.5 Conclusion 92

4 Deep Reinforcement Learning for Dynamic Things of Interest Recommendation in Intelligent Ambient Environment 94

4.1 Introduction 95

4.2 Deep Reinforcement Learning in RSIoT 97

4.3 Background 98

4.4 Reminder Care System (RCS) framework 100

4.4.1 Notation 100

4.4.2 Overview 101

4.4.3 Deep Q network (DQN)for recommendation 103

4.5 Experiment 105

4.5.1 Datasets 106

4.5.2 Experiments results 109

4.6 Conclusion 111

5 Context-induced Activity Monitoring for On-demand Things-of-Interest Recommendation in an Ambient Intelligent Environment 112

5.1 Introduction 114

5.2 Contextual bandit (CB) for recommendations 117

5.3 Prompt detection stage 118

5.4 Conducting recommendations stage 120

5.4.1 Problem definition 121

5.4.2 Method 121

5.5 Dataset 126

5.5.1 PUCK dataset 126

5.5.2 ARAS dataset 128

5.5.3 ADL Normal dataset 129

5.6 Evaluation 130

5.7 Scope of improvements directions for RCS 136

5.8 Conclusion 137

6 A Blockchain Framework Data Integrity-enhanced Recommender System 139

6.1 Introduction 140

6.2 Related work 143

6.2.1 Smart m-health based blockchain 143

6.2.2 Recommender system with blockchain 144

6.3 Background 145

6.3.1 Blockchain . . . 146
6.3.2 Ethereum and Smart contract . . . 147
6.3.3 Blockchain oracle . . . 148
6.3.4 Chainlink . . . 149
6.4 Proposed framework for reminder care system (RCS) based blockchain 150
6.4.1 Managing the critical data using smart contract and Ethereum 155
6.5 Evaluation . . . 159
6.6 Conclusion . . . 163

7 Conclusion **164**

Bibliography **169**

List of Figures

1.1 Motivating scenario of a recommender system for IoT in the context of Eldercare. 2

2.1 (a)User-based collaborative recommender system,(b)Item-based collaborative recommender system 18

2.2 Content-based recommender system 22

2.3 The homepage of our fitness application [1]. 44

2.4 Initial screen and inquiry screen of lifecare recommendation mobile service [2]. 45

2.5 Smart Carpark Recommender PoC [3]. 46

2.6 Contextualization Architecture [4]. 47

2.7 Screenshots for the prototype [5]. 50

2.8 A sample application scenario in which different applications benefit from the SIoT by using other application's data [6]. 51

2.9 The structure of the UTravel mobile application [7]. 53

2.10 Screenshot of the webpage. [8]. 54

2.11 Number of publications, in-field cited works, and in-field citations. . . 56

2.12 Proposition of RSIoT applications. 56

2.13 Reminder care system framework. 64

2.14 Proposed framework for RSIoT. 67

3.1 Example of sensor deployment in home environment 79

3.2 Overview of the proposed methodology 79

3.3 Activity area modeling 80

3.4 The structure of DeepConvLSTM 83

3.5 Example of living area ontology 84

3.6 The confusion matrix of complex activity recognition 92

4.1 The framework of Deep Q-Network (DQN). 99

4.2 The architecture of our framework, which consists of two main parts: offline and online. The offline part focuses on training our agent G using different datasets. During the online part, the agent receives each activity and extracts the required features as state s, then nominates suitable item a_i for this state. After a periodic time called reward delay period T_r, the agent will receive a reward r. Two kinds of updating will be applied for the system: periodically update after every recommendation P and the total update after a period of time using the experience replay buffer. 102

4.3 Performance of our system among three datasets when the Reward Delay Periods T_r is 5s. 106

4.4 Performance of our system among three datasets when the Reward Delay Periods T_r is 10s. 107

4.5 Performance of our system among three datasets when the Reward Delay Periods T_r is 15s. 108

5.1 The three main concepts in the Contextual Bandit Approach:(1) State that represents the current situation for the user, (2) Action that provides the required item that meets the user need , and(3) Reward that considers as feedback for the system to improve the quality of recommendations 117

5.2 Overview of the proposed methodology. 119

5.3 (1) The agent recommends a coffee machine to Aris whereas she uses milk instead; then the system waits for 15s; (2) The feedback is received by the system as a reward and it is calculated accordingly as the coffee machine is the wrong item; and (3) two kinds of updaing for the system: minor update is after receiving the reward and the major update after certain period around 24 hours. 125

5.4 The comparison of models for each dataset. 131

5.5 The Reward Delay Periods T_r=5s. 132

5.6 The Reward Delay Periods T_r=10s. 133

5.7 The Reward Delay Periods T_r=15s. 134

6.1 Exemplary blockchain process[9]. 147

6.2 Overview of proposed framework 150

6.3 The processes of storing the normal sensory data off chain. Here, the data from temperature sensor is sent to the Raspberry Pi every half second and then it's stored in IoT server to be available for accessing at anytime 154

6.4 The processes of sending the critical sensory data on the blockchain. . 155

6.5 Emergency Notification Contract. 157

6.6 Temperature Consumer Contact. 157

6.7 Blood Pressure Consumer Contact. 157

6.8 Aggregate Data Contact 158

6.9 An example shows the data integrity for a patient's blood pressure. . 158

6.10 Advice Smart Contract. 160

6.11 Transaction Cost in Kovan and Rinkeby. 161

6.12 Smart Contracts performance in Kovan and Rinkeby. 162

List of Tables

2.1 Reasons for using a multi-agent system. 33

2.2 Trade off among RSIoT techniques. 58

2.3 Categorization by techniques type,applications domains and recommendations 68

2.4 Categorization by techniques type,applications domains and recommendations(continued) 69

2.5 Categorization by techniques type,applications domains and recommendations(continued) 70

2.6 Categorization by techniques type,applications domains and recommendations(continued) 71

2.7 Categorization by techniques type,applications domains and recommendations(continued) 72

3.1 Example of activities 80

3.2 Default reminders with rating (derived from Zhou et al.[10][1]) 88

3.3 Baseline test of elementary activity recognition 90

4.1 System Notations 101

4.2 Explaining policies and tuning hyperparameters for our agent. 105

4.3 The cumulative mean reward of our system among three datasets using different reward delay periods. 109

5.1 System Notations 120

5.2 Tuning hyperparameters for the OSL model policies 130

5.3 The cumulative mean reward of our system among three datasets. . . 132

5.4 The cumulative mean reward of our system among three datasets using different reward delay periods. 135

6.1 Transaction Cost Comparison 161

List of Publications

This dissertation contains parts of contexts of the following publications:

[1] M. Altulyan, L. Yao, S. S. Kanhere, X. Wang, and C. Huang, "A unified framework for data integrity protection in people-centric smart cities," *Multimedia Tools and Applications*, vol. 79, no. 7, pp. 4989–5002, 2020.

[2] M. S. Altulyan, C. Huang, L. Yao, X. Wang, S. Kanhere, and Y. Cao, "Reminder care system: An activity-aware cross-device recommendation system," in *International Conference on Advanced Data Mining and Applications*, pp. 207–220, Springer, 2019.

[3] M. Altulyan, L. Yao, X. Wang, C. Huang, S. S. Kanhere, and Q. Z. Sheng, "A survey on recommender systems for internet of things: Techniques, applications and future directions," *The Computer Journal*, 2021.

[4] M. S. Altulayan, C. Huang, L. Yao, X. Wang, and S. Kanhere, "Contextual bandit learning for activity-aware things-of-interest recommendation in an assisted living environment," in *Australasian Database Conference*, pp. 37–49, Springer, 2021.

[5] M. Altulyan, L. Yao, C. Huang, X. Wang, and S. S. Kanhere, "Context-induced activity monitoring for on-demand things-of-interest recommendation in an ambient intelligent environment," *Future Internet*, vol. 13, no. 12, p. 305, 2021.

[6] M. Altulyan, C. Huang, L. Yao, X. Wang, and S. Kanhere, "Deep reinforcement learning for dynamic things of interest recommendation in intelligent ambient environment," in *Australasian Joint Conference on Artificial Intelligence*, 2022.

Chapter 1

Introduction

Recent advance in Recommender system for the IoT (RSIoT) has enabled a number of applications such as recommender system for smart homes, m-health, tourism, and marketing. RSIoT is essential to improve human life since IoT collects myriad data every single moment of their life where computing systems can use them for monitoring, analyzing, and assisting procedures. It also plays main a critical role in promoting and investigating the advantages of IoT. They generally construct recommendations that facilitate users' choices based on their preferences. The most vital feature of such a RSIoT is its ability to exploit the efficient data source to predict the user's needs in a timely manner by considering any changes of the human pattern. However, three inevitable questions to ask are: How can we define the things we really want or interested efficiently? How can we conduct recommendations based on our preferences? How can we ensure the user will be interested in the conducted recommendation? The traditional recommender system paradigms can no longer deal with billions of things interconnected over the internet. We need a new and more efficient paradigm working in a proactive manner, which refers to the ability to address the three major problems in previous questions. To illustrate the important of RSIoT let's consider the case of Aris, a 77-year-old woman with dementia (see figure 1.1), who lives alone in a house and is preparing a cup of coffee in her smart kitchen. Motion sensors monitor her every move and track each coffee-making step. If she pauses for too long, a recommender application will remind her of what to do next. If she tries to prepare a cup of coffee late at night, the system considers the time and recommends she goes back to bed instead. Later that day, Aris's son accesses the secure application and scans a checklist from the phone in his mother's house. He finds that his mother has taken her medicine on schedule, slept, eaten regularly, and continued to manage her daily activities on her own.

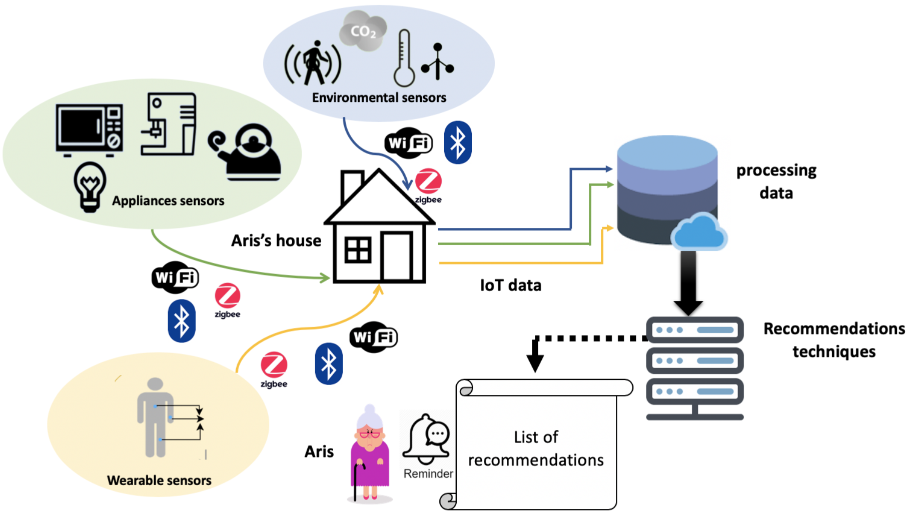


Figure 1.1: Motivating scenario of a recommender system for IoT in the context of Eldercare.

There are two main differences between traditional RS and RSIoT. These differences are briefly explained as following: (1) Most traditional RSs processes depend on two main sources: item and user and the history of their interactions while RSIoTs construct their recommendations by exploiting both history data and real-time data from sensors , and (2) most of traditional RSs such as commercial or marketing recomender systems focus on managing items whereas RSIoTs consider user preferences as a target.

1.1 Overview of Recommender System and Internet of Things

- **Internet of Things(IoT)**

  IoT allows internet-enabled physical things to connect, communicate and exchange data [11]. RFID, wireless sensor networks, and embedded objects, known as smart things, form a network that bridges the physical and virtual worlds. Smart mobiles, multimedia appliances, toys, and all kinds of other devices can be embedded with sensors to participate in the network. The IoT can be considered a combination of ubiquitous computing, pervasive computing, and mobile computing. The ultimate goal of IoT is to provide a seamlessly integrated platform through which applications involving either physical things or traditional virtual resources can be developed and integrated through the Internet. In other words, we can treat physical things as traditional web resources so as to access and interact with them. The IoT relies on the existing internet standards and architectures, and researchers are seeking ways of reusing and adapting current internet standards, such as TCP/IP, HTTP, and web services, to physical things, [12, 13, 11].

- **Recommender System (RS)**

  RS proactively recommends items that users may prefer. It has evolved through three main generations from RS for E-commerce, context-and social-aware RS, and RS that seek to handle IoT data [14]. Several approaches are used to build RS; however, the conventional approach comprises three main categories: collaborative filtering [15], content-based [16], and hybrid RS [17]. Collaborative filtering recommends items for a particular user based on the ratings of previous users. Content-based methods recommend items from the same category as the items that the user has targeted before. The hybrid approach combines two or more recommendation methods.

1.2 Challenges in RSIoT

Many conventional recommendation methods have been employed in RSIoT. However, these methods face several challenges and still are not efficient to be adapted for recommender systems that deal with sensors data in real time. Part of the challenges are connected with conventional recommender system such as product recommendations for online stores, and advertisement recommendations for social media platforms while some are unique to RSIoT and require dedicated approach to construct recommendations that meet user need in timely manner. However, RSIoT problems are still more complex than traditional recommender system as we mentioned above that RSIoT not depends only on the history data or the interaction between user and item in conducting recommendations, but also exploiting the real time data to make the recommendations more accurate and to be able to tackle any change of the human activity pattern during the recommendation procedures. To clarify this point, we use Aris scenario as an example; if Aris usually drinks her coffee in the morning, the traditional recommender system will keep recommending the coffee machine to her at the same time using the history data and the interaction between Aris and this item. So, when Aris is no longer drinking a coffee because a medical advice from her doctor, the system will keep recommending the same item to her based on the previous sources of data, but with RSIoT the system will stop recommending a coffee machine to Aris based on the medical advice that is received.

Here lists a few categorizes of challenges that faced by RSIoT, which constitute the main aspects of concern when designing an RSIoT:

- **Diverse relations**. A common challenge for an RSIoT is that it needs to consider heterogeneous relationships among users, things, data, information, and knowledge due to the poor interoperability between things and data [18, 19, 20]. Therefore, a recommendation technique should be able to conduct accurate recommendations by discovering and leveraging all the different relations. In chapter 3, we built an ontology model to discover the relations among three data sources: home appliance, environmental sensors and elementary activity recognition.
- **Scalability**. An RS should be able to perform consistently when the whole IoT-based system scales out and the related data accumulate [21]. An IoT-based system is subject to scaling out in both hardware and software, as more and more individual things are connected into the IoT and continuously contribute to the explosive increase in the number of things. Along with this process, the amount of data may also increase dramatically. For this reason, the recommendation framework should maintain stable performance in terms of providing a reasonable response time and consistent accuracy [22, 23, 24, 25].
- **Dynamicity**. An RSIoT needs to handle three aspects of dynamicity: dynamic discovery of things at the network level [26], dynamic discovery of user preferences based on their situation [27], and on-demand real-time recommendation [28, 5]. Besides, the location of things, environmental parameters, and related resources can dynamically change [29]. In chapter 4 and chapter 5, we

adapted two approaches to tackle the dynamicity issue of our system: deep reinforcement learning and contextual bandit.

- ***Big data management***. IoT devices generate massive amounts of data, so an efficient big data management system is needed to deal with various kinds of data and variable velocity [30]. Providing reliability and scalability for this system is important to ensure that it works with no downtime.

- ***Trust management.*** This challenge arises when the RS deals with a large distributed sensors network. The system should have the ability to defend against malicious nodes by establishing a technique to detect untrustworthy entities. One technique to ensure this trust and reputation among IoT devices is allowing each IoT device to evaluate the trustworthiness of the others [31].

- ***Privacy***. RSIoTs deal with a huge amount of sensitive data concerning individuals, people, organizations, businesses, and health providers. Authentication, authorization, data encryption and integration, and fine-grained access control are crucial to protect these data [32]. During collecting the data or even dealing with public datasets, the data were collected using sensors that do not violate the user privacy. There is no cameras are installed in the testbed that could make the users feel uncomfortable in their houses.

- ***Security***. Key security risks can be grouped into three categories: risks connected with physical components, such as counterfeit attacks, false attacks, information tampering, and network damage; communication risks, such as DoS and DDoS attacks; and risks associated with applications, such as information disclosure, authentication, illegal human intervention, and unstable platforms [33]. However, dealing with security issues for RSIoT is more complex than traditional RS because of the heterogeneity of the objects and the large scale of the network. Data integrity issue was highlighted with preliminary solution by adapting blockchain technology in chapter 6.

- ***Interoperability***. Enabling communications among IoT devices that have different standards and protocols can be a critical issue. Networking protocols should be adapted to eliminate the restrictions between constrained, and unconstrained entities in the IoT environment [34].

- ***Quality of service***. Quality of service is one of the main requirements of any system in IoT that ensures availability, efficiency, scalability, and adaptability. It is crucial for IoT systems to plan effectively, use resources efficiently, and respond to queries immediately and adaptively in a highly dynamic environment [35].

- ***Heterogeneity.*** There might be diverse resources in IoT, so significant distinctions among these resources need to be hidden to provide a consistent presence[18].

1.3 Conclusion

In line with previous questions, the overall goal of this thesis is developing a unified recommendation framework for data-driven, people-centric smart home applications. It consists of three main stages:

- Multi-Source Contextual Correlations. In this stage, we present a Reminder Care System (RCS) to discover human activity based on multi sources which includes interactions between user and things, spatial-temporal contexts and contemporary activities. This stage is covered by chapter 3.
- Context-aware Behaviour Recommendation. Two approaches are adapted for this stage: Contextual bandit and deep reinforcement learning to deal with dynamic environments and conduct updates without waiting for user feedback. This stage is covered by chapter 4 and 5.
- Trust-aware Internet of Things Service Recommendation. We focus on data integrity threats that may affect the accuracy and consistency of the data. This stage is covered by chapter 6.

1.4 Dissertation Organization

In view of the challenges in RSIoT, in this dissertation, we briefly introduce overview for IoT and RS and, challenges in RSIoT. The following chapters are organised as follows:

Chapter 2 Recommender system for the internet of Things Techniques and Applications: Literature Review The state-of-the-art RSIoT techniques and the development of applications for RSIoT in a variety of domains are surveyed in this chapter. Analysing with findings for the state-of-the-art methods are provided. We propose a reference framework to compare the existing studies and guide future research and practices.

Chapter 3 Reminder Care System (RCS): An Activity-aware Cross-device Recommendation System This chapter presents a work that aims to detect complex activities using three kinds of features: elementary activity, environmental sensor, and home appliance as input for our system. We first utilize DeepConvLSTM for elementary activities recognition with wearable sensors. Then, we build ontological models to considering spatial, artifactual and environmental contextual information, to boost our complex activity recognition by producing rules. Our experiments demonstrated that our behaviour recognition part is effective to be deployed as input for the Reminder Care System (RCS).

Chapter 4 Deep Reinforcement Learning for Dynamic Things of Interest Recommendation in Intelligent Ambient Environment The variety of human activity pattern one of the most challenging and common issues that could affect the quality of recommendations. Based on this, we harness the advantages of deep reinforcement learning in addressing not only the issue of capturing dynamicity patterns of human activities but also to update our system without a focus on user feedback. In particular, our RCS is formulated based on a Deep Q-Network (DQN) which works well with the dynamic nature of human activities. We further consider harvesting the feedback automatically in the back end without requiring users to explicitly label activities. Experiments are conducted on three public datasets and have demonstrated the performance of our proposed system. .

Chapter 5 Context-induced Activity Monitoring for On-demand Things-of-Interest Recommendation in an Ambient Intelligent Environment We consider a context as a feature to tackle dynamicity in human activities and to construct accurate recommendations that meet users' needs in various scenarios. Based on these, a contextual bandit approach is utilized in the formulation the proposed recommendation system. This approach achieves a competitive performance compared to the deep reinforcement learning. The main reason is that the contextual bandit (CB) approach can exploit the context of each state and recommends a suitable item for each state independently without effecting the next one. The experimental results on three real-world datasets demonstrate the feasibility and the effectiveness of the proposed system for real-world IoT-based smart home applications

Chapter 6 A Blockchain Framework Data Integrity-enhanced Recommender System Data integrity threats may affect the accuracy and consistency of the data particularly in the IoT environment where most devices are inherently dynamic and have limited resources that could fail in ensuring the quality of data transmission. Prior work has focused on processing big data and ensuring their integrity by considering Cloud Storage Service (CSS) as the popular way. Here, we address integrity of data leveraging blockchain capabilities to ensure the integrity of the critical data. We adapted the Ethereum blockchain to our RCS for ensuring integrity of data during sharing them between doctor and patient without handling their data by third party. We build four smart contracts that enable our system of gaining more advantage of blockchain. We evaluated the performance of our smart contracts in Kovan and Rinkeby test networks. The preliminary results show the feasibility and effectiveness of the proposed solution.

Chapter 2

Recommender system for the internet of Things Techniques and Applications: Literature Review

This section contains work published in:

[36] **M. Altulyan**, L. Yao, X. Wang, C. Huang, S. S. Kanhere, and Q. Z. Sheng, “Asurvey on recommender systems for internet of things: Techniques, applicationsand future directions,” *The Computer Journal, 2021.*

2.1 Introduction

With the development of new approaches and improvements to conventional recommendation approaches and techniques, numerous recommender systems for IoT (RSIoT) have been developed and implemented in various domains, such as smart homes, smart health, smart car parks, and smart tourism. However, certain new challenges can still be identified for RSIoT, and these, to some extent, are more complex than the conventional recommender approaches for three main reasons [37]:

- Dealing with and analyzing a massive amount of hugely heterogeneous data requires both comprehensive analysis and identification to conduct accurate recommendations.
- Exploiting rich contextual information is needed to provide recommendations that match the user preferences, while challenges like resource constraints could obstruct this process.
- IoT data are required to pass through several layers for extensive processing during their entire life cycle to be inferred and produce recommendations to the end-user. Accordingly, providing security for RSIoT usually requires additional security layers; it also affects a system's overall performance.

The last few years have witnessed some tremendous studies on recommendation systems. Most of them are focused on either conventional recommendation techniques, (e.g. collaborative filtering (CF) [38, 39, 40, 41]) or their applications (e.g. social recommender systems [42]). Although there are some reviews and outlooks for general recommendation systems [14], no previous research known to us has presented a comprehensive analysis of RSIoT. For example, Burke et al. [17] reviewed a landscape of hybrid recommender systems and compared them with traditional recommendation approaches. The work presented in [43] examined 177 papers on recommendation systems, classifying them into two types and describing their techniques and applications. The authors in [44] focused on context-aware recommender systems and algorithms. Also, evaluating a recommender system is usually considered to be a major challenge. In [45], the authors summarized the metrics to evaluate a recommender system and classify them based on their user dependency characteristics. Singh et al. [14] presented the three generations of recommender systems and discuss their similarity measures and evaluation metrics. Despite the significance of the RSIoT for both researchers and real-world developers, no previous research has comprehensively reviewed recommendations systems in the IoT environment to the best of our knowledge.

In this chapter, we conduct a comprehensive survey of RSIoT techniques and their application. We provide a panorama of current progress and an in-depth analysis of the reviewed techniques. Also, We provide analyses with findings for the state-of-the-art methods. Finally, a unified recommender system framework for the internet of things is proposed. It aims to compare the existing studies and guide future research and practices.

2.2 Recommender Techniques and Systems for the IoT: State of the Art

We focus on the techniques to build RSIoT. To analyze the growth of RS in the IoT environment, this section reviews the major techniques that are exploited to construct RSIoT, including conventional recommender techniques, context awareness, the social IoT technique, multi-agent algorithms, recommendations with a graph database model, recommendations with machine learning and deep learning techniques, and recommendations with reinforcement learning technique.

2.2.1 Collaborative filtering approach

This approach makes recommendations on items for a particular user that are based on the ratings of previous users [15]. It works in three stages: computing the similarities between users, selecting a group of users with the same preferences as the user who needs recommendations, and making a recommendation based on the group ratings. There are two main collaborative filtering (CF) techniques: memory-based CF [46], and model-based CF [47]. In memory-based CF, user recommendations or predictions of ratings on future items are based on the users' rating behavior by using correlations between items or between users. However, the whole training set is used each time to predict the recommendations that particularly affect the performance speed with a large dataset. This issue could be addressed by pre-calculating the necessary information and then updating them incrementally. Model-based CF is more scalable in cases where only the training set is used to build the model. It then uses this model to recommend future ratings. Compared with memory-based CF, model-based CF is considered less accurate because of the large fraction among the item-user values in the training part of the dense dataset [15].

In the last decade, CF algorithms have become a common approach to building an RS. This involves recommending items based on the history of the user or a group of users with the same interest and maximizing performance, especially when there is enough historical information. Although CF algorithms face an important challenge with new items (cold start problem), they have been implemented successfully in various domains. Some studies have investigated this approach in the IoT context. The following account provides an overview of the use of the CF approach in RSIoT, in which each RS based on this method is explained in detail. In [26], the authors proposed **a unified CF model based on a probabilistic matrix factorization recommender system** that exploits three kinds of relations in order to extract the latent factors among these relations: user-user, thing-user, and thing-thing. A directed weighted graph, probabilistic matrix factorization, and random walk with restart are all applied to extract these correlations. Such an approach was evaluated by measuring the accuracy. This was based on exploiting correlations among the social network and things with other approaches, based on probabilistic factor analysis. The results showed this approach gives better results than the other methods.

The author in [48] exploited the CF method to design an **IoT trust and reputation model** that investigated trust and reputation among IoT nodes. The main phases of this model are: (1) Alpha nodes are determined as being stronger nodes in the IoT network and are responsible for defining jobs and distributing processing among the nodes in the network; (2) Each node in the network provides ratings for its experiences, and consequently, a rating matrix is built. (3) Probabilistic neural networks (PNN) are used to divide nodes in the network that have the same functionality into clusters by calculating the similarity among them; (4) Recommendation weights are calculated, and Quality of Recommendation (QR) is defined as a score of trustworthiness; (5) The sensitivity found in each transaction is defined by using a flag parameter; (6) The trust value between the two nodes is computed; and (7) The nodes are classified into trustworthy and untrustworthy. Here, one of the PNN layers is used to calculate the probability density function (PDF) that defines the probability of the node belonging to either the untrustworthy or trustworthy class. The two main features of this model are: it deals with the cold start problem by predicting the value for each new node, and it classifies the data depending on its sensitivity in enhancing the system's performance.

In [49], the authors proposed an airport fog-to-cloud system, based on the F2C project, which conducts recommendations for both novice and experienced travellers. The main idea of the system is to use the data of the traveller's device to recommend a suitable event, such as a flight's last call, flight closing, check-in opening, etc. It consists of several entities: Users, who provide all the details about their destination; Sessions, which start when the user contacts any fog node; Position, to define an attribute for both POIs and users. The other entities are Topic, which is chosen by the user to initiate the user preferences; the POI, which represents the user preferences; Promotions, which are based on the POI; and Score, which considers the entities to conduct effective recommendations. To recognize a user's preferences, the CF adapts. This is based on either the historical actions of the user or similarities to other users.

For recommender system, knowing the user preferences is the key to conduct accurate recommendations successfully; consequently, sufficient data sources are needed to feed the system. In [50], the CF approach was adapted to address this issue. Here the authors exploited the weather and location data that was collected by the sensors to provide effective recommendations for the residents of that geographical region. This is called **the weather and location-aware recommendation system** and the architecture of this RS comprises the following functional blocks: W-Historical Weather Data Processing; S-Historical Sales Data Processing; and R-Making Recommendations. In historical weather data processing, weather data are collected and processed; if a record has missing data, it can be purged either by filling the missing interval by the closest reading or taking the average of previous readings. The Hidden Markov Model is used in this block to address the problem of defining weather conditions during short intervals. In historical sales data processing, two types of historical sales data are collected: temporal data and specific location data. If there are missing data, the system can manage the situation by using a trend model to extract the highest sales during a given time interval at a specific location. Recommender systems for location recommendations may fall short when dealing with the new region. The lack of information (the cold start problem) is the main issue in this situation. In [51], the authors designed an algorithm that has the ability to tackle this issue by considering the users' location history and user reviews information.

Some studies exploited two approaches of CF to build their systems. For illustration. In [52] the authors proposed a framework that combines cyber-physical systems and IoT to make recommendations. The system architecture consists of four layers: a selection layer, network layer, services layer, and application layer. The authors discussed the two main approaches to CF to build an RS: a user-based approach and an item-based approach. In the user-based approach, the user plays the leading role; people with the same taste are grouped, and the recommendations for the user are based on evolution items that are rated by the group with the same preferences as the user (see figure 2.1(a)). The item-based approach provides recommendations by building neighbourhoods for the items that are preferred by the user (see figure 2.1(b)). The work in [53] extended the user-based approach to provide group recommendations in an IoT environment by considering the member organization. The member organization can affect the decisions of members of other groups. The system consists of three main parts: (1) constructing the rating matrix from the history accessing services by normalizing the frequency of services access; (2)

defining the member organization by using three kinds of similarity metrics: group size-based, common member-based, and member preference; and (3) user-based CF based on group similarity, which is responsible for dealing with the cold start problem, predicting ratings to the new group and producing recommendations. The experiment's results showed that the member organization-based group similarity (MOGS) metrics outperformed the baseline approach. Also, in [54, 55], user-based

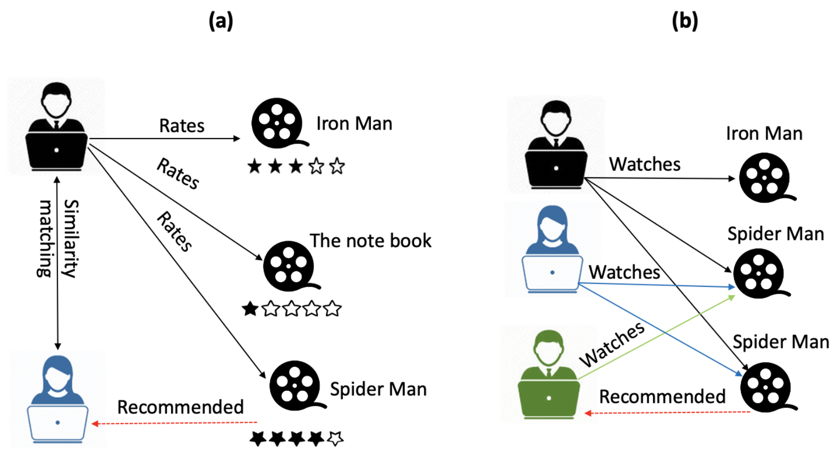


Figure 2.1: (a)User-based collaborative recommender system,(b)Item-based collaborative recommender system

and object-based are adapted to the RSIoT part in the **MUL-SWoT model**. The RSIoT is responsible for ranking the services that are received by the third party and filtering them to provide recommendations. The authors use some metrics to evaluate these algorithms; the results show that Object-Based Collaborative Filtering (OBCF) was better than User-Based Collaborative Filtering (UBCF).

Some studies have sought to make this approach to build an RSIoT more efficient by addressing the major problems of scalability, and cold start [56]. In [57], the authors proposed a system that could recommend the best among several nodes that had the same services. When the recommendations part of the system was built, it was based on the CF approach and optimized by addressing the two main challenges: the cold start problem and scalability. It tackled the first problem by using a graph-based trust and the second by allowing each node to predict the rating rather than a central system. The trust and similarity between the nodes were measured based on the rating and structure of the network. The experiment results showed that the combination of trust as a new influence and similarity as a traditional influence in CF improved the accuracy and coverage. However, this approach may not be able to address the same issue in RSIoT where the system deals with huge amount of data that is sensed by groups of sensors, and graph based depends on graphical analysing to predict the link between item and user (activity) as result complex computation and time consuming will be added to the system.

For medical-related problems, collaborative filtering cannot be adapted directly. Consequently, the authors in [58] designed a modified version called Advice-based CF to build an RS that provides suggestions for cardiovascular-disease patients. It combines three major parts that each had unique functions: data collection, classification, and recommendations. The first part is responsible for collecting data from the patient by using biosensors. The collected data is there after cleaned and filtered for extraction. The second part classifies cardiovascular diseases into eight classes. The authors proposed four classifiers of machine learning: SVM, Naive Bayes, random forest, multi-layer perceptron. The last and main part is used to produce recommendations for patients in a remote area.

The common problem with each RS, whether it is conventional or IoT-based, is the lack of information. Some studies attempt to address this problem by adding social network data as an input source [59]. For example, authors in [60] combined two main algorithms to improve the accuracy of avenue recommendation: a CF-based algorithm using social network data and the QoS-based algorithm. The main operations of the system have the following steps: first, the QoS algorithm calculates the cost for each avenue and then the atmosphere. The CF computes the impact level of the user's social friends, but, first, the algorithms are worked offline. After the completion of this offline initialization, the algorithms are executed online to conduct the recommendations in parallel. The recommendations of both algorithms are then combined using the WCombSUMi formula [61]. Finally, the repository data are updated to maintain the consistency of the system. The study [62] also proposed a location-aware POI RS that exploits three kinds of data sources to conduct accurate recommendations: user rating reviews, POI data, and user data. The system has the ability to provide recommendations for the user when he moves to a new region, even if there is no activity history for him. The Matrix factorization approach is applied to the recommendation engine, and this gives the system three main features: accuracy, scalability, and flexibility. The system is tested using a public dataset (yelp dataset), which gives a high degree of accuracy.

In [63], the matrix factorization approach is exploited to present a framework for museum tour recommendations. Authors in [64] adapted the collaborative filtering approach for their recommender system but with giving attention to the connection between the user preferences and time and dividing the similar users into groups. Although the collaborative filtering approach has been adapted in numerous studies, as we discussed in previous parts, there are potential problems that make it inefficient for RSIoT, as we discuss below. These are:

- **Scalability** RSIoT deals with a large amount of data that needs computation power to conduct the recommendations, as well as fast response to online user requirements.

- **Cold start problem** particularly when a new activity pattern joins the system as mentioned in Aris scenario. There is insufficient information to identify an appropriate group of previous activities for rating purposes or when a new sensor is added to the system. For RSIoT, addressing the cold start problem is more complex where it cannot be dependent only on the history data to fill the missing as in conventional RS. Especially, when the system deals with critical cases like Aris; so, the system needs to find sufficient data source to conduct the accurate recommendations in timely manner.

- **Data sparsity** may affect the accuracy of collaborative RS, especially RSIoT, which receives huge sensory data from multiple channels, as we elaborated in Aris's scenario that the system needs to consider any change in her pattern even the activity is still the same, it could be done in a different way. So, the correct item has to be recommended in a specific moment depending on both history data and real time data. Consequently, CF approach may fail to address this issue in RSIoT.

2.2.2 Content-based approach

The content-based (CB) approach recommends items that are similar to the items previously targeted by the user. Instead of relying on ratings, it uses the existing interest history to predict user interest in the target and match the content of similar profiles with the target content (see figure 2.2)[16]. Therefore, it may suffer from the cold start problem [52]. The CB approach has been used for recommendations in the IoT. In [65], the authors proposed two RS for their AGILE project, which aims to improve users' health conditions. As a result, two new apps handle patients'

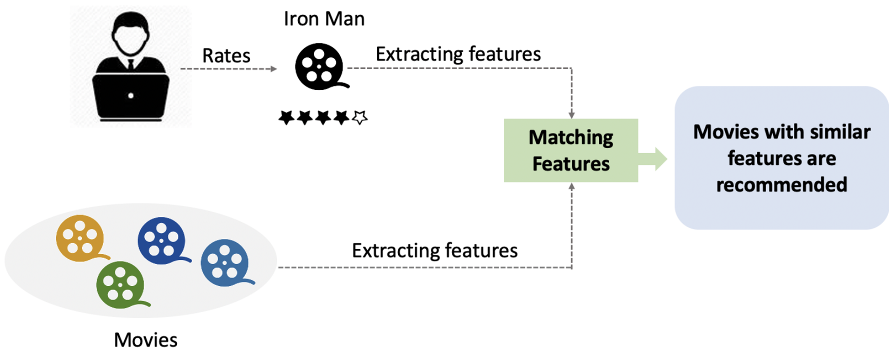


Figure 2.2: Content-based recommender system .

healthcare devices and physical activity plans. The CB approach is used to build the recommender engine for a second app as a basic recommendation approach. The idea of the app is to collect the patient's data through medical sensors that can be worn, and a virtual nurse is illustrated as a case to explain this. Based on these data, the app can recommend a suitable activity plan for the patient. In [66], authors shift from basic recommendation approaches to develop three recommendations approaches: Sequence-based recommendation (SEQREQ), Recommendations for configurators (CONFREQ), and recommending diagnoses (DIAGREQ) to meet the requirements of their scenario.

In another illustration, in [67], the authors adapted a CB approach to building a recommendation module for their smart restaurant, which aims to provide dish recommendations based on the customers' tracking history. Jaccard measure was adapted to measure the customers' similarity and the dishes to conduct recommendations based on user preferences. The CB approach has also been applied to parking applications. However, based on this technique, the applications will only make recommendations that are based on the similarities among the customers' parking profiles without considering their user preferences. Therefore, accurate recommendations resulted in parking issues. To address this problem, the work in [68] proposed a periodical recommendation that is conducted during three periods: recommendations at the entry gate, recommendations for the nearest parking zone, and recommendations for the new zone, when the recommended parking is taken. The system uses three main dimensions as data sources: user profile, user preferences, and nearest zone. The distance matrix is adapted to calculate the distance between the user and a zone. To minimize the computation time, the system can remove unrelated zones from the zone list. The system was also tested using a simulation dataset, and the result shows it outperforms other systems that focus on user profiles or user preferences only. Although CB has a number of features compared with CF, such as building a profile for each user which depends on his history rating, extracting features of each item to decide which item should be recommended, and dealing with new items; nonetheless, it has some shortcomings: (1) focusing on items and their features which affect the quality of recommendations when user's pattern is changed, (2) lack of ability to deal with a new user where there are no items belong to, and (3) keeping recommending items that are similar to those already rated without finding any unexpected [69]

2.2.3 Hybrid approach

The hybrid based (HB) approach combines two or more approaches to build an RS. For example, by combining the collaborative and content-based methods, the limitations of each can be addressed. The different ways of merging collaborative and CB approaches into a hybrid RS are as follows [17]:

- Building a hybrid RS by implementing content-based and collaborative features separately and mixing their predictions.

- Combining some content-based features into a collaborative approach.
- Combining some collaborative features into a CB approach.
- Building a general unifying model that combines collaborative and content-based characteristics.

Significant research efforts have been devoted to exploiting the hybrid approach to improving recommendations and search performance in IoT. In particular, the framework of the recommendations for Digital Signage (DS) in Urban Space [70] exploited the hybrid approaches to make DS more attractive. The framework has five components: IoT sensing, data pre-processing, recommender engine, DS user interface, and DS data storage. The recommender engine plays the main role in investigating product taxonomy, advertisement taxonomy, condition feature groups, and the DS recommender model of the framework. It can deal with multiple reviewers by switching to various reviewer modes. The system's results were based on a digital signage system deployed in a Taipei shipping mall and showed that both the demographic and context features are crucial factors in providing accurate recommendations. To reduce the latency of recommendations, in [71], a user-space customized RS platform system is proposed that will exploit the mobile edge environment. The conceptual architecture combines several components to build an RS that produces high-quality recommendations. The system produces the recommendations by exploiting user feedback data and sensor data that are closed to the user. This system deals with two user scenarios: (1) if this is the first time the user has dealt with the system, the recommendations are based on similarity with other user's preferences (collaborative filtering); (2) if this is not the first time the user has engaged with the system, the user's feedback data in the repository is considered in the recommendations (content-based). In [72, 73], the authors built their RS engine using a hybrid recommendation algorithm that combines CF and CB approaches. The RS framework is based on the cognitive process. It consists of three main layers. In the first layer (Requirement layer), the goal of the system is described by using a Cognitive Specification Language (CSL), which it then sends to the next layer. The Things layer observes the environment and provides the recommendations list. The main layer is the cognitive process layer, which contains the cognitive recommender engine that is responsible for receiving the sensory data from the Things layer and executing the recommendation algorithm. Authors in [74] proposed a recommender system for health care that combines machine learning algorithms to analyze the IoT data and a hybrid approach to model the recommendation part. In light of this, combining two approaches, such as content-based with collaborative, may tackle their issues, but it fails to tackle RSIoT issues where most of the traditional approaches focus only on the interaction between user and item.

2.2.4 Knowledge-based approach

This approach recommends items to the user based on knowledge about the users, items, and their relationships. The Knowledge-based (KB) approach has a functional knowledge base whose performance is based on the identified relationship between a user's need and a possible recommendation [17]. It does not depend on user rating or gathering information about specific users to provide recommendations. The KB approach addresses several limitations of other approaches, such as the ramp-up (cold start) problem. Ontology is a formal method of representing knowledge that is central to building RSIoT.

Creating and evaluating the ontology are complex and time-consuming processes, but several studies have used them to build RS in IoT. For example, the authors in [75] proposed a method for generating automatic rules and recommending the best rule, which requires no complicated configuration or extra efforts at the user end. Also, the user has the opportunity to add new rules for newly connected devices. To increase performance, the authors further applied two steps with three ontology models and collected open web data related to rules by a data pipeline. First, three ontology models were created: (1) Things (knowledge base), which provides all information for things; (2) Context (knowledge base), which provides contextual information about people, environment, and things; and (3) Functionality (knowledge base), which links the functionality of context and things. Second, data about the automatic tasks were gathered from the web (web crawler) to be stored in the IFTTT crawled data. Then the collected data from the web were mapped with the data of the three ontology models in order to produce the home rules. The user can add new automatic tasks when a new device is added (sensor) by recommending some tasks to the system. However, adding new automatic tasks might conflict with current tasks. For this reason, the authors defined four potential conflict situations to be solved. In [76], a conceptual framework called the web of objects (WoO) is proposed that provides smart spaces services recommendations by using semantic ontology. It combines three main layers: 1) a virtual objects (VO) layer that represents real-world objects, 2) a composite virtual objects (CVO) layer that accumulates VOs and their information, and 3) a service layer that is responsible for providing recommendations based on the results of the previous layers. Also, the authors in [77] adapted ontology formats for smart space technology to implement an RS for historical tourism. Also, for smart health applications, ontology plays a significant role in building a recommender system to supervise chronic patients and to provide long-term care [78, 79]. However, a classic ontology cannot define the membership value of risk and uncertain factors precisely; thus, the system provides inefficient results. To address this problem, [80] adapted a fuzzy ontology to build a system that monitors diabetes patients and recommends specific food and drugs. It combines four kinds of fuzzy ontology: patient ontology which contains the history of a diabetic patient, sensor ontology, which includes the data from sensors, knowledge and rules-based ontology, which consists of the rules that help to define the patients' conditions, and drugs and food recommendation ontology to provide recommendation services. The experiment's results show that the performance of the proposed system outpaces that of classical ontology. Although KB-based approaches have been barely attended in building RSIoT, they are able to respond to any change in the human patterns by updating relationships between user and item. Yet such models still face two major issues, which cannot deal with decisions that have no rules related to the system, and requires knowledge engineering in the building, and can be expensive [81].

2.2.5 Context-aware Recommendation

Context-aware computing is well known in computer science research. In [82], context is defined as "any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects".

The application of context-recommender systems in the IoT environment faces several challenges, such as context discovery, security, privacy issues, and sharing context, but some researchers have used context-awareness in their frameworks to build

RSIoTs. **Proactive multi-type context-aware recommender system** [5] exploited the user context to define when the recommendations must be pushed to the user and which kinds of recommendations are to be pushed. The proposed approach is demonstrated in three domains: gas stations, restaurants, and attractions. The authors built the system in two main phases: identifying the situation of the user and the type of recommendations phase and the item recommender phase. In the first phase, a context-aware management system (CAMS) performs the following tasks: (1) data acquisition, where data about IoT devices, such as sensors, GPS, etc. are collected; (2) data modeling to convert the raw data into understandable form; (3) context reasoning to deduce new knowledge based on the results of the previous stages – the neural network is applied at this stage to generate several scores for the three types of recommendations; And (4) scoring algorithm to define which types of recommendations will be moved to the next phase. In the second phase, a CF approach is used to recommend the items for the user. The proposed framework was adopted as an application (see subsection 2.3.4) and then tested and evaluated with good results. The authors in [3] proposed a **Context-as-a-Service (CoaaS) recommender system** that uses an IoT context service to enable applications to provide and consume contextual information seamlessly without the need for manual integration among IoT devices. The two core components are context service description language (CSDL) and context service matchmaking (CSM). CSDL enables developers to describe their services regarding semantic signature and contextual behavior in standard language. CSM is responsible for matching between context requests and context services. The CoaaS framework was tested by adapting it to produce smart car park recommendations via an application on a smartphone; this is explained in detail in section 2.3.2. The **EW4 model** [83] exploited contextual information and the mobility of the user's tweets to provide accurate recommendations. The framework combines both offline and online components. The author has proposed an algorithm using a generative process to model the offline components, which is based on four aspects: who (User), where (Geo), when (Time), and what (Words). Several applications based on this model could be built, such as location recommendations, user prediction, and location prediction. The EW4 model's performance for some tasks was evaluated using two datasets.

The authors in [4] proposed an RS that could provide accurate recommendations by considering contextualization with IoT data. They defined two operations for contextualization: contextual filter and contextual aggregation. In the contextual filter step, data on IoT devices and services are filtered based on the current context. Contextual aggregation involves combining the filtered data based on both contextual similarities and relevance. The authors adapted the framework to design a smart parking application (see subsection 2.3.2). Zhou et al. [84] proposed a model which could improve recommendations accuracy by exploiting context-awareness. The model consists of three main components: firstly, the server, which is responsible for providing service recommendations to users; secondly, the user, who provides the context to the server in order to get recommendations; and thirdly, the services provided by the server. The authors designed a Hierarchical Social Contextual Tree (HSCT) algorithm that uses three sources to provide accurate recommendations: service usage frequency, contextual bandit feedback, and social intimacy. The experiment's result showed that the system outperforms other algorithms, especially in two dimensions, namely scalability and the cold start problem. Also, in [85] the context is adapted as a source to build a recommender engine for recipe recommendations in a smart kitchen. The recommender engine consists of four main components: food item taxonomy, which represents the food items and their attributes in the refrigerator and cupboard; recipe taxonomy, which includes all the recipes and their ingredients; conditional groups, which combines environment and context features; and the recommender model, which utilizes the historical data to conduct recommendations. Based on the previous components, the authors described a motivation scenario to explain the importance of the proposed system. Also, context awareness was employed in building an RS for smart healthcare [86]. In the same vein, the authors [87] proposed an RS that provides social recommendations for cultural heritage (see subsection 2.3.3). The architecture consists of three main parts: socializing, social-based recommendations, and context-based recommendations. The system can deal with the sparsity issue, where the recommendations are conducted, even when the new user uses the system. Based on previous studies, context plays an important role in improving the quality of recommendations and helps the system to decide which kinds of recommendations should be conducted to the user in a specific situation. However, it still has some limitations, such as scalability, privacy, security, and sparse data.

2.2.6 Social IoT-based Recommendation Technique

The concept of social networking in IoT applications is to create a social relationship among things, which is also known as social IoT (SIoT). Such social connections can be the relationships among things and the relationships between users and things, where the relationships essentially represent social circles [6]. The authors in [88] introduced the concept of socialization between things, which focuses on finding solutions to allow smart wireless devices to establish temporary relationships. The main idea behind SIoT is that social objects can select, discover and compose services. Five main kinds of relationships are produced, based on the object's activities and features, which are explained in detail in [89]. These relationships depend on the object's activities and the features that are to be created and updated. Moreover, they require some modules to utilize these relationships, such as a service discovery module and relationship management module [6]. SIoT enables IoT applications to exploit IoT services efficiently. Hence the RS can exploit the data that are gathered from various applications. SIoT has led to some challenges that may make it not sufficient for RSIoT, such as lacking the interoperability among things and concerns on privacy, security, and trust issues, particularly with some IoT sensitive data. Yet some pioneer works have emerged recently, which are explained below.

The authors in [6] proposed a framework that could exploit the SIoT network to produce services recommendations. The SIoT network includes the profiles of all the IoT applications. The data profiles can be used to provide service recommendations or could be exploited by other IoT applications to search for similar conditions that have been solved in the past. The proposed framework has three main layers: the SIoT perception layer, which collects the data of all IoT devices and their relationships, which are extracted by using SIoT technique; the network layer, which connects the previous layer with the upper layer by using several communications protocols; and the interoperability layer, which enables all IoT applications to use the IoT data. The authors provide an example of how their framework could be adapted for various applications (see subsection 2.3.4). This would be done by exploiting the availability of social correlations among things and between users and things to provide services recommendations.

However, facilitating access to quality services and trustworthy devices in a SIoT environment has become a critical issue. To address this problem, a scheme of access service recommendation [90] is proposed that provides recommendations about trustworthy nodes. Trustworthiness is evaluated based on three parameters:

a feedback-based reputation system, social relationship, and an energy awareness mechanism. Recommendations are subject to an unreliable wireless environment and limited battery capacity in smartphone applications. In [91], the mobile IoT is exploited to build an RS for services and social partners. The RS has two main modules: a recommendation module and a physical layer module. The recommendation module combines three components: recommendation-database, service/interest-matching, and service/connection time-prediction. The recommendation database performs three main functions: (1) it collects all the information on users and service providers, including user preferences, features, and services; (2) it creates a social network page based on this information; and (3) records the historical information. The service/interest-matching (S/I-M) matches the services of the providers with user interest using different recommendation and data mining techniques. The service/connection time-prediction (S/CT-P) is used to detect a suitable time to create a D2D link, taking the quality of the wireless channel, battery capacity, and the mobility of the wireless device into account. The results of (S/I-M) and (S/CT-P) are evaluated to provide recommendations by updating the social network pages. The physical layer module is responsible for managing all the resources, such as the physical link with the partner, power consumption, and codeword rate, as well as providing communication recommendations to improve the performance.

2.2.7 Multi-Agent Algorithms

A multi-agent system (MAS) has two main goals: to provide rules to construct a complex system and provide techniques to coordinate each agent's behavior. In [92], several reasons for using a multi-agent system are presented; some of these are summarized in table 2.1. These considerations have led some studies to focus on the

Reason	Explanation
Speed	Using a multi-agent system provides several mechanisms for parallel computation, which increases the system's efficiency.
Parallelism	Each agent has various tasks or abilities.
Scalability	Some systems need to be more flexible in term of adding a new agent or new task to the system.
Robustness	Robustness is achieved by eliminating the single failure point.
Simpler programming	It is easy for programmers to tackle and control subtasks rather than tackle the whole systems.

Table 2.1: Reasons for using a multi-agent system.

use of MAS to build recommendations systems in IoT.

In [93], a multi-agent algorithm is exploited to improve the speed of the recommendations, with each thing described by thing descriptors (i.e., bit vector). Things with similar features will then be brought together and managed only by neighbor's cyber agents. Note that all cyber agents are connected by hops. The probability function is used to decide if the thing descriptor will be moved towards a linked cyber agent. The cyber agent evaluates the thing descriptor if it arrives from another linked cyber agent. The algorithm is designed to make it both intuitive and straightforward to produce a recommendable thing. When a query is established, a cyber agent matches thing descriptors and then forwards the query to a neighbor cyber agent with a maximum value of similarity. This process continues with other cyber agents until the node agent has a maximum similarity result, compared with the current agent, and the query process then finishes. After that, the query is forwarded to the asking cyber agent to produce the user's recommendations. The algorithm was tested and evaluated, and the results showed that the recommendations were produced faster.

The authors in [94] exploited the IoT data to build a **Mobility Recommender System (MRS)** for parking, especially in urban areas, by incorporating features such as an ideal door-to-door route. The system has two main parts: the data sources, which are divided into static and dynamic data as input to the system; and the recommender system, which is responsible for dealing with these data. A multi-agent system is designed to support a mobility query engine (MQE) that can distribute IoT data based on location, type, and complexity. The goal of using a MAS is to recommend parking spaces based on the consideration of different input data. For example, when the door-to-door route requires the user to use public transport, one agent who is aware of city policy may provide sets of recommendations to the route calculation planner (RCP) that take into account the city's regulations. Another agent who knows about the user's parking preferences may also provide several recommendations to the RCP. Finally, the MRS is responsible for collecting and aggregating the queries into a ranked list of recommendations that will be considered in subsequent recommendations, where a utility RS is used to calculate user preferences. More details about their parking application can be found in subsection 2.3.2.

Twardowski et al. [95] built an RS that uses data from mobile devices and other IoT devices to provide personalized recommendations in real-time. The system consists of two main parts: the Big Data Server Side and Mobile Devices Side. The Big Data Server Side is responsible for the collecting and processing of mobile devices by using a multi-agent system and the Mobile Devices Side, which combines the numbers of mobile devices, such as smartphones, sensors, and beacons. The main features of this system are that it uses edge services to reduce direct communication between the mobile devices and the server, deals with the sparse data by using matrix and tensor factorization techniques, and lastly, addresses the cold start problem by using the current context information and calculating recommendations in real-time. A concrete example in a smart m-health application of their RS idea is mentioned in section 2.3

The authors in [96] proposed an RS based on the multi-agent technique to optimize electricity consumption and save costs in a smart home. The system consists of three main modules with specific agents: a device module to collect data from each appliance; a crawler module, which contains the information about electricity prices; and a recommendation module that uses data from the other two modules to suggest recommendations. There is also a control agent that manages the interaction between the modules. The recommendations part is built based on two conventional recommendation techniques, which are KB and UB. However, the system has only been evaluated in a theoretical case study using the UKDALE dataset. Authors in [97] proposed a recommender system-based multi-agent approach, which focuses on the eating and physical activities of children who suffer from obesity. The system consists of two agents: a child agent that processes data about the child and a health advisor who receives the data from the child agent to produce a suitable recommendation. Based on previous studies, we can conclude that multi-agent systems can address one of the main issues of RSIoT, that is, scalability, by applying decentralization and self-management strategies. However, the MAS approach alone may not be able to tackle other RSIoT issues, specifically in dealing with diverse relations and dynamicity.

2.2.8 Recommendations with Graph Database Model

A graphical database model is defined to represent a structural schema. Models and diagrams, such as entity-relationship diagrams, are tools that designers/architects can use to test the different data structure relationships or display validity constraints graphically before implementation [98]. Graph data models are used in applications that consider the importance of the data and the relations among data at the same level. There are several benefits of using graphs as a modeling tool [99]:

- It allows the user to show all the information about an object in a single node, while the related information is referred to by arcs.
- Queries can be referred directly to a graph structure.
- Graph databases can provide efficient graph algorithms to investigate specific operations and efficient graph storage structures.

The authors in [100] exploited a Neo4j graph database to address one of the main challenges in IoT, namely, big data management. The sensory data from a smart city, which are used to produce service recommendations, are stored in a graph database. The proposed architecture has three main components: node-red, Neo4j graph database, and the recommender. Node-Red combines two parts: (1) data source flow, which is responsible for processing all data before they are transferred to the next part; (2) heating manager flow, which uses the received data to control the heating schedule for each house, as well as enabling users to interact with the system. The data stored in the Neo4j graph database define the relations among them in order to answer the user's questions. Finally, the RS exploits not only the predefined existing data (historical data) but also the real-time data stored in

Neo4j. The CF is suggested as an approach to defining the similarity among the users by using similarity metrics (Euclidean and cosine). The main feature of the architecture is that each component is independent, which enables the extension and scalability of the system's features. (Details are mentioned in subsection 2.3.7).

Also, the author in [101] exploited graph techniques to build an RS that provides IoT services recommendations to the user based on their own IoT devices. The system combines three main steps: IoT services modeling, IoT service catalog (classification algorithm), and service matching algorithms (Recommendations). The IoT services model their functions by using Typed Attributed Graphs to describe all the objects and their relations. IoT services are cataloged by using an algorithm that entails the following three steps: (1) Defining physical interfaces and their location in spaces in order to build the space's profile; (2) determining how many smart spaces share the same profile in order to distinguish the relations between the objects; and (3), a signature is defined for the list of profiles, based on the result of step 2, with the spaces that share each profile. Service matching algorithms are used to correlate between the user request and signature of the service classes. The authors discuss two different scenarios where the system can use these algorithms. One is when the user is an expert, which means he can add a new service to the catalog, and the other is when the system can make a service recommendation based on the requests of a typical user. Wang [102] proposed a service recommendations scheme for the IoT in a smart environment that combines three main components: tripartite graph, matter diffusion, and the habit feature. Despite preliminary benefits for graph database model, it is connected with two issues: generic frontends usually not adequate to support users in their tasks and the lack of dealing with time-series data while RSIoT depends on it.

2.2.9 Recommendations with Machine learning

Machine Learning (ML) can be generally considered a sub-field of Artificial Intelligence (AI). It entails the use of computers to simulate human learning and can collect and use real-world knowledge to enhance performance [103]. ML algorithms can be divided into three (supervised, unsupervised, and semi-supervised), based on the nature of the data involved, or into two (transductive and inductive learning), based on the concept learning perspective [104].

ML algorithms can be utilized to optimize the ability of traditional RS to provide accurate recommendations to the user. In [105], the authors designed the Optimal State-based Recommender (OSR) System by exploiting some machine learning algorithms, including Distributed Kalman Filters, Distributed mini-batch SGD (Stochastic Gradient Descent), Distributed Alternating Least Square (ALS) based classifier, and some ML platforms. It shifted conventional recommendations, based on user/item preferences only, into accurate recommendations that deal with real-time data. Some conventional recommendation techniques cannot be directly adapted for use in the design of RS. Accordingly, some studies have focused on improving some of these techniques or functions. For example, the authors in [106] proposed a framework to build an e-commerce RS that exploited the multi-sources of information to produce accurate recommendations. The framework has three main components: data sources, recommendation evidence weight, and fusion decision. The data source component uses microformats to provide a unified representation and platform for information about the mobile user. Recommendation evidence weight uses an improved radial basis function (RBF) neural network to define the weights of recommendations. With the improved function, the weight of each piece of evidence is easy to evaluate. In the final component, the Dempster-Shafer (DS) evidence theory has also been improved to fuse information and power spectrum to provide accurate recommendations. The framework was adapted for the exploitation of multi-source information that assists women with online clothes shopping. The results showed that exploiting multi-source information had a significant impact. In [107], the authors exploited an ML classifier to build a recommendation engine that provides personalized wearable technologies recommendations for proactive monitoring. This approach consists of three main models: (1) the classifier model, which is responsible for predicting at-risk diseases and suggesting some measurements for each person; (2) the optimization model, which is based on the measurement results and gives suitable wearable technology based on the previous measurements; (3) the Monitoring Framework, which is responsible for monitoring the readings from all the recommended devices and sending these readings to the classifier model to update its measurements.

Moreover, the K-Means algorithm [7] is adapted in the UTrave RS application (see subsection 2.3.5), which clusters user profiles to recommend points of interest for the user. The building of the UTrave application was based on Universal Profiling and Recommendation (UPR) that combine two main steps, namely, creating user-profiles and filtering. It is evaluated by two steps: (1) a simulated user to test the accuracy of the clustering user profiles and (2) real users to verify the whole system. In another experiment, Rasch et al. [108] adapted unsupervised learning to build an RS for smart homes. The system learns user patterns and conducts recommendations based on user contexts. The authors divided the system into two main phases: a training phase, with a sequence of sensor events as input, and a recommendations phase where the input is the current user context. The proposed RS is evaluated by using two publicly available datasets that are considered as a single person. Meanwhile, a decision tree [2] is used to build a system that provides lifecare recommendations.

The system combines two main parts: a peer-to-peer dataset and adaptive decision feedback. In the first part, the collection of the data is based on the peer-to-peer networking environment, Open API, and biosensors, after which a decision tree is used to defined and classify the data. The adaptive decision feedback part plays an important role in providing flexible results of the recommendations and facilitating the real-time lifecare services. In a more comprehensive study, Valtolina et al. [109] considered both the decision tree algorithm and social network to propose a multi-level RS. Others [110] adapted the Analytic Hierarchy Process (AHP) to build an RS for car parking recommendations. Simulation parking is used to test the system, where the result outperforms other modules using the same simulation. In another study, [111] ML algorithms were adapted to design an RS for music recommendations that was more accurate than traditional ones. The system used sensory data from wearable devices to detect the emotions of each user and then used this data as a source to conduct accurate recommendations. Three kinds of ML are used to classify the collected data into the target emotion, which are: random forest, kNN, and decision tree. In [112] authors adapted Frequent Pattern Growth(fp-growth) and association rule algorithm (SMILE) to build a recommender system for smart home which can recommend alternate actions when the deviated pattern for the user is detected.

2.2.10 Recommendations with Deep Learning

Deep Learning (DL) is generally considered to be an extension of ML. It applies two main steps: adding multiple layers, which increase the depth of the model, and transforming the data by using diverse functions representations and abstractions of multiple levels [113]. DL includes several components, such as activation function, convolutions, and pooling. It relies on different architectural paradigms (i.e., Multilayer Perceptron (MLP), Autoencoder (AE), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Restricted Boltzmann Machine (RBM)). The main advantage that enables DL to solve complex problems quickly and effectively is its ability to learn features. However, its limitations are a longer training time and the need for large datasets to describe the target problem. DL has been used in numerous applications that deal with continuous data, such as weather data. In [1], the DL technique is utilized to build an intelligent system for a fitness club (see subsection 2.3.1). The main goal of the system is to recommend suitable exercise or detect the users' fitness actions by monitoring the user. The authors exploited a 3D CNN to identify fitness actions that can extract time and space features from continuous frames. The public KTH dataset is used to test action recognition by using the 3D CNN, where the test accuracy reached 0.8865. Inaccurate recommendations have a negative impact on the user. To address this issue, a deep neural MLP has been adapted to make system recommendations more effective for a smart museum [114]. Authors in [115] proposed a recommender system platform to help doctors in determining the rehabilitation nutrition plan for cancer patients. Two algorithms were adapted to their system: the CNN algorithm and the BAS algorithm. Based on previous studies, machine learning and deep learning algorithms can play the main role for RSIoT in providing good data presentations and dealing with complex interaction patterns. However, they still have issues to be adapted for recommender system for the IoT [116, 117]. The novel evaluation metrics such as recall/precision accuracy are not enough to test a recommender system in the long term. New evaluation metrics should be considered such as interpretability, diversity, and trustworthiness where these metrics are not evaluating the performance but also focusing on the holistic experience for the user. Secondly, scalability where the system receives massive sensory data which means it must deal with them with no stop. Finally, hybrid approach, most of deep learning depends on a combination between two neural networks which needs to be studied well before it adapts for practical models.

2.2.11 Recommendations with Reinforcement Learning

With the recent tremendous approaches to RS, reinforcement learning (RL) has received increasing attention. It meets two majority requirements for recommendations: 1) treating the interaction between the user and the agent as the primary procedure for a recommendation; 2) learning the optimal policy to increase the cumulative reward without any predefined instructions. RL consists of three main components: the state (which observes the environment), action (which is taken by the agent based on each state), and reward (which represents the feedback for the agent based on its action). Some studies have started to adapt RL for their RS for the IoT.

Massimo et al. [118, 27] have proposed inverse reinforcement learning to model user behavior as a way of improving the quality of recommendations. This technique learns about user decision behavior by observing the user's actions and group observations to learn from a small number of samples. After the user behavior is learned by using a linear function, known as a reward function, the recommendations are produced by matching the user's predicted choices with a large group of observations of both the user and groups of similar users. The tourism domain applies the system in both outdoor and indoor environments (see subsection 2.3.3). The initial results showed that IRL could learn user preferences, even when the datasets were small and noisy. Authors in [119] used a reinforcement learning algorithm (Upper Confidence

Bounds algorithm (LinUCB)) to design a framework that provides context-aware recommendations in a smart city. Also, [120], the authors proposed a system, SML, to monitor the daily indoor activities of seniors with mild cognitive impairment. This aims to recommend the correct sequence of tasks for each activity so that the user can reach his goal. RL may provide a promising solution for RSIoT issues, particularly in tackling the dynamicity with the ability to learn the user pattern. However, it can be argued that RL still faces some issues in terms of dealing with large spaces, particularly with sequential interactions, modeling a human pattern in different scenarios, and formulating the reward.

2.3 Applications of RSIoT

RSIoTs are used in a variety of application domains, including smart homes, smart health, smart car parks, personal and social, and smart tourism. This section reviews the development of some of these applications.

2.3.1 Smart m-health

Over the last few years, m-health has attracted the attention of researchers who have seen it as a potential way of combining IoT with RS to provide long-term healthcare. As a consequence, there are now numerous applications. In [95], the data from mobile devices and other IoT devices surrounding them were exploited to provide personalized recommendations for dietary/fitness. The idea of this mobile application is that a higher-level context is calculated after the data is collected from smartphones and other sensors. Then, accurate recommendations are produced to match the user's needs. For example, when the user has finished an exercise class, the system recommends a suitable restaurant that could provide a healthy meal. Yong et al. [1] designed an intelligent application to guide users in gyms (shown in figure 2.3). This application includes a part responsible for providing course reminder recommendations that are based on the users' fitness. Another study [2] designed a lifecare recommendation mobile service which improves the quality of life, as shown in figure 2.4. Hence, under this category, our framework (see section 2.5) can be adapted to build an m-health application that connects hospitals with a smart home. Let us extend our previous scenario. If Aris with the mild stage Alzheimer's disease at home, the hospital can monitor the development of her condition remotely as well as providing medical recommendations such as taking a rest, doing some exercise, or avoiding some kind of food based on the received sensory data[121]. During the off-peak hours, the system keeps running and providing advice on using knowledge learned by the given historical information.

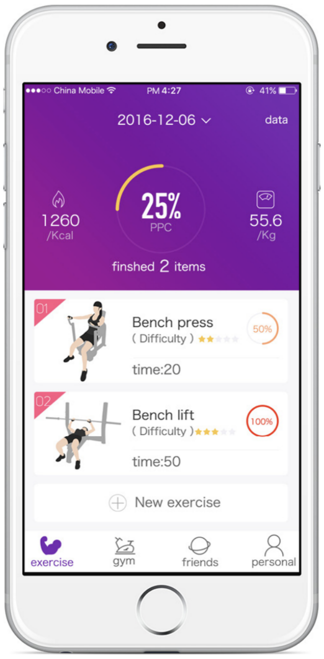


Figure 2.3: The homepage of our fitness application [1].

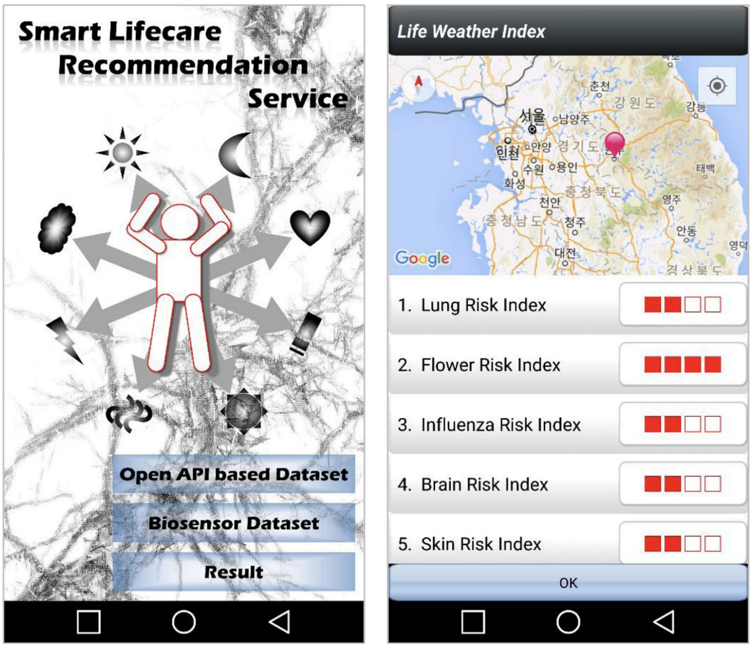


Figure 2.4: Initial screen and inquiry screen of lifecare recommendation mobile service [2].

2.3.2 Smart Car-parking

Car park recommendation is considered to be one of the crucial services in smart cities [122]. There is rapid growth in the design of applications for parking recommendations that exploit IoT data. In [3], one such application has been designed that exploits context services data owned by several providers in order to produce accurate recommendations for users. The basic idea of this application is that four context services are used as sources to produce accurate recommendations, which are described using CSDL. Also, an OBD II device is connected to the application using Bluetooth that provides some of the required sensory data, such as speed and fuel level. As can be seen in figure 2.5, the application considers the context services before providing recommendations. For example, if there is bad weather, it considers the short walking distance in the car park recommendations (figure 2.5 (a)), but if the day is sunny, the car park recommendations will consider low-cost (figure 2.5 (b)). The last part of figure 2.5 refers to the application running on a mobile phone and OBD II device, which is exploited as a data source in this application.

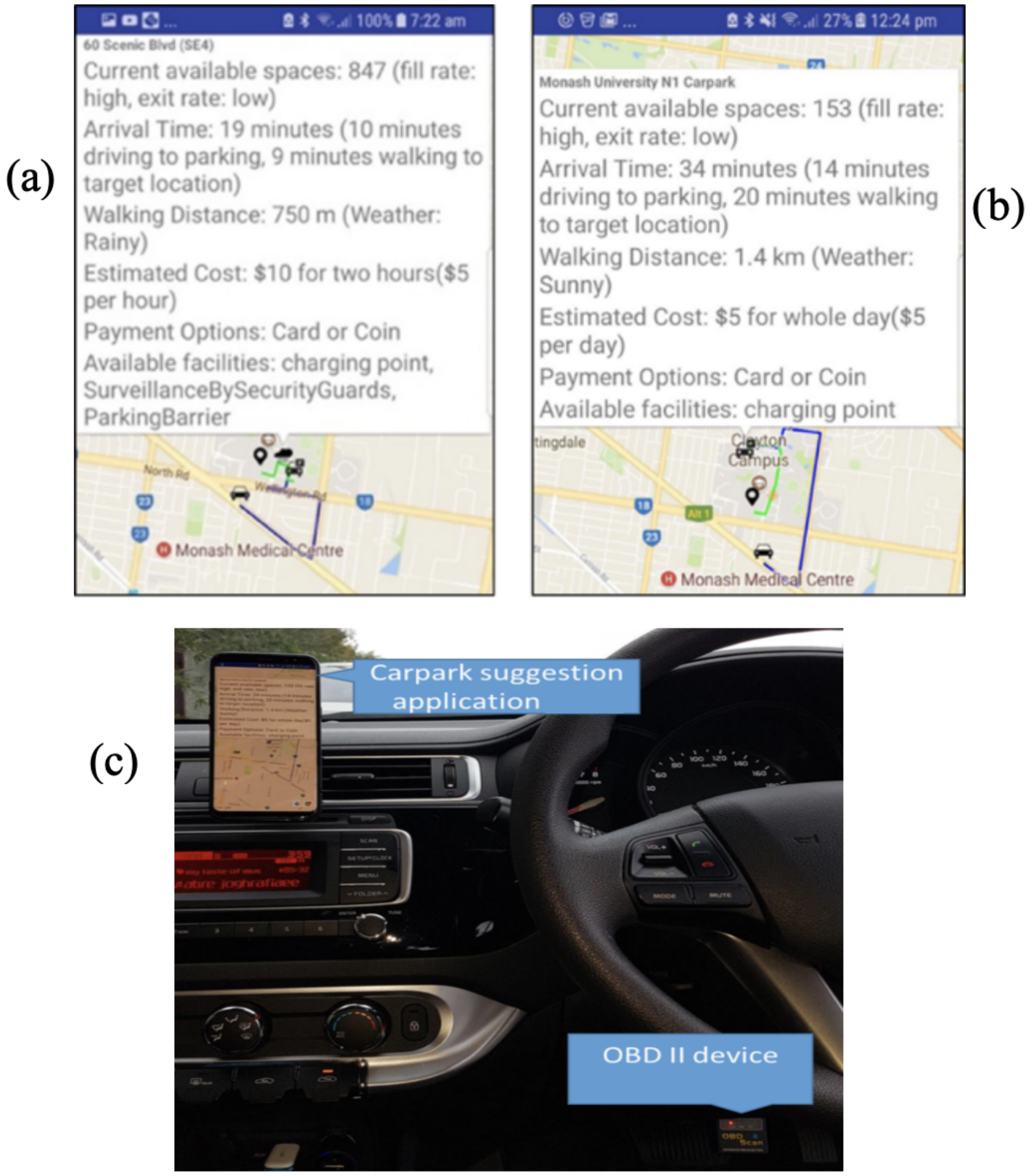


Figure 2.5: Smart Carpark Recommender PoC [3].

In [4], which is another smart parking application focused on collecting contextualized data for each driver, IoT data, and car park services data (see figure 2.6), contextualization methodology is applied to the collected data to provide parking recommendations. The experiment's results showed that the contextualization produced a three-fold reduction in query response time compared with other approaches. Another study [94] focused on using a mobility RS where each agent was

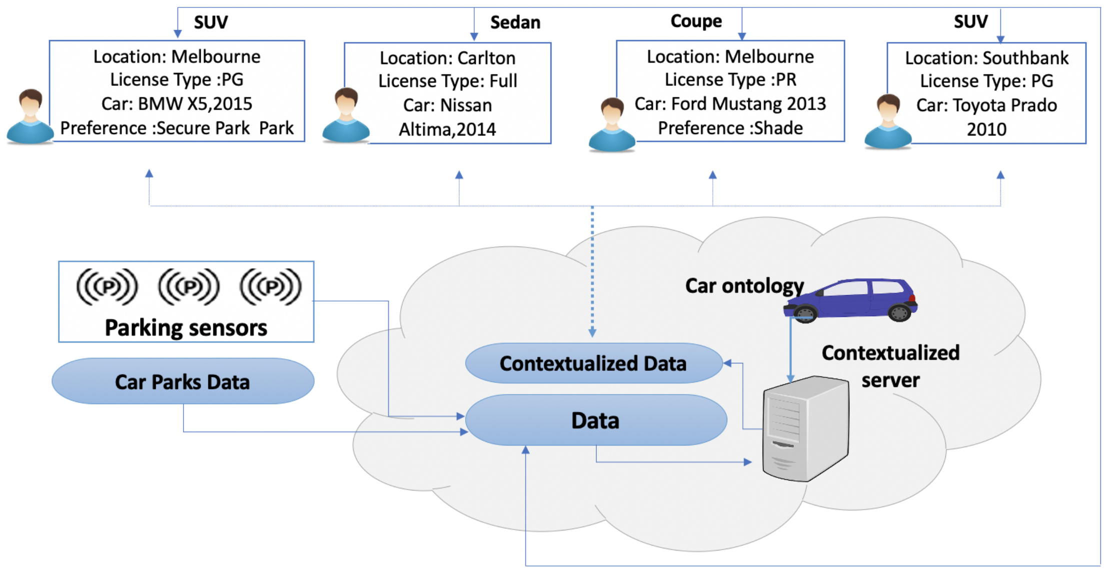


Figure 2.6: Contextualization Architecture [4].

responsible for providing a list of recommendations. For example, one agent provided only direct car parking spaces based on the destination. In contrast, another agent had city policy regulations, so the recommendations could be different from the previous one. The main feature of this application is that the user has several recommendations that consider his preferences from many perspectives. However, parking applications still face two main issues: using the nearest parking space instead of parking that is recommended and the recommended parking space being stolen. These two problems cause conflict parking. To address these problems, in [68], an application for parking conflict reduction was proposed that uses a periodical recommendation. Parking spaces that are available at different periods during the day are recommended to ensure accuracy.

2.3.3 Smart Tourism

Some recent tourism literature indicates that providing tourists with a unique and differentiated service has a positive impact on marketing. When tourists explore an area, they require high-quality services that will lead to memorable experiences. However, the dramatic increase in options available to them means tourists have difficulty making decisions. E-tourism RSIoTs address this issue by providing accurate recommendations. There are several tourism-based RS. For example, authors [123] designed a smart tourism application by exploiting two of the main sources to provide real-time recommendations: user mobility pattern and points of interest. The application sends recommendations that are suitable for each user and based on his or her location. Also, in [27], a tourism application is designed which recommends points of interest to visitors. User behavior is used as a source to reduce potentially false information that could affect the quality of the recommendations made by tourism applications. The idea of this application is that there are several media in different spaces that the visitor can watch by using the NFC tag. Each user has a specific ID, and the media are shown when the NFC tag is passed through the NFC reader. By monitoring and filtering all the interactions between the media and the visitors, the system can recommend points to visit that match the user preferences. The authors in [87] proposed an RS application that provides a list of artworks recommendations that are based on calculating the social affinity between the user and user experience. In [62, 124], authors proposed a recommender system application providing point of interest recommendations, with the exploit of the user history data and IoT information.

2.3.4 Personal Recommender System Applications

When a personal RS makes recommendations that resolve complex problems, the bridge between the user and objects is built. The design of this kind of system has become important, especially with IoT data. Accurate recommendations can be provided by exploiting massive amounts of IoT data and knowing when and what kinds of recommendations should be pushed. Our framework (see section 2.5) can be adopted to build a personal recommendations system. Let us consider the following scenario: if a user has a plan to save money during his work years, RSIoT can help him to achieve this goal by using data from his phone device and other kinds of sensors such as wearable ones. For example, accessing some emails related to shopping offers on his phone can help to recommend a good offer for him at a specific time, like after receiving his demographic information like income, and in a suitable place by accessing the user data location. In terms of saving food's budget, the system can recommend a restaurant with a good offer for lunch by using the history data about his favorite restaurant, their prices, and their locations. In previous studies, a number of personal recommender system applications were built. For example, an application (ProRec) [5] has been designed that supports the multitype context for recommendations. Three types are used: restaurants, gas stations, and attractions. When the user downloads the application, s/he can control it by using the options setting. The recommendations will be pushed to the user based on how close they are to the restaurants, rating, and other options (such as whether the restaurants are open at lunchtime). Figure 2.7 displays different screenshots of the application. Acceptance or rejection of these recommendations will affect the results. For example, if the user does not accept the pushed recommendations, the rating of these recommendations will be reduced.

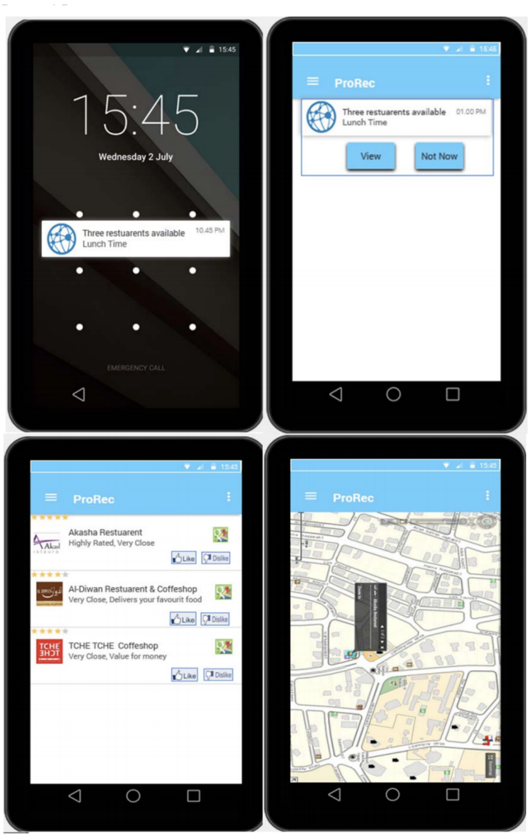


Figure 2.7: Screenshots for the prototype [5].

There are other applications that provide recommendations by focusing on exploiting the data of other applications. For example, the authors in [6] used a sample application scenario to show how social IoT provides benefits for IoT applications by using other applications' data. Several IoT applications are used: skiing, arranging meetings with friends, vehicle-to-vehicle communications, traffic monitoring, and demonstrations of wearable (see figure 2.8). An interoperability layer is responsible for enabling the interoperability between these applications so that the IoT RS can exploit the data to produce recommendations to the user. One of the most important features in this scenario is that IoT data can be exploited by different applications at the same time, and it can investigate scalability when large amounts of things are added to the network. However, some implementation challenges are identified in the paper. The work in [125] used the data from an email survey and

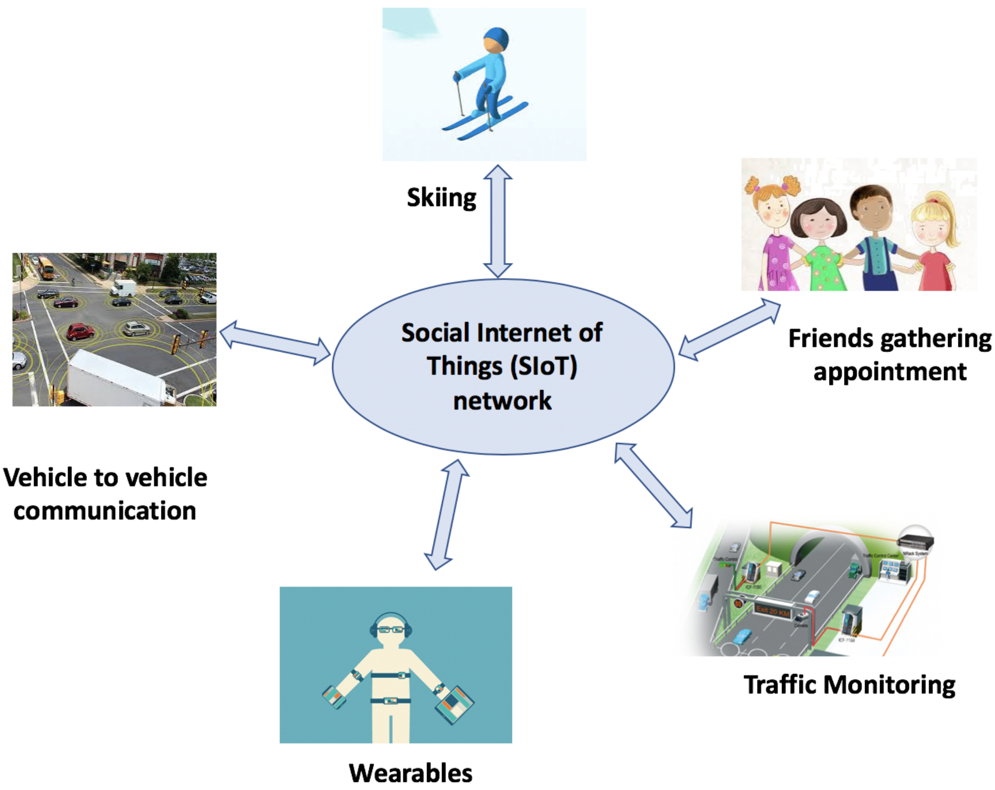


Figure 2.8: A sample application scenario in which different applications benefit from the SIoT by using other application's data [6].

traffic analysis to provide personalized recommendations to the user. The idea of this application is that the user has a chance to define their choices of interest, such as emotion, location, weather, and time. The system then monitors and analyses the smart devices' traffic to predict service recommendations that match their preferences. Another application [50] exploited IoT data, such as weather data, to provide several recommendations that are based on the weather. It uses sensory data to collect weather observation sequences and analyses them. Then it correlates current observations with previous ones to extract the trends that are predominant during this period and provides recommendations to the user. In [126], the authors designed an application that uses the IoT data to provide charging station recommendations for electric vehicles. With help from this application, time waiting at the charging station is reduced, the third party is not needed to ensure the security, reduced and simplified data is used, such as input such as GPS, and little information from the driver. Bhatnagar and Chandra [127] proposed a recommender system that can help farmers, agriculture scientists, and professions. The system promotes them to monitor the soil health data, including moisture, temperature, and pH.

2.3.5 Social Recommendations

Social recommender systems not only conduct item recommendations based on social network data but also help to create relationships between users based on their preferences [128]. Most applications for IoT are focused on exploiting social data to correctly define user preferences. For example, Amoretti et al. [7] built a UTrave application that recommends points of interest for visitors (shown in figure 2.9). This application consists of a mobile application side and a server side. The former collects personal data by using applications installed on the client's mobile devices; the latter leverages the collected data. The main feature of the application is that it could deal with the cold start problem; when the user sends a request recommendation for a new POI to the server, but it is not available, so the server starts to use the online data sources of the user (Demographic data) to recommend the nearest

POIs from the user. Yuan et al. [83] exploited some of the IoT data, such as tweet data (time, user, words, Geo), to provide accurate recommendations.

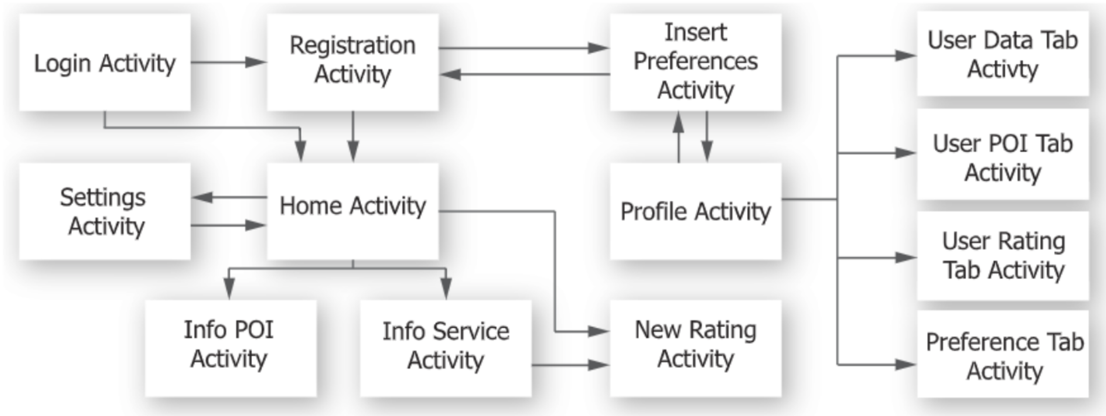


Figure 2.9: The structure of the UTravel mobile application [7].

2.3.6 Smart Homes

In the last few years, smart home applications have become increasingly popular; consequently, RS applications for smart homes have become available that help the user to choose the option that matches her/his preferences. For example, a heating schedule management application [100] is designed that exploits open data from the city. These are gathered by using the IoT infrastructure to provide heating recommendations for a specific user. The application considers several factors while compiling the list of recommendations. The user, for instance, can choose an application that automatically turns the heating on during a specific period. For example, a resident with a flat of 60 m^2 in a specific location might want to apply economic heating for the next ten hours. Another application [26] has exploited three kinds of correlations to produce accurate home recommendations: the relationships between users, the correlations among things, and the interaction between users and things.

Figure 2.10: Screenshot of the webpage. [8].

Also, in [129], the authors exploited conventional recommendation approaches for AGILE projects. These help the user to choose several kinds of recommendations; for example, recommending a suitable application for an installed device, suggesting some devices that may be required for some applications, or recommending some specific protocols which match devices. A recommender system [8] was designed that provides indoor comfort products recommendations for users. The system is divided into two main parts: the first is the data collecting part, achieved by using a network of sensors and extracting user preferences, while the second part involves using previous information to make product recommendations. Figure 2.10 shows a screenshot of the web page of the system, which provides the product recommendation at the top of the screen and the evaluation and a 'delete' button at the bottom.

2.3.7 Smart Marketing

In recent years, online shopping RSs have been developed to help customers by providing various guidelines. Most of these systems use conventional recommender approaches to providing recommendations for large websites, such as Amazon and eBay. However, with IoT, the recommendations could be made more attractive by exploiting IoT data as sources. A shopping-center application [73] has been designed that recommends healthy products to the customer by exploiting sensory data. The customer can also pay automatically using a digital wallet. An RS [70] is proposed which makes DS more interactive. The system exploits the interaction between the user and the DS to make the recommendations.

2.4 Analysis and Findings

Figure 2.11 shows the number of published works in two fields: cited works and citations for each year, from 2013 to 2020. The number of works that are related to RSIoT has increased each year steadily since 2013, with a surge of interest in 2017 and 2018. This surge represents a gradual shift in focus towards combining the internet of things with an RS. It also reflects the sharp perception of designing RS based on real-time recommendations. However, the number of in-field cited work have not kept up with the number of publications; particularly in 2019 and 2020. The main reason is that the majority of works might still not be detected by researchers, compared with those in other years, such as 2017 and 2018. While citation count may not be a perfect metric to evaluate the impact of research works, it does, however, show the extent to which a researcher's work is being acknowledged.

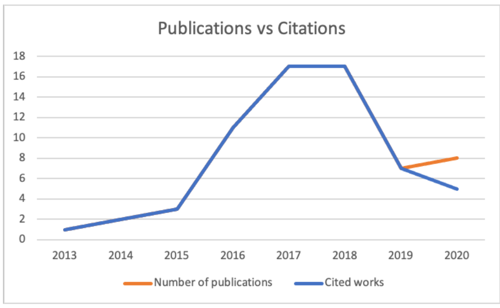


Figure 2.11: Number of publications, in-field cited works, and in-field citations.

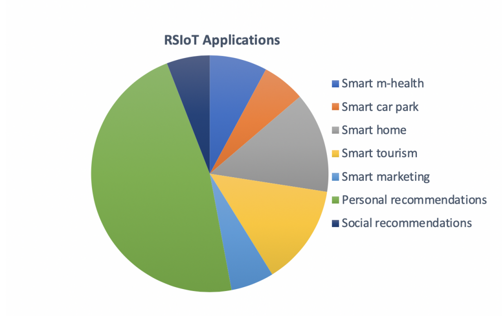


Figure 2.12: Proposition of RSIoT applications.

The following part reviews RSIoTs techniques and their applications. Table 2.7 shows the main recommendations techniques and applications. The main trends in RS for the IoT have been reviewed through studies published from 2013-2020. Some of the studies exploited conventional recommendation approaches, such as CF, CB, HB, and KB, in RSIoTs, but improvements in these approaches have addressed some of their limitations, which has had a positive impact on the development of other RS. However, as we mentioned in Aris's scenario, RSIoT should consider more than the interaction between users and items. For example, our RS will not recommend a cup of coffee to Aris just because she has entered her kitchen. It needs to collect more information before making any recommendation, such as time. Accordingly, using traditional approaches to make the recommendation could be inefficient for RSIoT. The RSIoT, based on the techniques that we discussed above, such as context-aware, Social IoT, multi-agent, and graph techniques, started to shift from recommendations that depend only on the interaction between the user and item as a resource to recommendations with more resources. For example, the context technique considers any information about the user and item, such as identity, location, state of people, etc., which provide rich input that helps an RS to make accurate recommendations.

However, the previous techniques were not able to learn the human pattern. In Aris's scenario, our RS should consider Aris's daily pattern, so preparing a cup of coffee in the morning would not be recommended if Aris changed her habit and it will consider a new activity instead. The last three techniques, which are machine learning, deep learning, and reinforcement learning, provide promising directions for RSIoT. We discussed in section 2.1 that exploiting knowledge of the human pattern plays an important role in conducting accurate recommendations. Reinforcement learning algorithms can address this issue when they can capture the user's temporal intentions and conduct recommendations in a timely manner. Most of the existing works focus on exploiting RL to learn the optimal recommendation strategy, which is based on the interaction between the user and agent or to increase the cumulative reward for each scenario. However, adapting reinforcement learning for RSIoT faces major challenges in terms of dealing with the human pattern in different scenarios. For example, in smart homes, RSIoT will face several scenarios during the day, as conducting recommendations during the morning period should be different from other periods, such as noon or the evening. Accordingly, this issue could be handled by combining reinforcement learning with the multi-agent approach, which would capture the sequential dependency of the human pattern in different scenarios. Also, agents will have the same memory of the history of the human pattern and work collaboratively to improve the performance of the RS. Table 2.2 summarizes the advantages and drawbacks of each technique.

Regarding the applications, we discussed seven major RSIoT applications that mostly fell in the smart home and personal domains, as shown in figure 2.12.

Technique	Advantages	Drawbacks
Collaborative filtering approach	- Focusing on conducting recommendations based on the similarity between users. - No knowledge domain is required. - Increasing of users leads to improve recommendations.	- Scalability - Cold start - Spars data
Content-based approach	- Creating a profile for each user which includes previous items. - No cold start problem for new items where the system can match their features based on previous items were rated by the user. - Conducting recommendations based on user preferences only to ensure transparency.	- New users have no information to build their profiles. - Limited information to conduct recommendations. - Lack of novelty because of focusing on matching features of the profile with items.
Hybrid approach	Combining two approaches such as content-based with collaborative may tackle their issues.	Focusing only on the interaction between user and item.
Knowledge-based approach	- Ability to detect any change of the user preferences. - Reliability. - Focusing on the recent data. - Using rule engine.	- Failed in dealing with decisions that do not belong to the system rules. - Requiring knowledge engineering. - Ignored the history data which could affect the recommendation quality.
Context-aware	Improving the quality of recommendation	- Scalability - Spars data - Privacy - Security
Social IoT-based Recommendation Technique	Exploiting the relationships between users- things , and between things and things.	- Interoperability - Privacy - Security
Multi-Agent Algorithms	Scalability	- Diverse relations - Dynamicity
Graph Database Model	- Showing objects with details. - Queries can be referred directly to a graph structure. - Graph databases can provide an efficient graph algorithms to investigate specific operations and efficient graph storage structures.	- Lack of dealing with time-series data - Generic frontends
Machine learning	- Discovering all possible relationships. - ML improves with increasing data.	- Sufficient data is required to fit the system. - Consuming time, - Interpretability. - Cold start problem.
Deep learning	- Dealing with complex interaction pattern. - Providing a good data representation using deep neural networks. - Flexibility which provides various tools to build a model.	- Interpretability where there is a lack of explainability in the hidden weights and activations of deep neural networks. - Sufficient data is required to fit the system. - Sensitivity of the hyperparameters.
Reinforcement learning	-Treating the interaction between the user and the agent as the main procedure for a recommendation - Learning the optimal policy to increase the cumulative reward without any predefined instructions.	- Lack of the ability in handling a system with different scenarios - The complexity of reward formulation. - The large space issue in the sequential interactions.

Table 2.2: Trade off among RSIoT techniques.

2.5 A Unified Recommender System Framework for the IoT

Defining a unified RSIoT would be of great benefit to researchers and professionals. Figure 2.14 presents such a framework that is based on state-of-the-art knowledge of RSIoT. The framework focuses on the following four main stages:

2.5.1 Data acquisition platform

Preparing an efficient platform for sources that are used to feed the RSIoT is crucial. There are three basic steps required to create an RSIoT platform. The first step is to turn physical things into smart objects. We can define smart objects as objects that can provide information and data about themselves or other related objects. They are also capable of communicating this information [130]. There are two mainstream ways to turn physical objects into smart things: tagging the object with RFID tags and embedding the sensors. However, the most distinctive issue is how to deal with the resource constraint of physical things. To address this problem, it is feasible to embed each physical thing to a server so that they each have communication, storage, computation, and data processing abilities and can provide services (e.g., smartphones, sensor nodes, and RFID readers). RFID and Wireless Sensor Networks are two main enabling technologies for bridging physical things to the virtual world. In the next step, every smart device (thing) should be connected to the Internet. Things can be connected, wired, or wirelessly. In RSIoT, a wireless connection will mainly be used. There are many ways to connect a smart device (thing). Based on the existing infrastructure, these are RFID, ZigBee, WPAN, WSN, DSL, UMTS, GPRS, Wi-Fi, WiMax, LAN, WAN, 3G, etc. The last step is to employ specific network protocols that are used in sensor networks and RFID networks to make them connect and communicate via the Internet. A number of IoT protocols have been proposed which consider sensor restrictions, such as limited energy and computation capability. For example, Message Queue Telemetry Transport (MQTT) [131] is used for M2M and IoT applications that need a lightweight subscribe and publish messages. The Constrained Application Protocol (CoAP) [132] works as a web transfer protocol for constrained nodes and M2M applications. Other IoT protocols for outing/forwarding have also been proposed that are designed for peer-to-peer communications or client-server communications, such as IPv6 over Low Power Wireless Personal Area Network (6LowPAN) [133] and WebSocket [134]. Sensor networks and RFID have their own communication stack; 802.15.4 is widely used for sensor networks, while RFID usually supports the 802.11 standards to the wireless network on the Internet. However, the IoT still suffer from different challenges in terms of battery constrained, storage capacity, network limitations, and redundancy issues. We require a number of strategies to reduce their effects, particularly when RSIoT has to deal with a large amount and continued sensory data. One of these strategies is an IoT offloading methods could be adapted to reduce the limitations of the IoT in terms of battery constrained, storage capacity, and network limitations. It sends the IoT data that need powerful computation into proficient devices such as cloud, fog, and edge and then receives the results of them again. Authors in [22], outline IoT offloading mechanisms by dividing them into three categories: based on architecture, based on partition, and based on decision-making processes. Also, data aggregation strategy plays the main role in tackling some of the IoT limitations. It provides an effective process in collecting the data, which reduces the energy consumption, ensuring the quality of data, increasing network lifetime, and removing the data redundancy. In[23], a systematic review of the literature for data aggregation mechanisms is explained extensively. A resource allocation mechanism [24] also has an important role in addressing the IoT issues and aims to manage loading data, reduce computational complexity, and save energy. As a result, the RSIoT system performance is improved as well as the quality of services.

2.5.2 Data processing for rich information

The data processing phase is required for rich information, as it ensures that the data can be used effectively for further processes. It has a critical impact on the accuracy of results and is therefore crucial for the success of the whole system. After the data is collected, as we mentioned previously, it may contain noise, missing values, and redundant features. Consequently, a number of sensor data processing methods can be adopted, which are classified into three main parts: preprocessing, segmentation, and dimensionality reduction. The first part includes operations to clean the data, dealing with missing readings, and transforming the data. Usually, the sensor's readings are converted from one measurement to another, such as from voltage to temperature, and this process can affect the data [135]. In order to tackle this problem, the sensor is recalibrated [136] or adapted via data-driven modelling [137]. In addition, the data from RFID is nosier than the data from a sensor because of errors related to reader tag communications and redundant data. Several techniques can be used to clean RFID data [138] or deal with losing RFID data [139]. In the segmentation part, the raw data from sensors continuously flow and therefore needs to be divided into smaller parts. The selection of the proper segmentation approaches has had a huge effect on the last part of the data processing. The authors in [140] categorized the segmentation approaches into three main types: temporal-based segmentation, activity-based segmentation, and sensor event-based segmentation. Dimensionality reduction plays the main role in extracting and selecting features from the raw data by using different approaches. In [141], the authors proposed preprocessing mechanisms that could handle any errors of raw data, specifically, accelerometer sensors. Sensor signal processing techniques are classified into three main domains: the time domain, which contains mathematical and statistical metrics to extract the basic signal information from sensory data; the frequency domain, which is responsible for capturing the repeated nature of the raw sensor data; and the discrete representation domain, which transforms the signal sensor into a string of discrete symbols.

2.5.3 Event generator and Rule composer

In a system that recommends things to users, we intuitively need to first understand what kinds of things the user prefers. There are two main tasks that should be implemented at this stage: (1) generating the events based on previous information and (2) defining a suitable rule for each event. In the first task, RSIoT should be able to extract an event from the extracted features at an earlier stage. The event depends on the kinds of applications and the extracted features. For example, in Aris's scenario, the system can detect that she prepares a cup of coffee in the kitchen at midnight. Three data sources are exploited to extract this complex activity event: activity recognition (standing), localization (the kitchen), and object usage detection (the cup of coffee). Several probability-based algorithms have been used to implement this task. The HMM can be adapted for activity recognition [142]. It can generate a hidden state based on the observable data and learn reliable model parameters from the history of the model output. However, it still has some limitations, such as in representing the number of interactive activities, capturing the long-range of the observations, and the failure in recognizing the available observations for a consistent activity [143]. Another model that could also be used is the conditional random field (CRF) [144]. This is considered to be more flexible than the HMM because it focuses on extracting the conditional probability instead of a joint probability distribution, such as HMM. ML models have been used for activity recognition such as Naive Bayes classifiers [145, 146] and decision trees [147]. Ontological models also help to recognize complex activities [148, 29]. In Aris's scenario, an ontological model could be built to represent human household activities and environmental domain concepts and objects. Another task for this stage is a rule composer. This contains a set of rules extended from the previous task and combined constraints. It is responsible for defining suitable recommendations based on each event. For example, Aris will not receive any recommendations about preparing coffee or certain kinds of food when she is in the kitchen at midnight. Also, the rules have a flexible nature that can be updated based on new events that are generated by the system. A number of efforts have been made to tackle rule modelling [149], build rule engines [150], and design techniques that enable rules for automatic learning [151].

In our previous work, we started [29] to cover this stage of the framework. We proposed a reminder care system that focused on detecting complex activities for the user and then providing reminder recommendations for patients with Alzheimer's disease. The framework is divided into three main stages: (1) Correlation analysis of devices, (2) Rule-based orchestration, and (3) Activity-triggered Recommendation (see figure 2.13). We adapted one of the deep learning algorithms, DeepConvLSTM, which is a state-of-the-art deep neural network that combines convolutional, recurrent, and softmax layers for the first stage. Experiments showed that the behavior recognition part of our system is effective, and hence the recommendation engine is practicable.

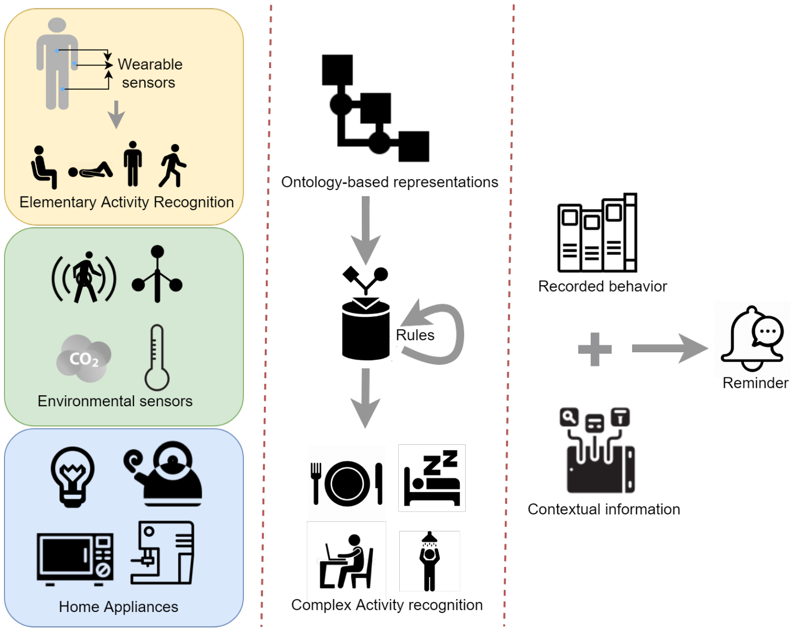


Figure 2.13: Reminder care system framework.

2.5.4 Accurate recommendations production

This is considered to be the core stage of the framework because RSIoTs depend more on context than conventional recommendations that are built based on web data. Consequently, the techniques used to design RSIoT can be more complex. This complexity is mostly related to the limitations of IoT, as identified in this article. However, the main challenge is how the RS can learn the human activity pattern in order to push the recommendations matching user preferences. Each user has a different pattern, and such patterns could be evolving with a certain period of time: A user who prefers to drink coffee in the morning on weekdays may sleep on the weekends. In our previous work [29], we adapted Non-negative Matrix

Factorization(NMF) for recommendations. However, this technique still suffers from the limitations as aforementioned in section 2.2.1.

The RL algorithms could address this issue with the ability to capture the user's temporal intentions and make recommendations promptly. However, as we mentioned previously, RL also faces some issues. For example, most RL algorithms attend actions based on the effects on the next states, and such dependence is not always desirable. Our study shows that RSIoT has to handle each state independently in some cases, to ensure that the user receives a suitable item based only on the current state. A contextual bandit approach for recommendations [152] combines two main common features of RL, which uses policy to take actions based on the context of the current state and of multi-armed bandit(MAB) by focusing on the immediate reward. Authors in [152], proposed a contextual bandit algorithm for news articles, which exploits the user and the article context to conduct recommendations. The result showed that the proposed algorithm outperforms the standard context-free bandit algorithm. Currently, this approach is adapting for our RSIoT to conduct automatic and accurate recommendations without requiring users' feedback.

2.5.5 Privacy and Security in RSIoT

The RSIoT environment deals with a massive amount of data, especially about the user, in order to improve the quality of recommendations. However, collecting these kinds of data, such as health data, location data, or even daily activities, creates a number of privacy concerns for users. In addition, devices could be targeted by malicious attacks that affect the accuracy of recommendations. Consequently, several requirements and parameters need to be reckoned, which are described below. Privacy and integrity mechanisms should be adopted because IoT devices and their data are susceptible to malicious attacks that can affect or change the data. Authentication techniques, mainly for RSIoT, are also required due to the heterogeneous nature of IoT devices, and there is a need to ensure that the access system is only available to authorized devices. The quality-of-service (QoS) should be investigated by avoiding different attacks, such as sinkhole attacks and jamming adversaries. The efficiency of energy consumption is another requirement that needs to be considered, particularly as IoT devices have some constraints with their storage, energy, and computation capability; thus, an attack on the system might increase these issues. RSIoT architecture also encompasses a wide variety of devices, starting from small devices (sensors) to large ones (servers); thus, there is a need to address security issues at different levels [153]. Recently, there has been a tremendous effort to cope with privacy and security issues in IoT. The authors in [154] explained the role of secure multi-party computations in preserving privacy for IoT users. Also, in [155], the authors explained IoT's main security issues and highlighted some future work solutions that could tackle these issues. With regard to the security issue, the studies can be classified into two parts: one targets the system's specific security issue [156, 157] and the other focuses on providing security for the whole system [158, 159]. However, addressing privacy and security issues for RSIoTs has received little attention. The authors in [48] proposed an RS that investigates trust and reputation among IoT nodes by using PNN. It can provide different levels of security according to the sensitivity of the data. However, it only focuses on providing a mechanism to carefully select IoT resources regarding their reliability. In our previous study [121], we proposed a holistic framework to ensure the integrity of data in smart cities, which covers the entire data lifecycle, using blockchain and other techniques. Blockchain (BC) technology can provide an efficient solution to the problem of protecting IoT recommendations. It provides a decentralized peer-to-peer network that applies stored encrypted and secure computations to the raw data. Moreover, it provides the user with authorization to access her raw data or allow another party to provide recommendations without displaying her private data.

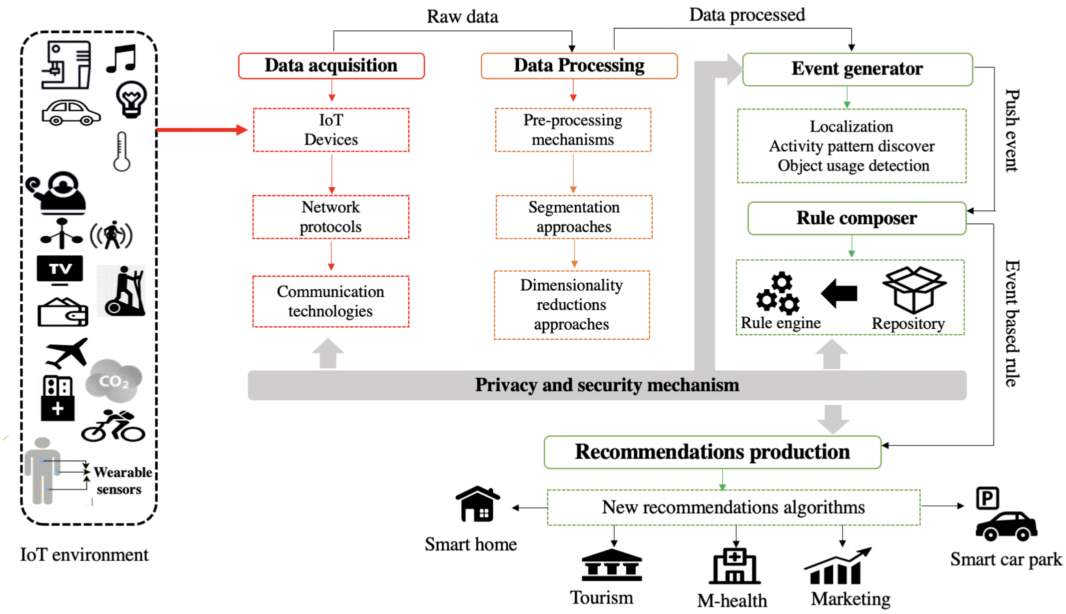


Figure 2.14: Proposed framework for RSIoT.

Perspective	Reference	Technique	Application	Thing of interest	Key contribution
Conventional recommendations techniques	Yao et al.[26]	CF,graph, random walk	smart home	device	designs a unified CF approach based on matrix factorization which exploits three kinds of correlations
	Asiri et al.[48]	CF,PNN	unknown	device	adapts PNN and CF to propose a recommender system based trust and reputation model
	Chakraverty and Mithal [50]	CF,HMM	personal	service	addressing two main problems: model weather conditions in short term using HMM and predict users preferences using CF
	Sawant et al.[52]	CF,CB	personal	service	combines cyber-physical systems and IoT to conduct recommendations based on users preferences .
	Lee and Ko [53]	user-based CF	personal	service	propose user based-CF to provide group recommendations by considering the member organization.
	Mashal et al.[54]	UBCF,OBCF ,bipartite graphs	personal	service	uses bipartite graphs and two algorithms based on CF to address the problem of recommending IoT third party services
	Cui et al. [64]	CF	personal	service	adapts collaborative filtering to build recommender system but with considering the connections between user preferences and time and dividing the users into groups based on their similarity
	Nizamkari [57]	graph-based trust ,CF(optimized)	personal	device	Addressing the common problems for traditional CF by using graph-based trust
	Salis et al.[49]	ML,CF	personal	service	adapt ML and CF to provide real time recommendations
	Jabeen et al. [58]	Advice-based CF, ML	personal	service	uses ML classification algorithms to detect cardiovascular disease and proposes advice-based CF approach to conduct a suitable recommendations based on the classification result

Table 2.3: Categorization by techniques type,applications domains and recommendations

Perspective	Reference	Technique	Application	Thing of interest	Key contribution
Conventional recommendations techniques	Li et al.[59]	CF	personal	service	enhances the the basic matrix factorization model and then build the trust relevancy which addresses the data sparse problem
	Margaris and Vassilakis [60]	CF,QoS	smart tourism	service	improves the quality of the recommendations by combining two algorithms : CF and QoS
	Yang et al.[62]	CF	personal	service	proposes a location-aware POI recommendation system
	Rossi et al. [63]	CF	smart tourism	service	Employs CF approaches to provide artworks for both individual and group visitors
	[71]	HB	personal	service/ product	uses mobile edge environment rather than center cloud to enhance the quality of service recommendation
	Erdeniz et al.[65]	CB	personal	service	proposes a new approaches to build recommender system in m-health
	Koubai and Bouyakoub [67]	CB	personal	service	proposes application which facilitates the work of employees in a restaurant and improves users experience
	Srisura and Avatchanakorn [68]	CB, Constraint-based	smart car park	service	Proposes a smart parking recommender system which address two main problems that face parking applications by providing a periodical recommendation
	Gyrard and Sheth [79]	KB	smart health	service	designs a recommender system, IAMHAPPy, to provid well-being recommendations using groups of ontology a smart parking recommender
	Yang et al. [62]	CF	smart tourism	service	designs an algorithm for recommender system which has ability to provide location recommendation to the user even in a new region

Table 2.4: Categorization by techniques type,applications domains and recommendations(continued)

Perspective	Reference	Technique	Application	Thing of interest	Key contribution
Conventional recommendations techniques	Tu et al. [70]	HB	smart marketing	service/ product	adapts HB to build recommender system engine which makes DS more attractive
	HamlAbadi et al. [72]	HB	smart marketing	products	adapts a cognitive system to build recommender system in IoT
	Hwang et al. [75]	KB	smart home	configuration rule	uses ontology and open data to build recommender system for smart home
	Kumar et al. [76]	KB	Personal	service	proposes a conceptual framework for space service recommendations
	Varfolomeyev et al. [77]	KB	smart tourism	service	implement recommender system for historical tourism by adapting ontology formats
	Martino et al. [78]	KB	personal	srevice	builds recommender system using ontology for smart health field
	Ali et al. [80]	KB	personal	service	adapts fuzzy ontology to build recommender system which recommends specific food for diabetes patients
	Felfernig et al. [129]	UB	smart home	software	apply the utility-based approach in an AGILE project to recommend technologies to a user who employs the AGILE gateway
Context awareness	Salman et al. [5]	multi-type context-aware	personal	service	exploits context-aware to build recommender system which provides multi type of recommendation
	Hassani et al. [3]	CSDL,CSM	smart car park	srevice	propose a Context-as-a-Service (CoaaS) recommender system
	Yuan et al. [83]	contextual information	social	service	exploit contextual information and mobility of the user's tweets to provide accurate recommendations.
	Yavari et al. [4]	contextual filter, contextual aggregation	smart car park	service	exploits contextualization with IoT data to provide accurate recommendations.
	Zhou et al. [84]	HSCT algorithm	personal	service	proposes a model which exploits context-awareness to improve recommendations accuracy
	Hong et al. [87]	Context awareness	smart tourism	service	proposes a recommender system that provides social recommendations for cultural heritage.

Table 2.5: Categorization by techniques type,applications domains and recommendations(continued)

Perspective	Reference	Technique	Application	Thing of interest	Key contribution
Context awareness	Kaur et al. [85]	Context awareness	smart kitchen	service	adapts context as sources to build a recommender engine for recipe recommendations in a smart kitchen
	Casino et al. [86]	Context awareness	personal	service	proposes a recommender system which provides health recommendations
Social IoT	Saleem et al. [6]	SIoT	personal	service	exploits socialization between thing to produce service recommendations
	Chen et al. [90]	SIoT	unknown	device	provides recommendations about trustworthy node
	Ren et al. [91]	SIoT	social	service	the mobile IoT was exploited to build a recommender system for services and social partners
Multi-agent system	Forestiero [93]	MAS	unknown	device	adapts a multi agent algorithm to improve the recommendation's speed.
	Martino and Rossi [94]	MAS	smart car park	service	builds a Mobility Recommender System(MRS) for parking
	Twardowski and Ryzko[95]	MAS	smart m-health	service	exploits the MAS to build a recommender system which uses the data of mobile devices and some IoT devices surrounding them to provide personalized recommendations in real time.
	Jiménez Bravo et al. [96]	MAS	smart home	service	proposes a recommender system based on the multi-agent technique to optimize electricity consumption and save cost in a smart home.
	Souza et al. [97]	MAS	unknown	service	Propose a recommender system to provide physical activities and eating recommendation to children who have obesity .
Graph Database Model	Palaiokrassas et al. [100]	Neo4j graph	smart home	service	exploits a Neo4j graph database to address one of the main challenge in IoT, namely, big data management.
	Noirie et al. [101]	Typed Attributed Graphs	unknown	service	exploits graph techniques to build a recommender system which provides IoT services recommendations to the user based on their own IoT devices.
	Wang [102]	tripartite graph	smart environment	service	exploits tripartite graph and matter diffusion to build recommender system scheme for IoT services recommendations.

Table 2.6: Categorization by techniques type,applications domains and recommendations(continued)

Perspective	Reference	Technique	Application	Thing of interest	Key contribution
Machine Learning	Sewak et al. [105]	Distributed Kalman Filters, Distributed Mini-Batch SGD , and Distributed ALS based classifier	unknown	unknown	adapts some machine learning algorithms to build the Optimal State based Recommender(OSR)
	Guo et al. [106]	RBF,D-S	smart marketing	products	proposes a framework to build an e-commerce recommender system that exploited the multisources of information
	Massimo et al. [27]	IRL	smart tourism	service	proposes inverse reinforcement learning (IRL) to model user behaviour to improve the quality of recommendations.
	Gutowski et al. [119]	LinUCB	personal	service	uses a reinforcement learning algorithm to build framework that provides context-aware recommendations in a smart city.
	Oyeleke et al. [120]	RL	smart home	service	proposes a system too recommend the correct sequence of tasks for each activity to ensure that the user can reach his goal.
	Asthana et al. [107]	a machine learning classifier	smart m-health	device	exploits a machine learning classifier to build a recommendation engine which provides personalized wearable technologies recommendations for proactive monitoring.
	Amoretti et al. [7]	K-Means, UPR	social	service	adapts K-Means algorithm in UTrave recommender system application which clusters user profiles to recommend points of interest for the user.
	Rasch [108]	unsupervised learning algorithm	smart home	service	adapts unsupervised learning to build recommender system for smart home.
	Yoo and Chung [2]	decision tree	smart m-health	service	uses a decision tree to build a system which provides lifecare recommendations.
	Valtolina [109]	decision tree and social network	unknown	service	adapts both the decision tree algorithm and social network to propose a multi-level recommendation system
	Rizvi [110]	AHP	smart car park	service	adapts Analytic HierarchyProcess (AHP) to build recommender system for car parking recommendations.
	Ayata [111]	random forest, kNN and decision tree.	personal	service	adapt ML algorithms to design a recommender system for music recommendations with more accuracy than traditional ones.
Deep Learning	Yong [1]	3D CNN	smart m-health	service	utilizes deep learning technique to build an intelligent system for fitness club.
	Hashemi and Kamps [114]	MLP	smart tourism	service	a deep neural MLP has been adapted to improve effective system recommendations for smart museum
	Han et al. [115]	BAS,CNN	smart health	Personal	proposes a platform for recommender system that conducts the rehabilitation nutrition plan recommendations for Cancer patients
Reinforcement learning	Massim et al. [118]	Inverse reinforcement learning	smart tourism	service	models user behaviour as away of improving the quality of recommendations by using IRL.
	Gutowski [119]	LinUCB	personal	service	designs a framework that provides context-aware recommendations in a smart city.
	Oyeleke et al. [120]	RL	smart home	service	builds a recommender system to support patients who have mild cognitive impairment

Table 2.7: Categorization by techniques type,applications domains and recommendations(continued)

2.6 Conclusion

In this chapter, we present a comprehensive survey to summarize the recommender system for the internet of things techniques. We review the development of applications for RSIoT in a variety of domains. We provide analyses with findings for the state-of- the-art methods. We provide a reference framework to compare the existing studies and guide future research and practices. The previous studies have been credited to highlight the challenges that need to be considered in building RSIoT. Notice that the following mentioned challenges are only the ones that we focused on in the next chapters:

- Human complex activity recognition which considered as main source to feed the recommender system. Knowing the current situation for the user helps the system to decide the correct item to be recommended.
- Capturing dynamicity patterns of human activities. As we mentioned before in Aris scenarios, each activity could have different pattern even if the same activity includes the same items.
- Harvesting the feedback automatically in the back end without requiring users to explicitly label activities. Specially in our use case, the system has to interact with Aris without asking her to hold a phone or providing her feedback.
- Data integrity threats may affect the accuracy and consistency of recommendations; particularly, with critical cases that need to receive confidential recommendations.

Chapter 3

Reminder Care System (RCS): An Activity-aware Cross-device Recommendation System

This section contains work published in:

[29] **M. Altulyan**, C. Huang,L. Yao, X. Wang, S. Kanhere, and C. Yunajiang, "Reminder Care System: An Activity-awareCross-device Recommendation System", in *2019 International Conference on Advanced Data Mining and Applications*, pp. 207—220, November 2019.

3.1 Introduction

Recent studies in human activity recognition have enabled various smart applications such as m- health [95], smart homes [26], smart tourism [123]. It is fundamental to humanity, since the collected data can feed the computing systems to be able to monitor, analyse and to assist users in their daily life. There are two kinds of systems for human activity recognition: video-based systems and sensor-based systems. The first kind depends on cameras to monitor the human behaviour. The second one depends on sensors that are attached to the human body or the area around them. In term of privacy, the sensor-based system provides more privacy and it also investigates the same goal for the first one. Thanks to the Internet of Things, myriad devices can be connected to the internet by embedding sensors. Various sensors around us have ability to log human's motion data without intrusive or interrupt.

There are enormous research works use various kinds of algorithms in human activity recognition area, most of them focus on elementary activity recognition especially with wearable sensor. In [160], a hierarchical model was proposed to detect the complex human activity where the elementary activities are predefined using the time series patterns. Yan et al. [161, 162] proposed a classification framework which consists of two tiers to extract the required features of elementary activities to describe the complex ones. Two drawbacks could be obvious here: (1) predefined the elementary activities needs deep knowledge and (2) depending on the features of elementary activities only leads to lack of the precisely and effectively detection for the complex ones. Peng et al. [163, 164] adapted used k-means clustering and topic model to define the complex activities components and to discover the latent semantic for each complex activity respectively.

Deep learning has highlighted promising results in many areas including HAR. Authors in [116] presented a survey for the state-of-the-art deep learning methods based human activity recognition. Moreover, challenges are related to each method were summarized and analysed.

DeepConvLSTM is a deep neural network which consists of three kinds of layers: convolutional layer to extract the features from sensor data, LSTM layer to process the sequential data, and the softmax layer which represents the output of the model. In [29], we model the first stage of our system using DeepConvLSTM.

The proposed system presents the design and development of a prototype system, RCS, for a smart home monitoring process that produces remind recommendations. RSC has the ability to automatically learn the contexts in a smart home environment, such as object usage and daily human activity, by analysing the motion sensor data generated from human movements and interactions with objects. It enhances Alzheimer's patients' awareness of their surroundings and provides better recommendations. In addition, the proposed system enables to create personal rules. Several IoT applications can benefit from the architecture and implementation of our system. So, detecting the elementary and complex activities was the target in this chapter which represents the main source of input to conduct high quality recommendations. During this work, we divided our methodology into three stages. However, this chapter will cover the first and second stages in details and the last stage will be highlighted extensively in chapter 4 and chapter 5.

The main contributions of this chapter are summarised as follows:

- We utilize DeepConvLSTM for elementary activities recognition with wearable sensors, which later been used as one of the features for complex activity prediction in our end-to-end reminder recommendation system.

- We build ontological models to considering spatial, artifactual and environmental contextual information, to boost our complex activity recognition by producing rules.
- We conduct experiments on two public datasets, and our experimental results demonstrate the feasibility of our system.

3.2 Related work

3.2.1 Smart home applications

A great deal of research has sought to identify and exploit Activities of Daily Living (ADLs) in the smart environment to help elderly people to complete their activities or reach their goal. The authors in [165] presented an end-to-end web based in home monitoring system, namely WITS, for convenient and efficient care delivery. The system exploits both the data and knowledge driven techniques to enable a real-time multi level activity monitoring in a personalised smart home. In [166], exploited a fuzzy temporal data-driven technique to design an anomalies recognition and assistance provision system. The work in [167] proposed a SmartMind system to monitor the activity of Alzheimer's patients. However, this system has several limitations in addition to cost where the patient still depends on the assistance of a caregiver to guide them to complete their activities or reach their goal.

Oyeleke et al. [120] proposed a system, SML, to track the indoor daily activities of elderly people with mild cognitive impairment. It aims to recommend the correct sequence of tasks for each activity to ensure that the users can reach their goal. Latfi et al. [168] proposed an ontological architecture which consists of seven ontologies for the Telehealth Smart Home. It aims to build high-level intelligent applications, particularly for older people suffering from loss of cognitive autonomy .

3.2.2 Deep learning for human activity recognition

Recently, deep learning algorithms have demonstrated their effectiveness in different areas, plenty of them have been exploited to tackle the issues in activity recognition. In [169], author proposed a system based on deep learning to monitor elderly people during their daily life activities. The system consists of three main stages: extracting features for the activity recognition using a deep neural network(DNN), distinguishing the normal from anomalous activities using an overcomplete-deep autoencoder (OCD-AE), and predicting the next activity using a long short-term memory (LSTM) algorithm. The performance was evaluated using real smart home datasets. Authors in [170] proposed an activity recognition approach to match labels to activities instances and to discover anomalies. Here, the Probabilistic Neural Network (PNN) was adapted to extract features and the H2O autoencoder to detect the the anomalies. The proposed system was evaluated using two publicly Smart home datasets. In [171], authors developed a hybrid model which combined two networks: CNN to extract the features and LSTM to learn temporal pattern. The results show the validity of the developed model for HAR applications. In [172], both traditional machine learning and deep learning algorithms was exploited to monitor activities of elderly people. However, LSTM algorithm shows satisfying results compared with others. In [173] LSTM was adapted for HAR; the model consists of one hidden layer and 200 units. The main advantage is that the model can be able to extract the features without using any techniques unlike the other two conventional algorithms: RF and HMM. The results show that LSTM achieved around 75% compared with RF and HMM algorithms.

Most of previous works focus on the interaction between user and items as main source for activity recognition. However, in our work, in addition to exploiting the interaction between user and things, we extend the model to the relational network of three main data sources: localization, user and thing interaction and simple activities. This improves performance in detecting latent features from the relational network of these sources that can be used to define complex activities.

Our proposed approach has two obvious advantages over the related work discussed above: (1) the context in a smart home environment can be learnt automatically by the system (2) users have ability to create personal rules through a graphical interface by exploiting the interaction with user, objects and, contextual events.

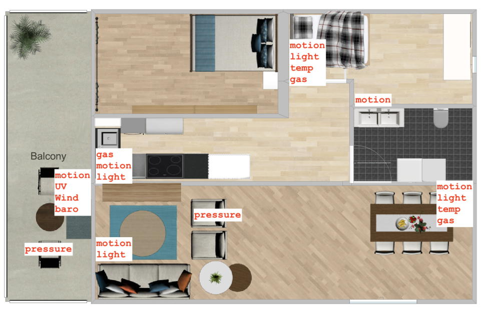


Figure 3.1: Example of sensor deployment in home environment

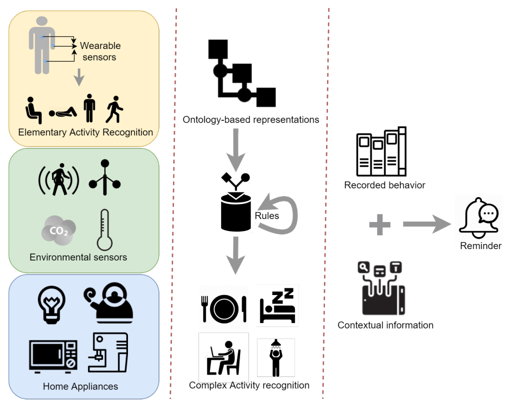


Figure 3.2: Overview of the proposed methodology

3.3 Reminder Recommender Care System

In this section we describe the three major part (see figure 3.2) of our proposed system for reminder recommendation: (1) *Correlation analysis of devices*, where we construct a general ontological model for representing human domestic activities, domain concepts and objects; (2) *Rule-based orchestration*, where strategy will be developed based on a semantic distance-based rule matching method and rule generation; and (3) *Activity-triggered Recommendation*, which utilized non-negative matrix factorization technique to exploit information learnt in the previous two steps, to give reminder recommendations to users.

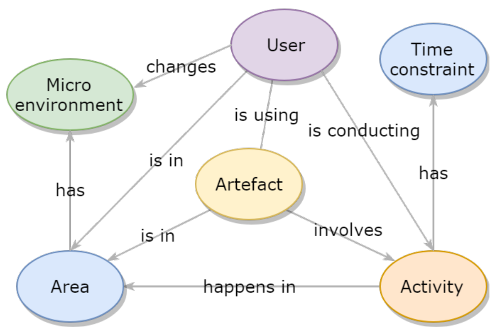


Figure 3.3: Activity area modeling

Elementary activities	Complex activities
Lying	Sleeping
Sitting	Watching TV
Standing	Preparing tea
Walking	Cooking
Running	Calling phone
	Eating
	Taking medicine

Table 3.1: Example of activities

3.3.1 Correlation Analysis of Devices

Intuitively, as a system to recommend reminders to user, we need to firstly understand what activity or activities our user is undertaking in relatively high level, that is, a Human Activity Recognition(HAR) task.

Studies on HAR has been studied extensively, yet the most ones are still simple or elementary activity recognition, especially with wearable sensors. And elementary activity recognition often only reveal limited information about the user, which can hardly meet the requirements of complex practical applications. Thus, there emerging studies on complex activities recently. And some [174, 165] have shown that decomposing complex activities into basic or elementary activities works well.

Here we also adapt the strategy that, exploits the recognized elementary activities as one source of our input, along with environmental sensory (see figure 3.1) data and the usages of home appliances, to reveal the correlations of devices.

Definitions

Our input sensory data is continuously streaming, with a certain sample rate limited by hardware. We denote the data reading d from sensor σ at timestamp t as a tuple $r =< t, \sigma, d >$. Accordingly we denote a recent set of reading of sensor σ in a duration between timestamp t_1 and t_2 as $R^{\sigma}_{t_1,t_2} = \{r_1, r_2, ..., r_n\}$, $n \in \mathbb{R}$, is the number of timestamps.

Given a recent set of sensory data readings $R^{\mathcal{S}}_{t_1,t_2}$ where $\mathcal{S} = \{\sigma_1, \sigma_2, ..., \sigma_m\}$ denotes a set of sensors in this reading, our task is now firstly recognize complex activities set $\mathcal{A} =< A_1, A_2, ..., A_k >$ happened within the time range (t_1, t_2), and then make recommendations of reminders $\Gamma =< \gamma_1, \gamma_2, ..., \gamma_l >$.

Elementary activity recognition

Here we assume to follow the set up of PAMAP2 [175], using 3 wearable inertial measurement units (IMUs) as data source. The IMUs in the setting can collect 3D-acceleration data, 3D-gyroscope data, and 3D-magnetometer data. Also as illustrated in left upper part of figure 3.2, the IMUs are deployed on the chest, wrist, and ankle of our subject. Here we considering first 5 status related elementary activities in PARMAP setup, namely: lying, sitting, standing, walking, running (see table 3.1).

Given a series of readings $R_{t_1,t_2}^{\mathcal{S}_{\mathcal{IMU}}} = \{r_1, r_2, ..., r_j, ..., r_n\}$, $r_j \in \mathbb{R}^K$, where K is the number of IMU sensory data we collect at one timestamp, in this step we require an series output of elementary activities $\alpha = \{a_1, a_2, ..., a_b s\}$ accordingly. As here the input is time series signals from multiple IMUs, DeepConvLSTM [176] classifiers have been proved powerful in handling such data. It consists of two units: Convolutional neural networks (CNNs) that has ability to extract features which represents the input for our system, and Long-short-term memory recurrent (LSTMs) that has a memory to detect the temporary dependencies in time series problem. Note that for such a classifier, the input is a batch of n, and here we set it to be the number of data we collect at one certain time duration with an overlap.

DeepConvLSTM is a state-of-the-art of deep neural networks which combines convolutional, recurrent and softmax layers. The ReLU Convolutioanl layers firstly take the size-n sequential batch inputs over time $[t_1, t_2)$, and they produce a 2D feature map of size $\eta \times K \times f$. Here f denotes the number of filters. The feature map is then received by 2 dense recurrent layers, which in here is made up with LSTM cells, and finally a softmax layer outputs the prediction results.

More details of our settings for this classifier can be found in the Experiment section later.

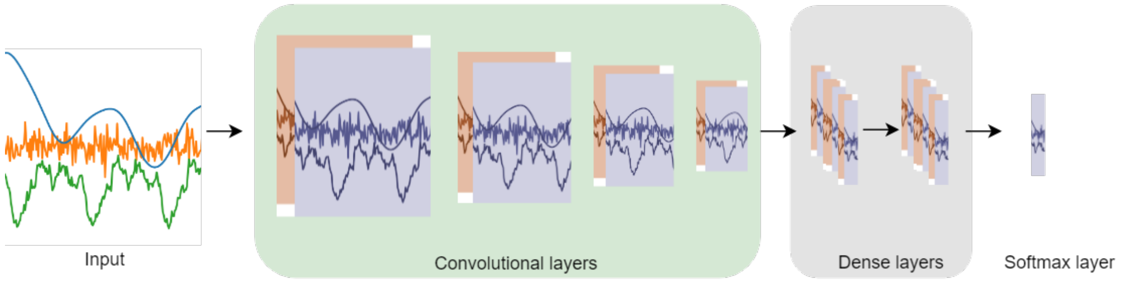


Figure 3.4: The structure of DeepConvLSTM

Ontology for complex activity recognition

To our knowledge, it is Yamada et al. [148] firstly come out the idea to apply ontology model in HAR, that involves interactions, i.e., subject performs activities that involves interactions with certain things. And such interactions often result in changing in status of things and environment, produce some contextual information. Let's consider a boiling water is the complex activity, the ontology for this activity as the following: 1) local environmental changes like motion (IR radiation by human) and locally high humidity in the kitchen, 2) triggering the usage status of the kettle, 3) time constraint as such activity is highly unlike to happen in early morning before sunrise. For this activity, we can observe firstly spatial contexts: motion detected in kitchen, that is the place this activity happens; secondly artifactual context: the kettle has been used; thirdly the environmental contexts: the risen in humidity.

Noted that in our settings there are some more context information that can help further, in subsection 3.3.1, we utilised wearable IMUs and DeepConvLSTM recognized elementary activities of our subject, and in this example, it outputs some

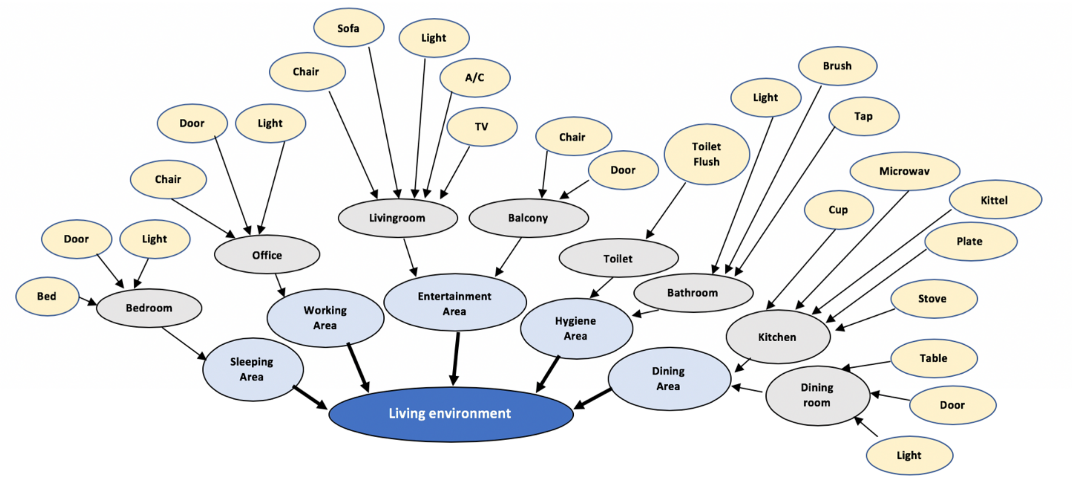


Figure 3.5: Example of living area ontology

activities include while not limited to standing and arm motions. This basic model of activity with area, environment and time constraint can be presented as figure 3.3, and figure 3.5 shows the relationships between artefacts and area.

Accordingly, we build a OWL (Ontology Web Language[1]) ontological models to help recognize complex activities, which include ontological models for *a*rtefacts, *e*nvironment, *l*ocations, and *a*ctivities. And figure 3.5 shows the ontology for location, which represents relationships between artefacts and area.

Noted here we have relatively manageable number of sensors deployed, which makes us possible to manually create instances. As for the collected data, we largely have sensors with binary outputs: PIR sensor indicates the absence or presence of human, i.e. user, in certain area with a 0 or 1 readings; electronic valves status indicates if water tap is running. Even sensors with complex value, we can still using simple rules to map the reading into standard semantic presentation. For example, UV meter with reading less than 3 stands for low, between 3 and 5 can be interpret

[1]https://www.w3.org/standards/techs/owl

moderate, etc.; Temperature/barometer sensors indicates temperature/pressure rise or fall.

3.3.2 Rule-Based Orchestration

This part is mainly use previous collected sensory data and ontological models, to recognize the complex activities the user is performing. It contains set of rules, which is extended from the ontological models and combined constraints. The rules response accordingly with set of inputs, and output recognized activities, following the knowledge carried by ontological models and semantically represented sensory data, along with elementary activities within the sensed duration.

Given our ontological models built, rules to recognize complex activities can be produced accordingly, which includes artefacts involved in this activity, area that the activity taken in place, key elementary activities that this activity contains, local environmental changes and the time constraint.

Following the example of activity *boiling water* above, we can set the ontological rule in descriptive language as:

$$\begin{aligned}
& boilingWater \\
& \sqsubseteq Cooking \\
& \quad \sqcap \exists\, involving \text{ Artefact.Kettle} \\
& \quad \sqcap \exists\, user_is_is \text{ Area.Kitchen} \\
& \quad \sqcap \exists\, user_is_conducting \text{ Activity.Standing} \\
& \quad \sqcap \exists\, risen \text{ Micro_environment.Humidity} \\
& \quad \sqcap \neg \exists\, has \text{ Time_constraint.EarlyMorning}
\end{aligned}$$

Similarly, by traversing the ontological models, we can build rules for all defined activities.

3.3.3 Activity-Triggered Reminder Recommendation

Early works for activity-triggered reminder recommendation usually using Fuzzy Logic to make decision. While studies show these approaches can work adequately, they can still be improved. One shortcoming is the logic is basically predefined and they may either unable to handle undefined scenarios or the system can be too complex to deal with unpredicted scenarios.

Collaborative Filtering technique, especially Matrix Factorization, has been widely applied in recommender systems, while limited works in reminder recommendation can be identified. Here we propose to use Non-negative Matrix Factorization(NMF)[177] as our predictor for reminder recommendation.

Activity A is already been recognized in previous steps, in addition to the descriptive ontology O for activity A with properties set P, and now for reminder γ, user have a rating ρ. After certain period of time, we can easily collect a set of activities records $\mathcal{A}$, related ontologies $\mathcal{O}$ with properties $\mathcal{P}$ and accordingly for a set of reminder Γ with ratings ϱ. For simplification, we note the matrix $[A; \Gamma; P; \varrho]$ as M, the NMF has goal to decompose M so that $M \approx VH$, which can be achieved by minimizing the error function:

$$min_{VH}||M - VH||_F, \text{ subject to } V, H \geq 0 \tag{3.1}$$

For $M' = VH$, we restore activities $\mathcal{A}$, reminders Γ and learned ratings ϱ', where the rank ρ's to $[A; \Gamma]$ is the list of our recommendations of reminders. According to the rounded ρ's, we issue the users with high priority reminders.

However, the system still struggles with cold-start. Common solution will be let the user give ratings for random reminders for a certain period of time, which is arguably undesirable. Here we exploit a previous study by Zhou et al. [10], where a survey is done and they have some ground truth on certain ratings under 20 different scenarios. This provision can help to boost our system in cold-start. We derived the data presented in their paper [2], and table 3.2 shows the data. Notice that the adapted technique for this stage only a proposed solution without evaluation. However, more details about the promising techniques and experiments are illustrated in chapter 4 and 5.

[2] Here we use “user expectation level of reminder (UEL)" as rating and 5 levels converted to numbers(5 stand for “very high" and respectively).

ID	Current activity	Triggered reminder	Rating
1	Sleeping	Taking medicine	4
2	Sleeping	Washing clothes	1
3	Sleeping	Turning off gas	5
4	Sleeping	Cooking	2
5	Answering phone	Taking medicine	2
6	Answering phone	Washing clothes	2
7	Answering phone	Turning off gas	5
8	Answering phone	Cooking	2
9	Watching TV	Taking medicine	4
10	Watching TV	Washing clothes	2
11	Watching TV	Turning off gas	5
12	Watching TV	Cooking	4
13	Wandering	Taking medicine	4
14	Wandering	Washing clothes	4
15	Wandering	Turning off gas	5
16	Wandering	Cooking	4

Table 3.2: Default reminders with rating (derived from Zhou et al.[10][3])

3.4 Evaluation

In this section we present systematic evaluation of our proposed reminder recommender system, via experiments on public datasets. As the proposed system consists of 3 major parts(see figure 3.2), the evaluation is conducted accordingly.

We will first introduce datasets used for evaluation, and then presents experiment settings and evaluation and result analysis.

3.4.1 Dataset

We evaluate the proposed system on well-benchmarked public dataset. Firstly we evaluated our elementary activity recognition module, since the recognized elementary activities are used as one of the inputs of our second process stage, it is essential to ensure its performance. This is evaluated with both PAMAP2 dataset [175] by UC Irvine and also the PUCK dataset [178, 179] in Washington State University(WSU) CASAS project in 2011.

After that, the system proceed to complex activity recognition, and this is mainly evaluated on the PUCK dataset, as it contains both smart home environmental sensors and wearable sensory data from smartphones, which match exactly our needs.

PAMAP2 dataset

The PAMAP2 dataset contains 9 participants performing 12 daily living activities including both basic actions and sportive exercises. The activity sensory data is collected from 3 Inertial Measurement Units (IMUs) attached to three different positions, namely the dominant wrist, the chest and the dominant side's ankle. Each IMU contains two 3-axis accelerometers, two 3-axis accelerometers, one 3-axis gyroscopes, one 3-axis magnetometers and one thermometer with sampling rate of 100 Hz.

Method	PAMAP2	PUCK
SVM	58.7	63.4
KNN	69.3	71.8
Random Forest	62.8	72.1
LSTM	71.9	**77.4**
DeepConvLSTM	**73.4**	77.2

Table 3.3: Baseline test of elementary activity recognition

PUCK dataset

The PUCK dataset[4] is collected in Kyoto smart home testbed in Washington State University, and it is a two-story apartment with one living room, one dining area, one kitchen, one bathroom and three bedrooms. Various environmental sensors are installed in this testbed, including motion sensors on ceilings, door sensors on room entrances, kitchen cabinet doors as well as microwave and refrigerator doors, temperature sensors in rooms, power meter, burner sensor, water usage sensors, telephone usage sensors and some item sensors for usage monitoring.

It is worth mentioning that there is also mobile wearable sensor data available for some subjects, in similar settings of PAMAP2, although less in the numbers of sensors. And in this work, the annotated ground truth is rather rare, we combined subject 3, 8 and 10, because they are the only source of data where environmental and wearable sensor both available publically. We identified 5 complex activities in this dataset, namely 16 instances of *Making phone call*, 4 instances of *Washing hands*, 4 instances of *Cooking*, 4 instances of *Eating* and 4 instances of *Cleaning kitchen.*

[4]http://casas.wsu.edu/datasets/puck.zip

Noted that with our ontology based rules, some more activities such as *Sleeping*, *Wandering* can be easily identified as well, yet not quite suitable for evaluation.

3.4.2 Elementary Activity Recognition

As mentioned, firstly we studied the effectiveness of DeepConvLSTM on our tasks, to identify the gesture of subject, that is the elementary activities. We tested several baselines, including SVM, kNN(k=3), Random Forest(128 trees) and ordinary LSTM with 3 layers and 64 cells. As for the DeepConvLSTM, we adapted a common configuration, where there are 4 convolutional layers with feature maps, 2 LSTM layer with 128 cells. All above tests are conducted with the same settings on both PAMAP2 and PUCK datasets. The above results (in table 3.3) prove that our choice of DeepConvLSTM, which outperform all tested baselines on PAMAP2 and adequate on PUCK, where the gap is so small that can be arguably ignored.

3.4.3 Complexity Activity Recognition

Given our ontology based rules ready, we can perform complex activities recognition. The tests result is reported in figure 3.6.

Here we can easily read that the overall accuracy for complexity activity recognition is of 65.625%, which is quite reasonable, especially considering that this is based on elementary activity recognition, where our accuracy is around 77.2%, as well as the sample is still quite rare. Also we can tell our framework tend to mis-classify *Phone call* and *Eating*, *Washing hands* and *Cleaning kitchen*. This may be caused by the sharing characteristics of those activities where similar places and devices are being

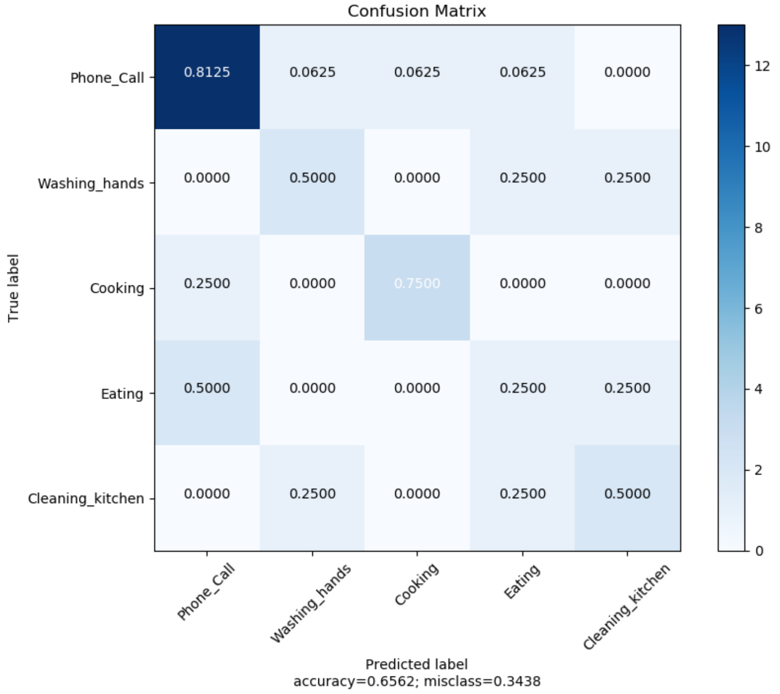


Figure 3.6: The confusion matrix of complex activity recognition

used.

3.5 Conclusion

In this work, we explored the feasibility of combining environmental sensors and wearable sensors, to recommend reminders to the people need extra care, especially Alzheimer's patient, in an activity-aware and cross-devices basis. We take advantage of one state-of-the-art HAR recognition approaches, DeepConvLSTM to identify the gestures of subject, combining which with predefined ontology representations the environmental sensory data, to produce rules for complexity activity recognition. And this output of subject behaviours are feed into a proposed recommendation engine, and output reminders to the user accordingly. The proposed recommeder system also will retrain itself given appropriate feedback from the user. Experiments demonstrated that our behaviour recognition part is effective and hence the recommendation engine practicable.

However, our evaluation of this system is still not comprehensive enough to prove the premium of this framework, mainly due to lack of test data sources to fit our complex experiments need. Also, in regarding to build an ontology; it could face three main issues that should be addressed:(1) requiring knowledge engineering skills to build the rules, (2) time consuming and difficult process particularly, when there are missing rules that are needed to take a decision, and (3) scarcity of available of standard datasets that are used to evaluate ontology models.

In the future, we will build a running system with a sensor enabled test-bed, to collect inclusive and adequate data, and test our framework in reallife use case.

Chapter 4

Deep Reinforcement Learning for Dynamic Things of Interest Recommendation in Intelligent Ambient Environment

This section contains works published in:

> [180] **M. Altulyan**, C. Huang, L. Yao, X. Wang, and S. Kanhere, "Deep Reinforcement Learning for Dynamic Things of Interest Recommendation in Intelligent Ambient Environment", in *Australasian Joint Conference on Artificial Intelligence*, pp., 2022.

4.1 Introduction

With the rapid growth in the number of things that can be connected to the internet, Recommendation Systems for the IoT (RSIoT) have become more significant in helping a variety of applications to meet user preferences, and such applications can be smart home, smart tourism, smart parking, m-health and so on. On the one hand, RSIoT can recommend an item that users might need in situations. On the other hand, it can save time and cost by actively allocating specific IoT resources accordingly to the very situations. Lets get back to Aris 77-year-old woman with dementia where RSIoT can play a caregiver role to help her in living independently and safely in her own home

Numerous efforts have been made to develop RSIoT using different approaches. Most of the existing works adapted conventional recommender system approaches, including collaborative filtering [181], content-based [16] and hybrid-based approach [182]. However, those conventional RSIoT approaches face two main issues. The first issue is treating the recommendation procedures as statics and ignoring the dynamicity in human activity patterns. More formally, human activity patterns could be changed at any time during the day or even after a period of time. The second issue is making recommendations for users while the system must wait for user feedback to update itself. While this may provide the system with accurate labels, it can have an impact on the end-user experience. RSIoTs should able to be updated based on the recommended item status only, which means no need to hold any device or to deal with any application.

Deep Reinforcement Learning (DRL) is inherently profitable for overcoming dynamic environments and thus has been adapted in interactive recommendation systems. It has been shown the ability to learn user decision behavior by observing the user's actions and conducting accurate recommendations even from a few samples by grouping its observations. Furthermore, it considers the feedback from the environment as a reward to update the system. Significant efforts have shown the notable performance of DRL methods in conventional recommendation systems [183, 184, 185]. Also, there are only very few studies on RSIoT systems based on RL [120, 119, 118, 186]. However, no previous research known to us has adapted DQN based RL for RSIoT. Inspired by [185], we design a Reminder Care System (RCS) based on DQN, which can tackle two main issues: dynamicity patterns of the human activity and the focus on the user feedback during system updates. We first formulate our system based on a Deep Q-Network (DQN), which captures the user's dynamicity pattern using three kinds of extracted features that address the first issue. Subsequently, we calculate the probabilities for items and nominate only one item with the highest probability as a recommendation. To tackle the second issue, we introduce our reward function that enables the system to receive feedback automatically without waiting for the user. Finally, we propose a new term called a Reward Delay Period which improves the evaluation for the quality of recommendations.

The main contributions of our proposed system are summarized as follows:

- We design the Reminder Care System (RCS) and formulate it based on the Deep Q-Network (DQN) which utilizes three main features: past activities features, current activities features, and item context features as an input (State).
- We formulate the reward function that helps the system to be updated automatically without needing feedback from the user by checking the status of

items after a period of time.

- We conducted extensive experiments on three public datasets, and our experimental results demonstrate the feasibility and effectiveness of our system.

4.2 Deep Reinforcement Learning in RSIoT

DRL has received significant attention in building recommender systems [187, 188, 184, 183, 189] for two main reasons; coping with dynamic environments by updating the strategies during the interactions and the ability to learn a policy that maximizes the long term reward.

Author in [184] proposed a deep recommender system framework (DEERS). It aims to exploit both negative and positive feedback to conduct recommendations in a sequential interaction environment. However, the negative feedback usually is much larger than positive ones which incorporate them simultaneously considers a main challenge. Consequently, authors model their framework by adapting a Markov Decision Process (MDP) and leverage Reinforcement Learning (RL) to learn the optimal strategy for recommendations.

For the medical field, RL was exploited to build a recommender system [190] which has ability to conduct medical recommendations for the hypertension patients. The evaluation results show the effectiveness of the proposed system in lowering the occurrence of hypertension. Authors in [191] adapted DRL for Home Energy Recommendation System (HERS) to manage the smart devices. It aims to minimize the electricity cost and to provide comfort residents. The system has ability to incorporate the direct human feedback to learn the user preferences and the user activities to learn the devices usage pattern. In [183], DRL was adapted for the page-wise recommendation to tackle two main issues: (1) using the real time feedback of the user to update the recommendations strategy, and (2) conducting accurate recommendations. Here, authors adapted the Actor-Critic framework which has ability to face the two main challenges of DRL: the dynamicity of space of action, and the computational cost of selecting the optimal action. DRL has also been utilized to propose a DEAR framework for online advertising recommendations [189]. It exploited the architecture of DQN to manage the internally actions, maximize the ads revenue and minimize the negative impact of ads. The extensive results show the effectiveness of the proposed framework. Also, authors in [185] proposed a novel DRL-based recommendation framework. It tackles two main issues: the dynamic nature of new features and users' preferences and the lack of information to improve the quality of recommendations. Here, authors exploited: (1) user activeness as source, and (2) a new exploration method (Dueling Bandit Gradient Descent) to improve the quality of recommendations. The experiments results show the effectiveness of the the proposed framework in term of the recommendations accuracy comparing with other methods that ignored the user activeness or used traditional exploration methods. This work was inspiration for this chapter.

4.3 Background

Deep Q Network was proposed by DeepMind in 2015 to deal with wide range of Atari games. The idea of the algorithm is to enhance the performance of classic RL by combining the RL with deep neural networks. Here, the deep neural network maps state and action pairs to Q-value using. DQN aims to maximize a cumulative future reward by recommending the correct item a_i for each activity that needs a recommendation.

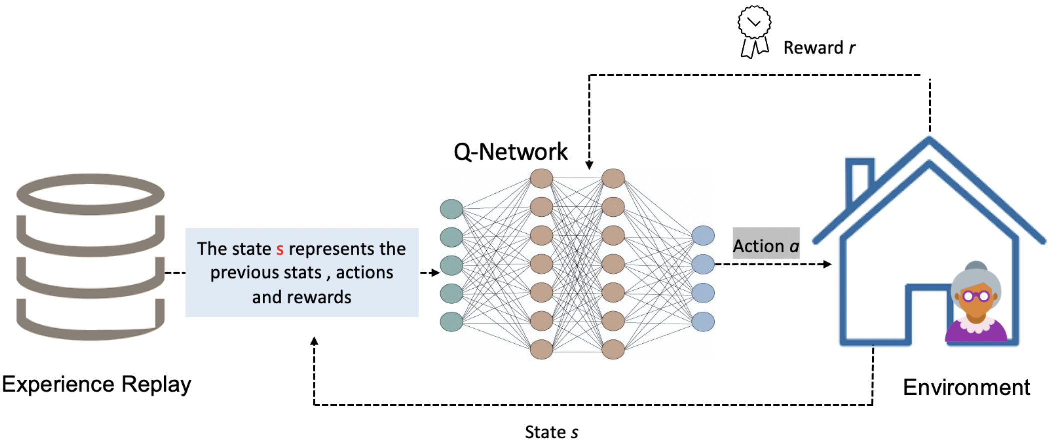


Figure 4.1: The framework of Deep Q-Network (DQN).

Figure 4.1 shows the DQN framework. It consists of three main components for the DQN: (1) Q-Network which could be a standard neural network or regular network depending on the state and Q-Target is identical to the Q Network that is held for stabilize learning ; (2) Environment which represents the state of the user and (3) Experience Replay that stores all the interactions with the environment and uses them as mini-batch to update the network. Here, in each time slot the current state s represents all previous actions and states, when the state s passes to the network as input into the Q network, the Q-Network selects the action a as output. After the action a is executed, the system receives the next state $s^{'}$ and a reward r. Finally, the experience reply buffer stores a record of $< s, s^{'}, a, r >$ and repeat over it again. When the experience reply buffer reaches the maximum of the capacity, random samples will be picked randomly to calculate the current target Q value using the following equation:

$$y = r + \gamma max_{a'} Q(S_{t+1}, a^{'}; \Theta_i^-) \tag{4.1}$$

where y represents the output of the neural network, S_{t+1} represents the new state , t represents the time slot t, γ is a discount factor , θ_i and θ_i^- represent the weight of the Q-network and the target-network at iteration i respectively.

The DQN has two main features compared with other RL algorithms: (1) using the experience replay buffer to store the agent experiences $E = \left\{s_i, a_i, r_i, s_i^{'}\right\}$ which represents state, action, reward, and next state respectively, (2) adjusting any update for the target network.

4.4 Reminder Care System (RCS) framework

In this section, we introduce Reminder Care System (RCS) in detail. First, we define the problem and notations; then, we provide an overview of our framework. Finally, we describe DQN and explain the process of our agent.

4.4.1 Notation

Our problem is framed as follows: when extracted features of the complex activity v where $v_i \in V = \{v_1, v_2, ...v_m\}$ is received by the agent G (The extracted features will be explained in details in section 4.4.2). Notice that the agent receives the extracted features of the activity that needs recommendation only as an input (state) s. Then the agent nominates an appropriate item a from a fixed candidate set of items A for the particular activity. In other words, the algorithm generates ranking list $\Gamma =< \gamma_{a_1}, \gamma_{a_2}, ..., \gamma_{a_l} >$, where γ_{a_i} denotes the probability of the item a_i where the user needs to finish the current activity. Unlike a conventional recommendation system that typically recommends more than one item for users each time, our agent

Notation	Explanation
G	Agent
v, V	Activity, set of activities
s, s'	state, next state
a	action(item)
r	Reward
A	set of items
Q	Q-Network
W	Parameters of DQN parameter
Γ	Ranking list
γ_a	Probability of the item
O	Value of each item
E	Experience replay buffer

Table 4.1: System Notations

recommends only one item with maximum probability for the activity that needs a recommendation. Table 4.1 summarises the notations used throughout this chapter.

4.4.2 Overview

In this section, we describe our framework as shown in figure 4.2. We divided it into two main parts online and offline. In the offline part, our system will be training to deal with the activities that need a recommendation. Notice during the training, we treat each activity that needs a recommendation as a session. During the online part, the agent receives the required features as an input (state) s; then recommends an appropriate item a_i for the activity. There are three kinds of features that should

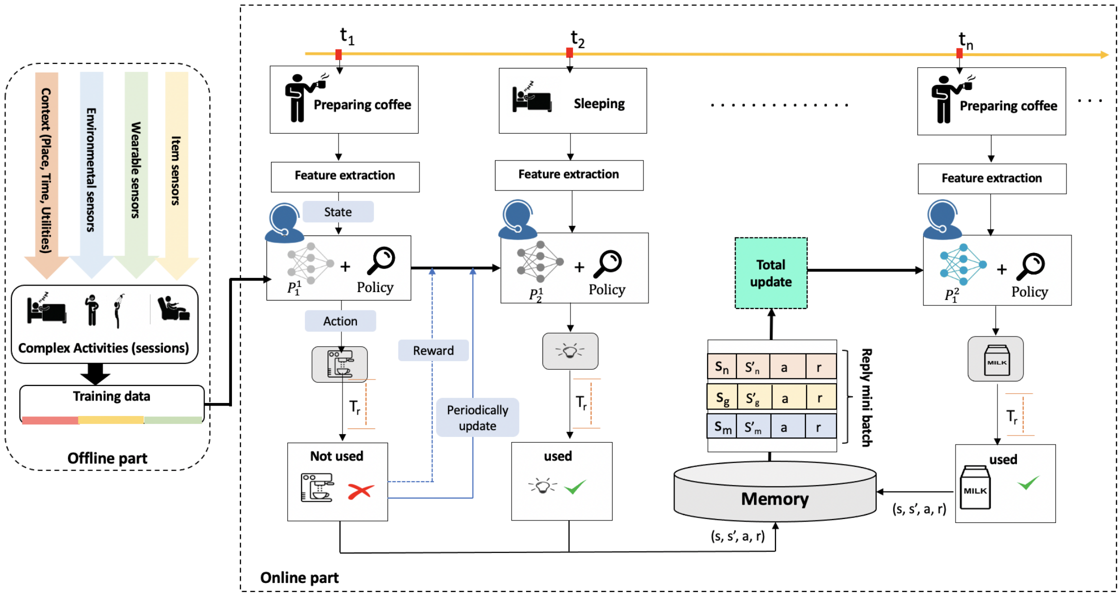


Figure 4.2: The architecture of our framework, which consists of two main parts: offline and online. The offline part focuses on training our agent G using different datasets. During the online part, the agent receives each activity and extracts the required features as state s, then nominates suitable item a_i for this state. After a periodic time called reward delay period T_r, the agent will receive a reward r. Two kinds of updating will be applied for the system: periodically update after every recommendation P and the total update after a period of time using the experience replay buffer.

be received for each activity (session) that needs a recommendation:

- Past activities features. Since each activity can have a different pattern, for each activity, the system extracts the path/sequences of items used. They enable agent G to learn different patterns of each activity.
- Current activities features. It can define where the user is stuck by reviewing all previously used items for this activity. This feature helps the agent to

ignore all the used items before and to choose from the rest.

- Item context (IC). It includes information about items, such as to which activity this item belongs to, how long it could be in use, and how many times the user needs it for the current activity.

To improve the system's recommendation accuracy, we consider all features that help our system learn the best action in a specific state. The public datasets did not meet all required features , which affect the results of our systems. After the features are extracted, we apply the DQN algorithm to model our agent.

4.4.3 Deep Q network (DQN)for recommendation

Here, agent G (Q-network) will be trained using the offline part. During this part, the agent learns to map each stat for suitable action. Then, the agent calculates the reward as feedback to updated the system. We summary the agent roles as following:

1. Receiving the extracted features of current state s during the interaction with the environment at timestamp t.

2. Generating a list of recommendations Γ that includes top items to be recommended using exploration and exploitation policy. Notice, the agent will pick only one item with the highest ranking.

3. Calculating the reward which considers as feedback to update the system using the following equation:

$$r(a) = \frac{\sum_{t=0}^{T_r} O_{a,t}}{\sum_{a=0}^{A} \sum_{t=0}^{T_r} O_{a,t}} \tag{4.2}$$

where O represents the value of each item at each time step and T_r is a Reward Delay Period.

However, the agent G has to wait for the reward delay period T_r (will be discussed in the next part).

4. Periodically updating after every recommendation by comparing the performance of the Q-network with target network using the following loss fucntion:

$$loss = MES(predicted\ Q - Value, Target\ Q - Value) \tag{4.3}$$

5. Total updating for the system to tackle the dynamicity of human activity pattern, the agent G after a period of time (it is defined to be 24 hours for our system) will use the experience replay buffer to update the network Q using the following loss equation [192]:

$$L_i(\theta_i) = E_{(s,a,r,s')} \sim U(D) \left[\left(r + \gamma max_{a'} Q(s^{'}, a^{'}; \theta_i^{-}) - Q(s, a; \theta_i) \right)^2 \right] \tag{4.4}$$

Most traditional recommender systems focus on "click" or "not click" as feedback to calculate the reward function immediately and to update the system. In contrast, our system recommends an item to the user and then waits for sufficient time to decide if the recommended item is used or not—by checking its status (on/off or moved/not moved). For example, if the system recommends a coffee machine to Aris when she is preparing a cup of coffee, whereas she wants to use it later yet not immediately. This does not mean that the recommended item is incorrect, and it is better for the system to ignore this false negative feedback this time. To facilitate the above, we introduce a Reward Delay Period T_r, which accounts for the different paces of users in carrying out activities, and we consider T_r a hyperparameter.

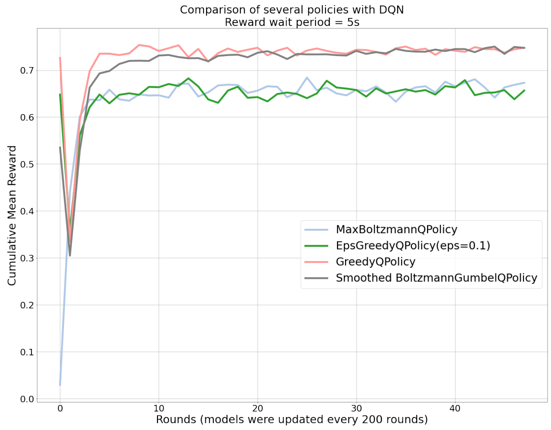
Policies	
(1) EpsGreedyQPolicy:	it combines both exploration by taking a random action with probability epsilon and exploitation by taking the current best action with probability (1 - epsilon)
(2) GreedyQPolicy:	it focuses on exploration where it calculates the probability of choosing the action with the highest Q-value
(3) BoltzmannGumbelQPolicy:	it is an exploration rule which defines probabilities of actions based on their Q-values.
(4) MaxBoltzmannQPolicy:	it adapts the Gumbel-softmax trick to address the classic Boltzmann exploration issues
Hyperparameters	
batch_size	200
Epsilon	0.1
target_model_update	1e-3
nb_steps	50000
verbose	1

Table 4.2: Explaining policies and tuning hyperparameters for our agent.

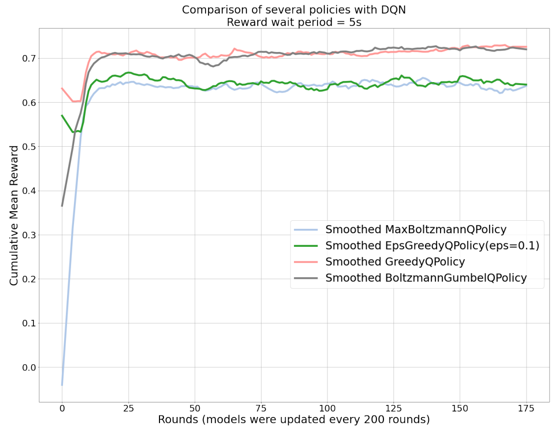
Here, our agent acts using different policies: EpsGreedyQPolicy, GreedyQPolicy, BoltzmannGumbelQPolicy, MaxBoltzmannQPolicy. Table 4.2 shows details about the policies with DQN that we used for our agent and their parameters.

4.5 Experiment

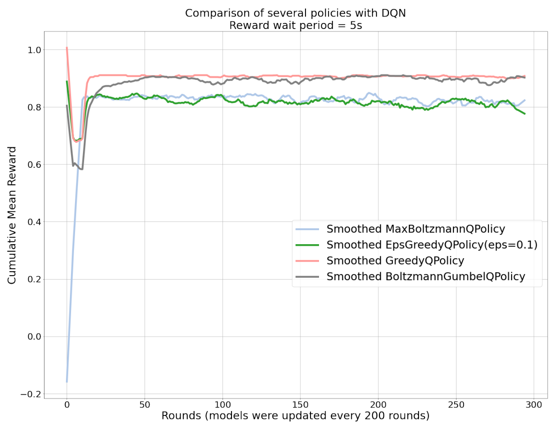
In this section, we first introduce three public datasets that we used for our experiments. Next, we conduct some experiments that show the effectiveness of our proposed RCS and evaluate the performance among these datasets.



(a) PUCK.



(b) ARAS(houseA).



(c) ARAS(houseB).

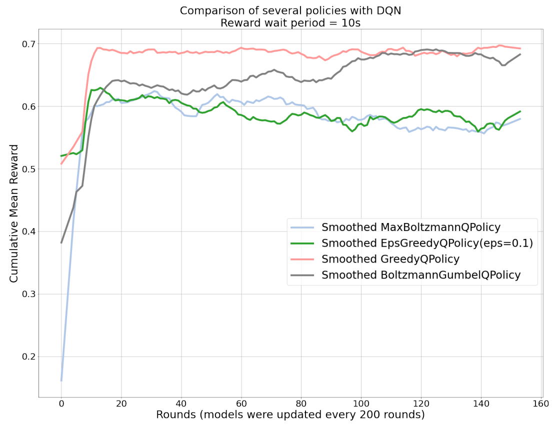
(d) ADL.

Figure 4.3: Performance of our system among three datasets when the Reward Delay Periods T_r is 5s.

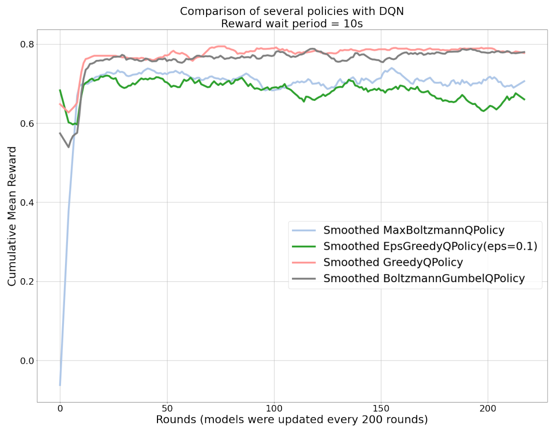
4.5.1 Datasets

Our evaluation focused on the offline part, and we left the evaluation of the online part to future work. The evaluation has been applied on three public datasets: PUCK [178], ARAS [193], and ADL [194].

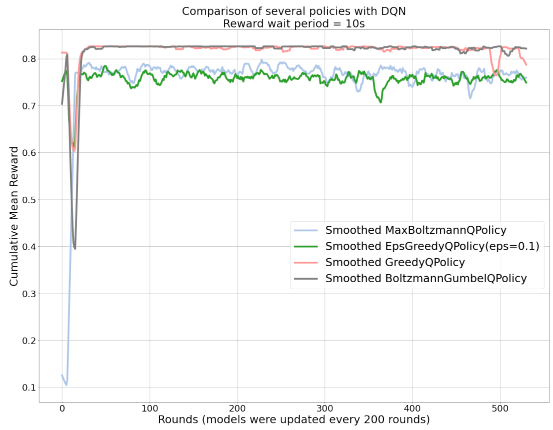
(a) PUCK.



(b) ARAS(houseA).



(c) ARAS(houseB).

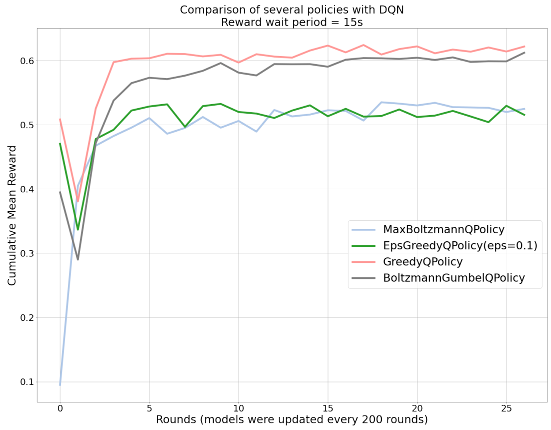


(d) ADL.

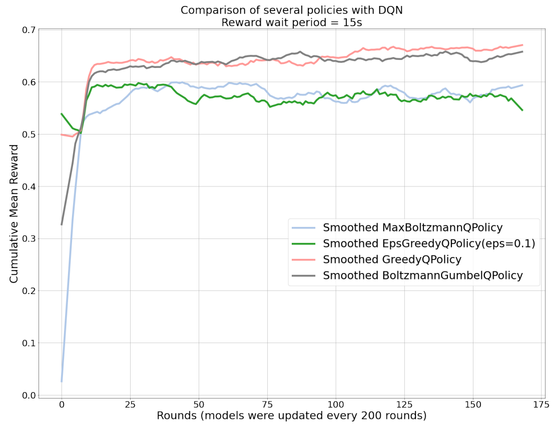
Figure 4.4: Performance of our system among three datasets when the Reward Delay Periods T_r is 10s.

The PUCK dataset[1] was collected in Kyoto smart home testbed in Washington State University, and it consists of a two-story apartment with one living room, one dining area, one kitchen, one bathroom, and three bedrooms. A number of environmental sensors have been installed in this testbed. The ARAS dataset contains two houses of two residents who performed 27 daily living activities. The activity-sensory data was collected from 20 binary sensors. The ADL Normal dataset represents a public

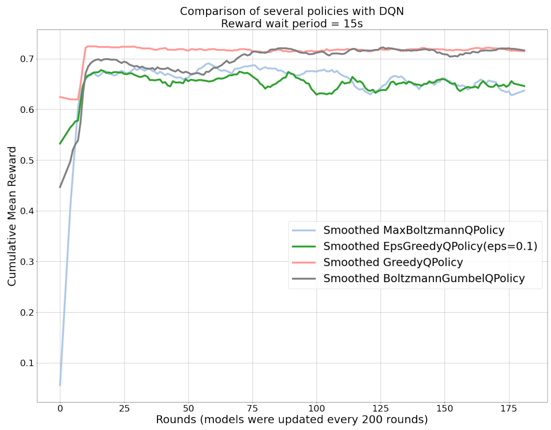
[1] http://casas.wsu.edu/datasets/puck.zip



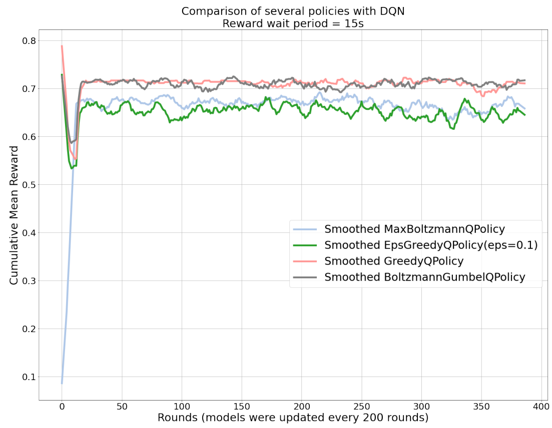
(a) PUCK.



(b) ARAS(houseA).



(c) ARAS(houseB).



(d) ADL.

Figure 4.5: Performance of our system among three datasets when the Reward Delay Periods T_r is 15s.

dataset published in 2010. The dataset was collected from a Kyoto smart apartment testbed in Washington State University. It includes five complex activities. The activities are performed by 20 participants. Features engineering and data processed are explained in detail in our previous work [195].

4.5.2 Experiments results

We first evaluate the effectiveness of our RCS in recommending the correct item to the user in case the user's current activity needs a recommendation. The agent uses all the extracted features as the state to make a recommendation of the correct item. We use the DQN model that is provided by one public available 'keras-rl' package of python for our experiments. The hyper-parameters of the DQN model are configured as follows: the number of layers in DQN is set to 6 with two Flatten layers, three Dense layers, and one Activation layer. The package provides a number of policies that help our agent to map each state with a correct action.

Dataset	T_{r}	Policies			
		EpsGreedyQPolicy	GreedyQPolicy	BoltzmannGumbleQPolicy	MaxBoltzmannQPolicy
PUCK	5s	0.65	**0.75**	**0.75**	0.67
	10s	0.58	**0.68**	0.66	0.60
	15s	0.51	**0.64**	0.62	0.52
ARAS House (A)	5s	0.64	**0.72**	0.71	0.63
	10s	0.59	**0.69**	0.68	0.57
	15s	0.54	**0.67**	0.66	0.59
ARAS House (B)	5s	0.77	**0.90**	**0.90**	0.83
	10s	0.68	**0.78**	**0.78**	0.70
	15s	0.65	**0.73**	**0.73**	0.64
ADL	5s	0.70	**0.77**	**0.77**	0.72
	10s	0.74	0.77	**0.82**	0.75
	15s	0.64	0.71	**0.72**	0.65

Table 4.3: The cumulative mean reward of our system among three datasets using different reward delay periods.

The performance of our system is shown in figure 4.3. As we can see, the cumulative mean reward for the three datasets; However, the two policies: GreedyQpolicy and BlotzmanngumbleQpolicy, produce the highest performance compared with another two policies. Also, the ARAS dataset for house B (see figure 4.3c) has the highest cumulative mean reward compared with other datasets. There are different reasons that could affect the results on different datasets: (1) the number of item sensors used to collect the data, (2) the time period between each reading of the sensors values, i.e., half a second for PUCK and a second for ARAS and ADL datasets, (3) the type of sensor value such as binary or continues values, (4) the number of items that are included in each activity. Moreover, all the previous datasets do not consider time as a context which is an important feature for our system. As mentioned, the Reward Delay Period parameter T_r has direct impacts on the model's performance which controls when the agent should receive the feedback as a reward. Adjusting this parameter is important, and it is various from one activity to another depending on how much time each item consumes to be used. For example, some items take a little bit of time to be picked, and others may be a little bit longer. We assume three values of T_r: 5s, 10s, and 15s, then we monitor the performance among these different values. Figure 4.3, figure 4.4 and, figure 4.5 show the performance of our RCS on the three datasets using three different values of T_r: 5s, 10s, and 15s respectively. Table 4.3 summarizes all cumulative mean rewards of our system among three datasets using different reward delay periods. We can observe that our proposed system performance when the T_r=5 consistently outperforms the other two values: T_r=10, and T_r=15 for the two datasets: PUCK, and ARAS. However, the ADL dataset demonstrates that increasing the T_r to 10s improves the cumulative reward to be around 0.82 instead of 0.77 and 0.72 for the T_r=5 and T_r=15 respectively. Moreover, table 4.3 shows the effectiveness of The GreedyQPolicy and BoltzmannGumbleQPolicy policies compared with the other two policies. However, among all the policies, the GreedyQPolicy performs well and is stable except in the last dataset with T_r=10.

4.6 Conclusion

In this study, we designed a Reminder Care System (RCS) that uses deep reinforcement learning to capture dynamic patterns of human activities and update the system automatically without waiting for user feedback. The RCS uses DQN to formulate the agent and considers the reward delay period to account for the different paces of users in carrying out activities. We conducted experiments on three real-world public datasets to show that the effectiveness of our system. For future work, we will test our system on a real-time testbed that considers all features requirements for the proposed system.

Chapter 5

Context-induced Activity Monitoring for On-demand Things-of-Interest Recommendation in an Ambient Intelligent Environment

This section contains works published in:

[196] **M. Altulyan**, C. Huang, L. Yao, X. Wang,and S. Kanhere, "Contextual bandit learning for activity-aware things-of-interest recommendation in an assisted living environment",in *Australasian Database Conference*, pp. 37—49, Springer, 2021.

[197] **M. Altulyan**, L. Yao, C. Huang, X. Wang,and S. Kanhere, "Context-induced Activity Monitoring for On-demand Things-of-Interest Recommendation in an Ambient IntelligentEnvironment", *Future Internet*, vol. 13, no. 12, pp. 305, 2021.

5.1 Introduction

The widened applications of IoT-based smart-home environments birthed the idea of a recommender system, reminder care systems, which are adapted to improve the management of patients with AD. A reminder care system considers patients who suffer from the mild stage of AD where patients just start losing short-term memory however, they still have the ability to use such a system [198, 199]. A reminder care system is designed to exploit sensory data from various sources such as the environmental sensors, wearable sensors, and appliance sensors for effective reminder recommendations. A feedback system is not necessarily required to improve the quality of recommendations. The scenario presented in figure 1.1 reveals the importance of a reminder care system for Alzheimer patients. For example, the system is designed to promptly recommend switching off of appliances if she forgets to do so after usage. The system is subsequently improved based on her acceptance of every recommendation without needing Aris' explicit feedback.

Here, the system considers all contexts about the user and items, for example, if Aris starts to prepare a cup of coffee at midnight, no item will be recommended by the system but instead it will remind her to go back to sleep because time is considered as a key context in conducting the recommendations. Moreover, the system has the ability to learn all new patterns and ignore other old patterns. For illustration, when Aris no longer adds milk to her coffee, the system will not recommend milk during this activity. This scenario could be extended to become not only smart home application but also for m-health application. Again, Aris at home can be better monitored by the hospital by utilizing the data from sensors that are installed in her house to remotely monitor the development of her condition. Also, the hospital can provide medical advice recommendations such as recommending a specific time for resting, recommending some exercise to be done, or recommending some kind of foods. Notice, our proposed system is suitable only for the first stage of Alzheimer because in other stages patients face more changes in their behaviour where they need an intensive care. Number of studies have focused on reminder recommender systems aimed at providing assistance to elderly people who suffer from AD. Oyeleke et al. [120] designed a recommendations system for monitoring the daily indoor activities of seniors with mild cognitive impairment while Ahmed et al. [200] proposed a smart biomedical assisted system to help patients with Alzheimer. In some studies, smartphone applications were developed for the provision of care services to AD patients [201, 202, 203, 204].

However, the dynamicity in the complexities of human activities is yet to be adequately addressed, thus delivering low-quality recommendations. Another notable issue is the increased focus on monitoring which gives a reminder to patients while the system has to wait for patients' feedback to update itself. From Aris scenario, consider Aris follows the following sequence of actions when starting to prepare a cup of coffee in the early morning: first, switching the coffee machine on then bringing a cup, next, filling milk, and then adding sugar. Supposing then she grabbed a cup and forgets what to do next? the system should remind her of grabbing the milk. Nonetheless, if one day, she changes this pattern by deciding not to add milk in the future, the system should also cope with that. From Aris' perspective, the system should be a caregiver i.e. to offer help only when necessary without actively requesting feedback. Therefore, a well-designed reminder recommender the system must be capable of assessing the quality of recommendations without necessarily relying on user feedbacks.

In one of our previous works [29], we implemented a prototype system with great consideration of the dynamicity of human activities which was capable of detecting the complex activities. Then, in [180], we presented a Reminder Care System (RCS) which in addition to being able to learn the dynamicity of human activities, it could also remind patients about their needs correctly, without requiring their feedback. The problem was formulated using Deep Q-Network (DQN), which works well with the dynamic nature of human activities. However, most of RL algorithms is their inability to handle a system requiring learning and selection of the best action from different scenarios where each state is treated independently. As we mentioned in Aris scenario, the system works as caregiver which means acting based on the patient behaviour by paying attention to her needs. Consequently, the RSIoT system needs to consider all context from both user and item at the same time to decide which item should be conducted at this moment unlike previous approaches. In this work, a contextual bandit approach was adapted to formulate the same problem. The main contributions of this chapter are as follows

- Proposition of a recommender system based on contextual bandit approach by fusing the context information from the past and current activities to recommend the correct item.
- Provision of a minor and major updates to help tackle the dynamicity in human activities while improving the quality of recommendations.
- Evaluation of the developed model using three public datasets.

5.2 Contextual bandit (CB) for recommendations

Contextual bandit (CB) approaches can exploit both offline data and environment interactions that help in constructing recommendations with high quality. In CB, there are three main concepts in CB: State which defines which activity is performed by the user, Action that represents the item that the user need for this state and the Reward that the system receives based on the quality of the recommended item. Contextual bandit utilized the common features of RL by using policy to decide an action based on the context of each state this is similar to multi-armed bandit (MAB) which focuses on the immediate reward. As shown in figure 5.1, both action and state affect the reward which have a positive impact to increase the quality of the recommendation. In contrast, the action in RL effects not only the reward but also the action which means RL cannot deal with different scenarios as we motioned before [205]. Some studies have adapted CB for their recommender systems.

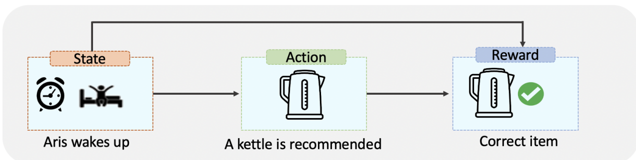


Figure 5.1: The three main concepts in the Contextual Bandit Approach:(1) State that represents the current situation for the user, (2) Action that provides the required item that meets the user need , and(3) Reward that considers as feedback for the system to improve the quality of recommendations .

Li et al. [152] utilized the contextual bandit for the recommendations of news articles. The presented algorithm, LinUCB, was reportedly applied to sparse and large data combined with other algorithms such as e-greedy. The study presented in [206], an online learning recommender system was developed by adapting CB where information for history students were learnt, and current students are used as context to conduct the learning recommendations to the students. Author in [205] adapted CB to decide on action to be carried out by a robot deployed to help dementia patients with their behavioural disturbances. Zhang et al. [207] proposed a novel CB method named SAOR for online recommendations. The study offers sparse interactions which distinguishes between negative response and non-response to improve recommendation quality. In this study, CB is adapted by utilizing three kinds of features, context past activities, current activity, and items. Also, a reward model for automatic update of the system requiring no feedback is also presented in this work. The proposed system has three major stages as shown in figure 5.2 and these are: complex activity recognition stage, prompt detection stage, and the recommendation stage. The complex activity recognition stage is hinged on three data sources, data from the wearable sensors, environmental sensory data, and home appliance usage data. The prompt detection stage utilizes data mining approach to determine if an ongoing activity requires an item recommendation while at reminder recommendation stage, CB approach is applied to extract context from the two previous stages to recommend items to the user during an activity. The stage one was presented in chapter 3. The stage two and stage three will be presented in the next sections below.

5.3 Prompt detection stage

At this stage, the data collected from the previous stage is used to determine the prompt of an activity. The prompt is considered in two main situations: when the user appears to be stuck within an activity for notable period without taking an action, and when the user uses a wrong item that does not belong to this activity.

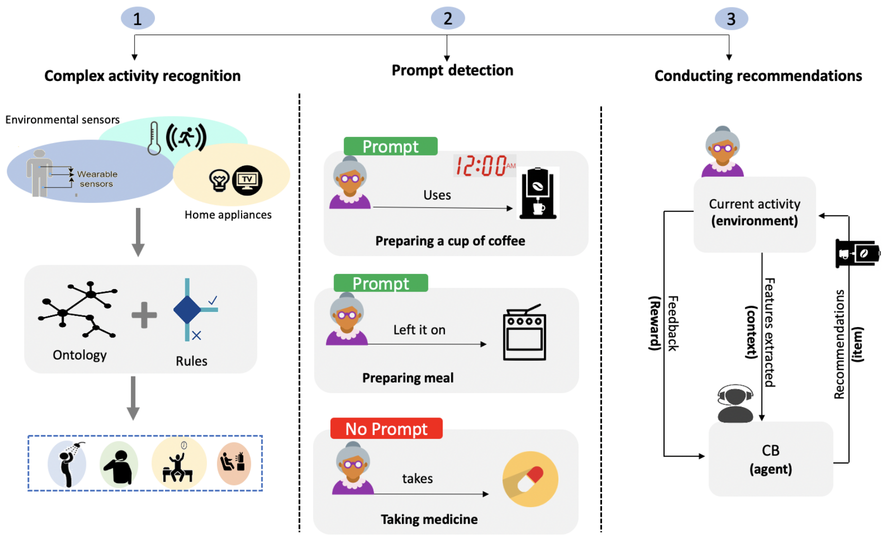


Figure 5.2: Overview of the proposed methodology.

In the previous stage, the developed complex activity detection module is capable of complex activity detection and learning of different activity patterns. The extracted features of each activity is used to build prompt detection system. As mentioned earlier, Alzheimer patients in the mild stage may not be able to complete their activities due to forgetfulness. For example, if Aris forgets to turn off the stove after making her tea, the system can detect that the user needs a prompt and immediate recommendation to turn off the stove. Various learning models are applied in the determination of when a user needs a prompt during a monitored activity. Das et al. [178] test several classification algorithms (Support Vector Machines (SVM) [208], Decision Tree [209] and Boosting [210]) on the PUCK dataset. In particular, Boosting applies a classification algorithm to re-weight the training data versions sequentially and then extracted a weighted majority vote of the previous sequentially classifiers. And it generally outperforms the other two methods. For our experiments, only one dataset provides the labels where the user needs prompt or not by adding class from 0 and 1. However, for the other two datasets we create the points that define when the user need prompt as we will be explained in details in section 5.5.

Notation	Explanation
G	Agent
x, X	Context, set of context
a	Action(item)
r	Reward
A	Set of items
M	Memory
S	State
Miu	Minor update
Mju	Major update
T_r	Reward Delay Period
$\prod$	Policies
SV	State value of each sensor

Table 5.1: System Notations

5.4 Conducting recommendations stage

Having determined that a user's activity requires a prompt, the system at this stage then decides which item can be suitably recommended based on the user situation. One of the main challenges, is handling each activity differently. The system must always consider what correct item is to be recommended even if it is the same activity by considering the user situation. For this reason, each activity is treated differently as a session during the training where it helps the system to learn different pattern of each activity. This stage represents our main contributions in this chapter.

5.4.1 Problem definition

When a complex activity that needs a prompt v where $v_i \in V = \{v_1, v_2, ...v_m\}$ is received by the agent G at time t, our algorithm extracts the context x and nominates an appropriate item a from a set of actions A for the current activity. Notice that our system recommends only one item at each time that needs a prompt. Then, the agent receives feedback as reward r for the recommended item. Finally, the system is being updated based on the received reward which is called minor updated Miu and the major update is after a certain time of period Mju. Table 5.1 summarises the notations used throughout this chapter.

5.4.2 Method

The problem is formulated based on a contextual bandit approach to tackle the dynamicity of human activity patterns and to recommend the correct item without having to wait for the user's feedback. Contextual bandit provides a learning model based on context. Three kinds of context are extracted at this stage:

- **Past activities context (PAC)**: Note that each activity is desired to have a different pattern, so, for each activity, the system extracts the path/sequences of items used from the past records (recorded in the log file) as a type of context. The observed paths of each activity are then stored in a memory based on which the agent can decide an item to be recommended at a specific situation.
- **Current activity Context (CAC)**: The contexts on the current states are extracted from the received data obtained from the previous two stages. For

ALGORITHM 1: Our procedure to recommended a correct item for user's activity. It takes context x as input, and returns a recommended item as output a

Initialize the capacity of storage memory M
Initialize a timer= $24hours$
for session$v_t \in V$ **do**
  Observe state s_t
  Extract x_t, where $x_t = PAC, CAC, IC$
  Execute action following set of policies $\prod$
  waiting for T_r
  Compute $r(a)$ according Equation (5.1)
  Minimize R according Equation (5.2)
  update Miu
  Put x_t, r_t, a into M
  Update Mju
end for
return

example, when the system receives that the user needs a prompt for preparing coffee, the context of the current activity (locations, previous items, user position, and time.) will be extracted.

- **Item context (IC)**: This basically concerns information about items, such as determining which activity an item belongs, how long such item can be in use, and how many times such items is needed by the user for the current activity. For example, a coffee machine as an item can be used for the activity of 'preparing coffee', where it can be used for around 2 minutes each time.

The contextual features of each session of the activities is received by the agent as input (Algorithm 1). The CB combined three main components: an environment, which represents the context of the user's activity $x \in X$, an agent G, which chooses an action $a \in A$ which is represented an item in our system (notice that the common name in CB is Learner but we call it an agent in our case) based on the received context, and a reward $r \in [0, 1]$, which the agent aims to maximize by recommending the correct action at each round t= 1, 2, ...,T. We calculate the expected reward after each recommendations as fallowing:

$$r(a) = \frac{\sum_{t=0}^{T_r} SV_{a,t}}{\sum_{a=0}^{A} \sum_{t=0}^{T_r} SV_{a,t}} \tag{5.1}$$

Notice that based on the reward equation, the system will receive a reward when the agent recommends an item only. So, after the reward calculation the system can define the if it is a negative reward or a positive reward. If the agent does not recommend any item that means there is no reward, and the system can be in an idle mood.

Where T_r is the Reward Delay Period(It is explained below) and SV depicts the state value of each sensor at each time step t. The reward function for automatically obtaining the feedback without waiting for the user is formulated. As shown in figure 5.3, the calculated reward gives a non-correct recommended item (the coffee machine). Consequently, the system considers it for updating. Two types of updates are carried out to keep the system in constant interactive recommendation, minor update and major update. In minor update Miu, the system is updated after the agent receives the reward where the system compares the recommended item with the item already used by the user instead of it. Such item is then updated with the newly used item to keep tracking the remaining items of the activity. Major updates are considered after certain period before the end of the day where the system is updated using the memory of historical data as obtained from the agent in the last

24 hours. The major update Mju helps the system to tackle any dynamicity of the user pattern during the day as we mentioned before in Aris scenario that her pattern could be changed even if she has still doing the same daily activities.

Most traditional recommender systems focus on 'click' or 'not click' as feedback to promptly evaluate the reward function and to update the system. In contrast, our system having recommended an item, waits for sufficient time to determine if the recommended item is used or not by checking its status (on/off or moved/not moved). This status is then used to update the system accordingly. Furthermore, if the system recommends a coffee machine to Aris (see figure 5.3)) when she is preparing a cup of coffee, whereas she wants to use it later and not on the immediate. The recommended item may not be seen as incorrect due to the false negative feedback at this time. The recommendation though not needed at this time can be used at another time. To facilitate the above, we introduce a Reward Delay Period T_r, which accounts for the different paces of users in carrying out activities and we consider T_r a hyperparameter (to be detailed section 5.6).

The agent G can choose from a set of policies $\prod \subseteq \{x \to A\}$ to map each context for a suitable item by employing two streaming models: Linear regression and stochastic gradient distance (to be detailed in section 5.6). The goal of using different polices is to minimize the regret R between the expected reward of the best action a^* and the expected reward of selected action a. The regret is calculated by using the following equation:

$$R_t(T) = E\left[\sum_{t=1}^{T} r_t, a_t^*\right] - E\left[\sum_{t=1}^{T} r_t, a_t\right] \tag{5.2}$$

Here, we adapt three categories of policies as the following:

- Randomized (AdaptiveGreedy) which focuses on taking the action that has the highest reward.

- Active choices (AdaptiveGreedy) which is the same for AdaptiveGreedy but with active parameter ! = None which means actions will not be taken randomly.

- Upper confidence bound (LinUCB) that stores a square matrix which has dimension equal to total numbers of features for the fitted model. Details about parameters for each policy of two streaming models: Linear regression and stochastic gradient distance will be detailed in section 5.6.

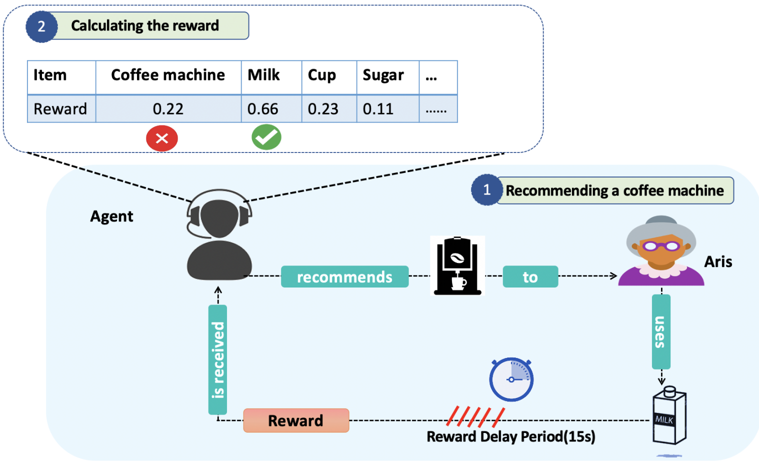


Figure 5.3: (1) The agent recommends a coffee machine to Aris whereas she uses milk instead; then the system waits for 15s; (2) The feedback is received by the system as a reward and it is calculated accordingly as the coffee machine is the wrong item; and (3) two kinds of updaing for the system: minor update is after receiving the reward and the major update after certain period around 24 hours.

5.5 Dataset

We have to mention that our evaluation focused on the third stage of our system which is conduction recommendation. This is evaluated on three public datastets: PUCK [178], ARAS [193], and ADL [194].

5.5.1 PUCK dataset

The PUCK dataset is a public dataset published in 2011. The PUCK dataset collected from a Kyoto smart home testbed located in Washington State University in two-story apartments with one living room, one dining area, and one kitchen on the first floor and, one bathroom and three bedrooms on the second floor. It combines three types of sensory data: (1) environmental sensors, including motion sensors on ceilings, door, sensors on room entrances, kitchen cabinet doors, microwave, and refrigerator doors, temperature sensors in rooms, power meter, burner sensor, water usage sensors, and telephone usage sensors, (2) items sensors for usage monitoring, and (3) two wearable sensors. Eight complex activities are defined: Sweep and Dust, DVD Selection and Operation, Prepare Meal, Fill Medication Dispenser, Water Plants, Outfit Selection, Write Birthday Card, and Converse on Phone. Also, activities are divided into ordered steps, which can help detect whether the activity is completed correctly.

Features Engineering

The PUCK dataset has four fields (date, time, sensor ID, and sensor value). To adapt the PUCK dataset for our system, the following steps were taken to process the PUCK dataset by extracting the required features:

1. Combining the environmental data sensors (motion, items, power meter, burner, water usage, door etc.) with the wearable sensors for each participant by matching the time step among them.
2. Labelling the complex activities for the whole dataset.
3. Extracting the start and the end of each activity as a session to define when the user needs a prompt.
4. Selecting only the common sensors among all participants where the total measurement counts and participants each greater than 25th percentiles.
5. Dividing the sensors into four groups to be processed: movement sensors, motion sensors, count sensors and, continuous values sensors and process each group as following:
   (a) In movement sensors group, each measurement includes six values (X, Y, Z, Yaw, Roll, Pitch). We extracted the following features: Mean (X, Y, Z, YY, RR, PP), STD (X, Y, Z, YY, RR, PP) Correlations (X//Y//Z) and (Yaw//Roll//Pitch) which leads to 36 features in total.
   (b) For motions sensors group, if at least one trigger in a group is counted as trigger for the group, count and then compute the fraction counts across the groups. Based on the PUCK dataset we have 11 groups (features) altogether.
   (c) Count sensors which have on, off measurements such as (door, item, shake, and medicine container sensor) we count and compute the fraction counts of each session (20 features).

(d) For the last group, we calculate the average for continuous value sensors such as electricity and temperature (3 features).

6. After extracted the all features (70 features), we apply the previous groups process for all the participant sessions.

Two methods are taken to overcome the item usage imbalance problem (i.e., only a small number of items are frequently used): first, the outliers from the items are dropped. This method is simple but very effective in improving the performance and secondly, the sampling order random points in activity sessions to increase the prompt points, although this does not help balance the item usages as dropping the outliers.

5.5.2 ARAS dataset

The ARAS dataset is consisted of 2 houses of two residents where they perform 27 daily living activities: Going Out, preparing breakfast, having breakfast, preparing lunch, having lunch, preparing dinner, having dinner, washing dishes, having Snack, Studying, Having Shower, Sleeping, Watching TV, Toileting, Napping, Brushing Teeth, Using Internet, Laundry, Shaving, Cleaning, Talking on the Phone, Listening to Music, Having Conversation, Reading Book, Having Guest, Changing Clothes and Others . The activity sensory data is collected from 20 binary sensors including force sensitive resistors (FSR), pressure mats, contact sensors, proximity sensors, sonar distance sensors, photocells, temperature sensors and infrared (IR) receivers. Because of the differences between house A and house B, every house has different topology of the Wireless Ambient Sensor Networks (WASN) where house A has 2 of the Personal Area Networks while house B has only 1. The sensory data is collected for full month for each house with the time stamp of one second.

Features Engineering

The ARAS dataset consists of 22 columns where the first 20 columns represents sensors and the last columns for the activities labels of each resident. We added a new column for the time to facilitate the feature engineering. The ARAS dataset does not require complex feature engineering process because it has binary values for the whole sensors. For both houses, the activities that interact with items sensors are used for the experiments and others are removed. Unlike the PUCK dataset, the ARAS dataset does not contain the prompt points. Based on that we added random prompt points for each activity (session) and we consider them to be before the session is ended around the reward delay period T_r.

5.5.3 ADL Normal dataset

The ADL Normal dataset represents a public dataset published in 2010. It's also collected from a Kyoto smart apartment testbed in Washington State University. The data contains 20 participants performing five complex activities are defined: Making a phone call, washing hand, cooking meal, eating and taking medicine and cleaning. It is collected per second and annotated using the activity number and number of each participant. The ADL Normal combines only motion sensors, item sensors, burner sensor, phone usage and, water sensor.

Policy	Note	Hyberparameters						
		beta_prior	alpha	smoothing	decay	refit_buffer	active_choice	decay_type
LinUCB[152]	LinUCB policy stores a square matrix which has dimension equal to total numbers of features for the fitted model.	None	0.1	.	.	.	.	.
AdaptiveGreedy[211]	It focuses on taking the action that has the highest reward.	None	.	(1,2)	0.9997	.	.	percentile
AdaptiveGreedy(Active)	It is the same for AdaptiveGreedy but with different hyberparameters	((3./nchoices, 4), 2)		None	0.9997	.	weighted	percentile
SoftmaxExplorer[211]	It depends on softmax function to select the action	None	.	(1,2)	.	50	.	.
ActiveExplorer[211]	It depends on an active learning heuristic for taking the action	((3./nchoices, 4), 2)	.	None	.	50	.	.

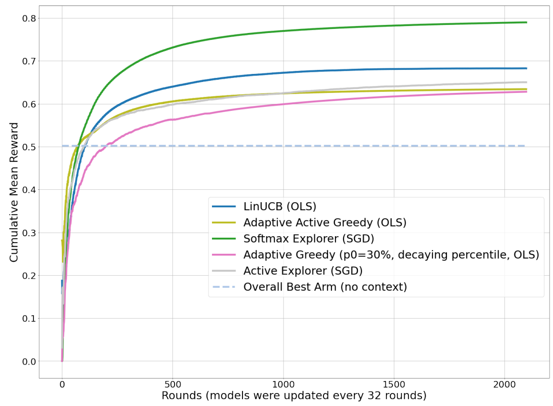
Table 5.2: Tuning hyperparameters for the OSL model policies

Features Engineering

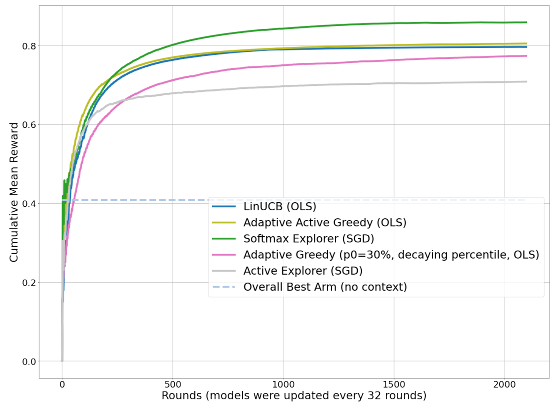
The same features engineering of the PUCK dataset is applied except that of the wearable sensors processing. Also, we add random prompt points for each activity but with considering that these points should be before the session is ended following the same process for ARAS dataset.

5.6 Evaluation

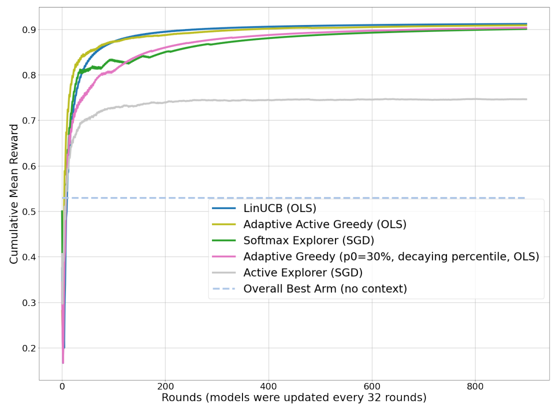
First, the evaluation of the effectiveness of the CB approach in recommending the correct item to a user in case the user's current activity needs a prompt is carried out. All the extracted features are utilized by the system as contexts to make a recommendation of the correct item. One publicly available CB python package is selected for our experiments. The package offers two types of models, full batch and streaming models. Due to sample limitation of datasets, the streaming models namely SGDClassifier (SGD) and LinearRegression (OLS) are focused on. Both models are sensitive to hyper-parameters such as beta_prior or smoothing. Nevertheless, SGDClassifier offers stochastic matrices while LinearRegression (OLS) has matrices which are closed to the solution, and it updates them incrementally. Details about the parameters are given in table 5.2 . Figure 5.4 shows a set of policies used for each model. For the PUCK figure 5.4a and the ARAS (house A) (figure 5.4b) datasets, the SGD model particularly the Softmax Explorer policy is more robust and it provides a better cumulative mean reward of item recommendations. On the other hand, ARAS (house B) figure 5.4c and ADL figure 5.4d are more likely to provide good results by LinUCB policy of OSL model. This plot confirms that both models lead to a promising results based on each dataset from several policies being used. Table 5.3 shows the cumulative mean reward of our system by considering the three datasets.



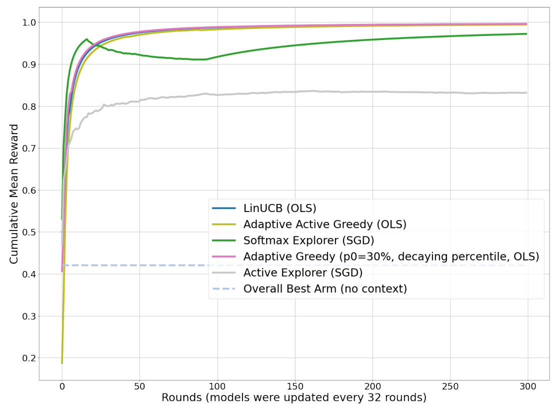
(a) PUCK.



(b) ARAS(houseA).



(c) ARAS(houseB).



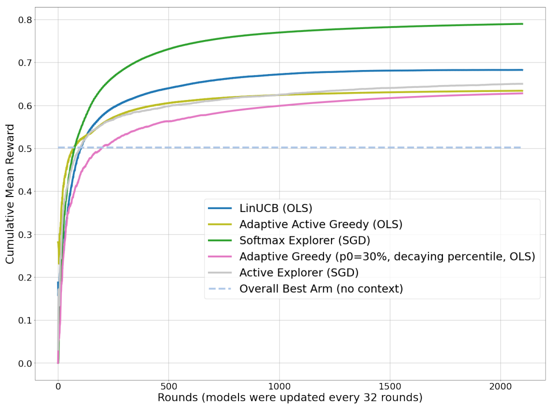
(d) ADL.

Figure 5.4: The comparison of models for each dataset.

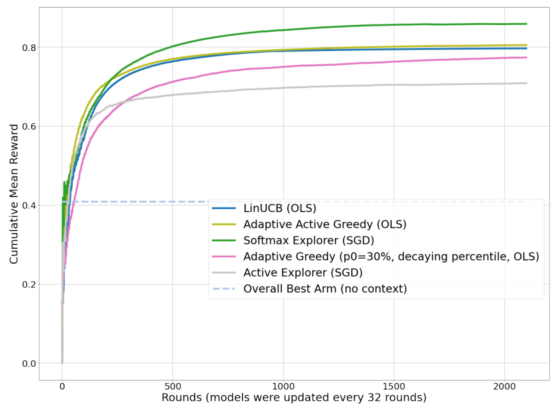
The Reward Delay Period T_r, as earlier explained, helps in the determination of the

Dataset	Policies				
	LinUCB (OSL)	Adaptive Active Greedy (OLS)	Adaptive Greedy (OSL)	Softmax Explorer (SGD)	Active Explorer (SGD)
PUCK	0.68	0,64	0.63	**0.79**	0.65
ARAS House (A)	0.80	0.81	0,77	**0.85**	0.69
ARAS House (B)	**0.92**	0.91	0.91	0.90	0.75
ADL	**0.99**	**0.99**	**0.99**	0.97	0.83

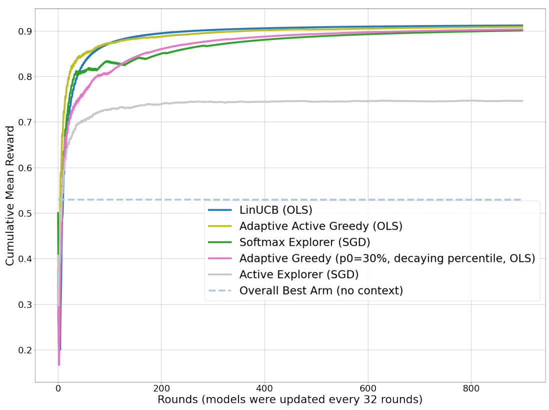
Table 5.3: The cumulative mean reward of our system among three datasets.



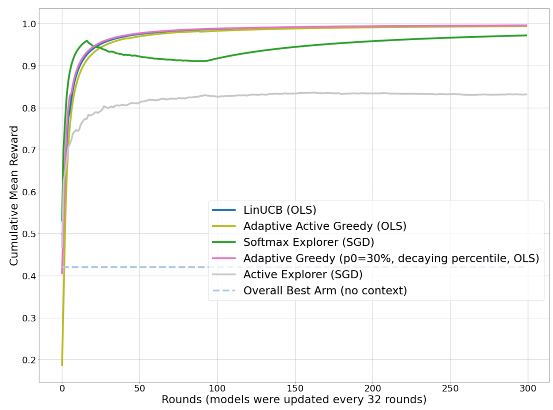
(a) PUCK.



(b) ARAS(houseA).

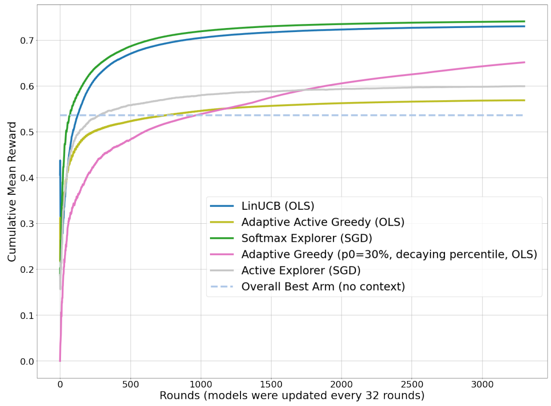


(c) ARAS(houseB).

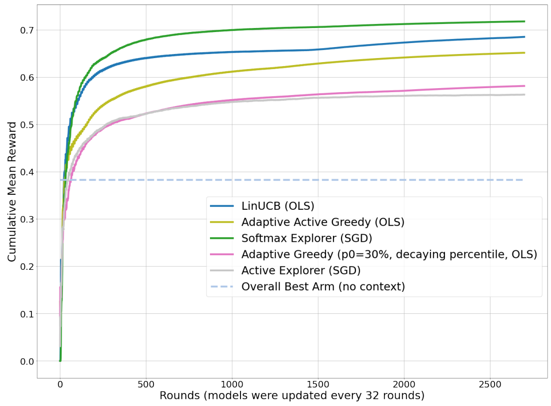


(d) ADL.

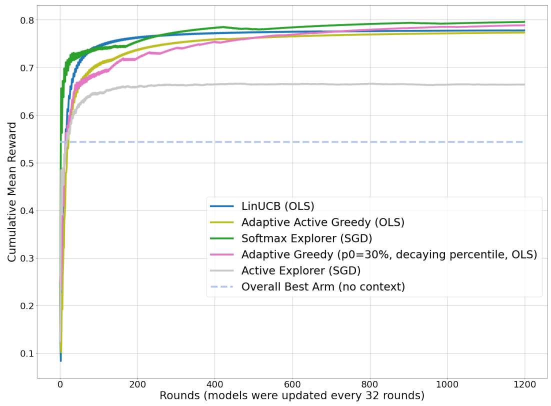
Figure 5.5: The Reward Delay Periods T_r=5s.



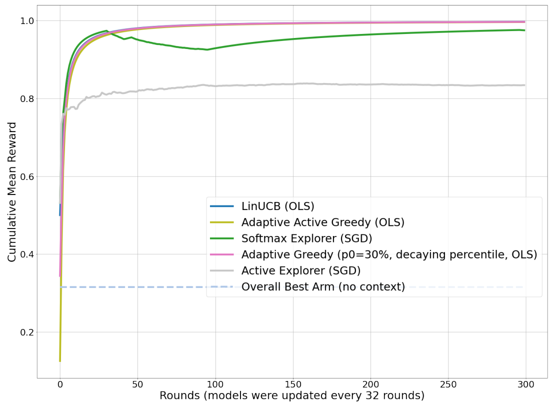
(a) PUCK.



(b) ARAS(houseA).



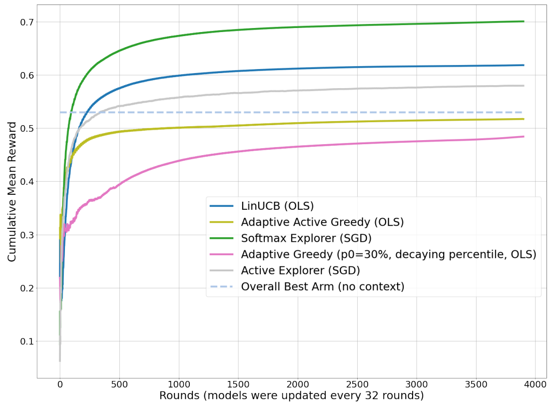
(c) ARAS(houseB).



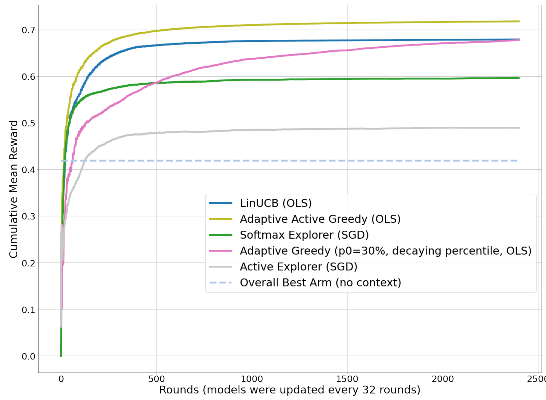
(d) ADL.

Figure 5.6: The Reward Delay Periods T_r=10s.

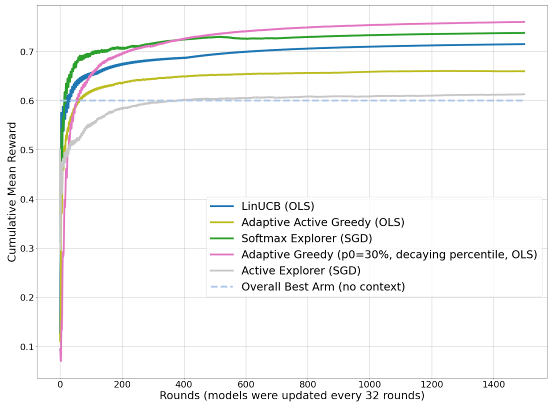
suitable time for an agent to receive the reward as a feedback of the recommended item. Tuning T_r is important as decreasing it could consider that the recommended item is not used while increasing T_r could confuse the agent specifically when the user starts to use other items before receiving the feedback about the recommended one. The results in figure 5.5, 5.6 ,and 5.7 show how T_r can affect the performance. The reward delay period with 5s (see figure 5.5) provides higher performance with the three datasets compared with the reward delay period of 10s and 15s as shown in figure 5.6 and figure 5.7 respectively. Also, policies are affected by the reward delay period such as softmax explorer of SGD model has a good performance with



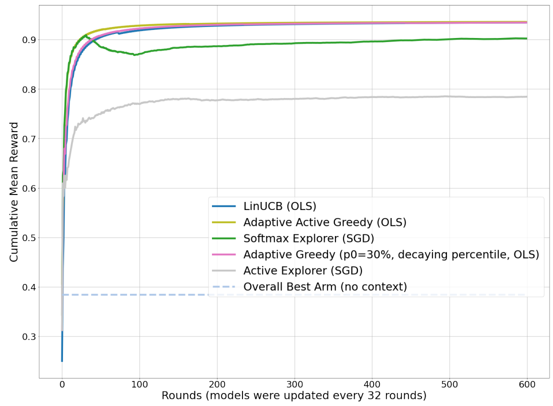
(a) PUCK.



(b) ARAS(houseA).



(c) ARAS(houseB).



(d) ADL.

Figure 5.7: The Reward Delay Periods T_r=15s.

5s among the datasets but this performance started to be reduced when the reward delay period is increased into 10s and 15s. Table 5.4 summaries all cumulative mean reward of our system among three datasets using different reward delay period.

There are two reasons that make the reward delay period of our system vary from dataset to another or from user to user in real time system. The first reason is that each activity has specific items and each of these items existing in different place. For example, preparing a cup of coffee activity includes the following items: a cup, sugar, milk and a coffee machine. So, if the agent recommends milk and the fridge was so far from the user, it will take some time to receive the correct response while if the agent recommend a cup and the user was standing near of the cupboard it will take less time to receive the response. The other reason is that users have different behaviour in their response, some of them response immediately after they receive the recommended item and others may take a little bit longer. However, our system is targeted Alzheimer's patient where it should expect reward delay period with long time comparing with normal person. Here, we treat T_r as a hyperparameter that can be adjusted based on each item; we will leave it to our future work. In addition, it is observed that the system achieved the desired result of not requiring any feedback from the user to receive the reward. Consequently, it is calculated automatically after the Reward Delay Period. This feature is important for our system because it deals with Alzheimer's patients who experience difficulty in holding a smartphone and confirm their response for recommendations.

Dataset	The Reward Delay Period	Policies				
		LinUCB (OSL)	Adaptive Active Greedy (OLS)	Adaptive Greedy (OSL)	Softmax Explorer (SGD)	Active Explorer (SGD)
PUCK	5s	0.68	0,64	0.63	**0.79**	0.65
	10s	0.74	0.55	0.65	**0.75**	0.60
	15s	0.62	0.52	0.48	**0.70**	0.58
ARAS House (A)	5s	0.80	0.81	0,77	**0.85**	0.69
	10s	0.68	0.65	0.58	**0.72**	0.56
	15s	0.68	**0.72**	0.68	0.60	0.49
ARAS House (B)	5s	**0.92**	0.91	0.91	0.90	0.75
	10s	0.76	0.76	0.78	**0.79**	0.66
	15s	0.71	0.65	0.76	**0.74**	0.61
ADL	5s	**0.99**	**0.99**	**0.99**	0.97	0.83
	10s	**0.99**	**0.99**	**0.99**	0.98	0.83
	15s	**0.95**	**0.95**	**0.95**	0.90	0.78

Table 5.4: The cumulative mean reward of our system among three datasets using different reward delay periods.

5.7 Scope of improvements directions for RCS

Despite all the advantages of the proposed system, there are still aspects that need to be considered in the future. Some of them are discussed below.

- **The RCS with real life**. As we mentioned that our system only tested on public datasets. Dealing with real time data, our system should be able for synchronization among the three stages starting from the complex activity detection till the user receives an item recommendation. We need to build our model for the prompt detection that can exactly define when the user needs a recommendation. Failure in this task makes the system constructed unbeneficial recommendations that could affect the quality of the system.

- **RSC testbed**. Building a testbed helps to evaluate our system in the real life. Also, tackling the main issue with public datasets which is missing the required features. For example, the time period of each activity whereas some activities rarely happen at night time such as Aris preparing a cup of coffee at midnight. So, if the system was feeding with the time period of each activity, it will be expected to recommend going back to bed for Aris and mentioned the time to remind her.

- **Trust-aware of the recommendations**. Our system deals with sensitive and critical data about the patient, lack of the integrity could harm the user life by conducting incorrect items such a recommended a medicine when the user already has been taken it. To ensure the safety of the recommendations, the data that feed our system need to be protected. The blockchain is planning as a potential step forwards towards to address the integrity challenge. Our previous work [121] introduce a conceptual framework for data integrity protection.

- **Unexpected action**. In some status, our system could face an issue when the user use two items at the same time and there is only a short time period between them. This case can make the agent receive a wrong feedback about the recommended item which could affect the system update. For example, if the agent recommend turning the coffee machine on whereas the user brings the milk at the same moment and then accepts the recommendation. After calculating the reward, it seems that is the milk is the correct item not the coffee machine.

- **Easy to handle**. As we mentioned before, we target Alzheimer's patient in the mild stage, our system should consider that elderly people cannot hold a phone to receive the recommendations. Consequently, designing a system that acts as caregiver for the patients is important to meet the user's expectations.

5.8 Conclusion

In this work, the feasibility of building a reminder recommendation system is explored. The recommendation system is adapted for Alzheimer's patients for only when they need a reminder. We take advantage of the contextual bandit approach to formulate our problem and tackled two main issues: dynamicity of human activity patterns and recommending the correct item without needing explicit user feedback. Experiments demonstrate the effectiveness of our recommender system. Some limitations observed in our evaluation of the system is that our experiments are still not comprehensive enough because the datasets that we use does not meet our system's requirements such as time labels which are, however, one important and critical type of context. Also, only PUCK dataset that considers the wearable sensors as a source to detect the complex activity was analysed, however, the other two datasets include items and environment sensors. The number of samples and complex activities in each dataset is also considered as limitation that affect our experimental results. In the future, we are creating our own test-bed to collect inclusive and adequate data for complex experiments, and testing our framework in real-life scenarios.

Chapter 6

A Blockchain Framework Data Integrity-enhanced Recommender System

6.1 Introduction

Recently, Recommender system for the IoT (RSIoT) have been adapted in variety of applications to meet user preferences, and such applications can be smart home, smart tourism, smart parking, m-health and so on. Most of previous works focused either on modelling the recommendations engine using different approaches or selecting data sources to feed the system. However, ensuring the integrity of the data has not been received enough attention. In [197], recommender system for the IoT called RCS was proposed which supports Alzheimer patient to live independently and safely in their own homes. Using Alzheimer disease as use case led to an inspiration to dive in the most inherent characteristics that not only meet the users' preferences but also to protect them of any harm could be caused by using the system. Ensuring the data integrity is one of the crucial requirements of any system where in Alzheimer patient case any failure could cause user's life. Here the system not only conduct item recommendations but also providing a health recommendation from medical professional; we need to be certain that all these health recommendations are built on data with high integrity. Data integrity is defined as following: maintaining and assuring that all data are accurate and consistent over its entire life cycle.

The following scenario can demonstrates the motivation of this manuscript: Alice is 79-year-old woman with mild AD who experienced a heart attack last month and who lives alone. Our RCS [197] allows authorised personnels to monitor her health status, and the information can be made available to all hospitals in the city. It will also ensure that Alice receives a timely medical response in the case of a critical health issue. Her house will be equipped with passive RFID tags and medical sensors to keep an eye on her activities and movements. Biometric sensors will also be in place to monitor cardiac activity, glucose levels, temperature, CO_2 levels, brain activity, blood pressure, GSR stress levels, and oxygen blood levels. Such vital data can be collected and analysed in real time and stored for future access. For example, when her body temperature sensor yields a reading above $39C^{\circ}$, this value will be classified as sensitive data and will be sent it to Alice's doctor as an emergency notification. Alice' doctor can access her health data that can help in taking a medical decision. However, before taking any decision, the doctor should ensure the integrity of received data more precisely, whereas the data is accurate and consistent since it was sent till it is received by the hospital. Otherwise, any decision could affect Alice' life or could waste in the hospital resources.

Recently, Cloud Storage Service (CSS) [212] is considered the most popular way in processing big data, particularly the data that is produced by IoT devices; however, with CSS the third party cannot be avoided, and users can no longer control their data which means data integrity at risk. In light of healthcare data breaches, some statistics refer to that around 249.09 million were affected from 2005 to 2019 [213]. Decentralizing RSIoT can address numerous of security and privacy issues including data integrity. It passes data without relying on trusted third parties for handling any critical data for users especially the sensitive health data about patients like in our case. The Blockchain is distributed ledger technology has been gaining attention after the Bitcoin whitepaper was published [214]. It not only impacts the financial world but also it is considering a rise technology that provides a critical impulse in protecting security and privacy data. In data integrity, there are a number of works have been introduced. Zikratov et al. [215] proposed a blockchain-enabled system to protect the integrity of outsourcing file. The system consists of two main parts frontend which enables the user of uploading/downloading data and backend that contains all the implemented services to verify the data integrity. In [216], authors proposed the design a novel approach that combined the IoT with permissioned blockchain network to address the data integrity challenge.

Although the concept of blockchain is straightforward, it's adaption for RSIoT poses some critical number of challenges as the following: IoT devices cannot afford power consumption to run blockchain particularly consensus mechanisms; big data production from IoT makes it inappropriate for us with the blockchain specifically, when every node in blockchain network maintains a copy of all data; and blockchain cannot communicate straight forward with outside world like IoT devices.

The common theme in this chapter is leveraging blockchain capabilities in decentralization to ensure the data integrity in RIoT. To the best of our knowledge, no previous research works on this theme. This chapter presents recommender system for the IoT based blockchain which ensures data integrity during its entire life cycle without dealing with third party services. The main contributions of this chapter are as follows:

- Proposition of the RCS based on blockchain (Ethereum) to ensure the data integrity for the critical data.
- Design four smart contracts: emergency notification contract, health data access contract, health data aggregation, and medical advice contract.
- Evaluation of the developed smart contracts using two test networks

6.2 Related work

In this section, we discuss some of the related works have been done by considering two main aspects: Blockchain based smart m-health and recommender system with blockchain .

6.2.1 Smart m-health based blockchain

Recently , smart m-health elaborates that shifting the current centralized IoT system towards the decentralized ones is the right way to address numerous of security and privacy issues. In [217], authors adapted blockchain for medical system to tackle three main issues of current centralized IoT system which are a single-point failure, lack of security, and high loading of data and computing. Authors in [218] developed m-health system based on blockchain where it has ability to resistant any tamper and to secure the system. In [219],a blockchain technique as protocol is applied to manage the commutations among nodes in Pervasive Social Network. However, adapting blockchain for m-healthcare is not straightforward and needs tackling the critical issues such as high resource consumption, latency and scalability. In [220], a remote patient monitoring system is proposed using a customized blockchain. The idea of the customized blockchain is to select a miner which reducing the computational effort, managing multiple blockchains and and using prefix tree in modifying a block to minimize the energy consumption. In [221] a novel blockchain for telecare medical information systems (TMIS) is proposed. It provides the required characteristics: decentralization, confidentiality, multi-level privacy preserving, retrievability and verifiability to ensure multiple level of privacy among TMIS parties. In [222], consortium blockchain with smart contract were adapted to secure both the transfer and accessing data of a health care system based on the IoT. The smart contract manages the data transmission between the patient and the service provider based on predefined conditions. Authors in [223] integrated two main technologies: blockchain and cloud to address the challenges of dealing with medical data: heterogeneity and interobablility. Data integrity is considered one of the critical security issues involving both data storage and data transmission. In [224], a private blockchain and off-chain were introduced to address the third party issue in a remote health monitoring system. In [225], the authors tackled the security issue in massive IoT data of healthcare system by adapting blockchain. Here, the time latency of blockchain is taken account by proposing sharding schema. Authors in [226] proposed electronic health wallet (EHW) system based using blockchain which provides the patients the full control of their data to ensure the privacy.

6.2.2 Recommender system with blockchain

RIoTs play a critical role in many smart applications such as smart marketing, smart tourism, smart homes, and smart m-health. It will be impossible to construct recommendations with high quality without a massive amount of data about the users. However, security and privacy are still intriguing problems that may be absent in such systems. Combining RS and blockchain technology can allows previous applications to be more secure and private. Authors in the paper [227] proposed a system for smart pharmaceutical industry based on blockchain which can face the counterfeit drugs issue. Two modules are included: block-chain model to secure the drug supply chain management and recommender system module to recommend the best medical option for the user. In [228], a platform for a recommender system based on blockchain is designed. It exploits permissioned blockchain to address one of main concerns about the user privacy where the user information is secured by limiting the access only for the platform instated of companies. Paper [229] proposed a Decentralized Tourism Destinations Recommendation System is proposed using blockchain to support data sharing among three nodes user, sensors and server. In [230], a consortium blockchain was adapted which enables the data sharing between blockchain and cloud in high security way to address distributed service recommendation. Authors in [231, 232] Authors adapted a blockchain technology to secure collaborative filtering approach. The recommender system is built on top of blockchain to uses the stored data in constructing the recommendations which preserves the user privacy. In [233], a recommendation mechanism based on blockchain is proposed. In this mechanism, Inter-Planetary File System was combined with blockchain to enhance the quality of the communications. Also, two techniques are included to minimize the computation load and to ensure the privacy: local sensitive hashing and local differential privacy. Authors in [234] proposed a recommender system based on collaborative filtering and blockchain. The proposed system aims to conduct trusted online recommendations by securing the input data using blockchain.

6.3 Background

We first provide an overview of the three main techniques used in our framework, followed by a detailed description of the framework.

6.3.1 Blockchain

Blockchain is considered a promising technology that involves sequential chained blocks including ledgers that are replicated in each node of a peer to peer network. Firstly blockchain is used to store financial transactions that are shared by all users ; consequently, any manipulation in these transactions can be easily detected. The typical workflow of a blockchain system is summarised as the following (see figure 6.1): when a transaction is created, a block is made; then all nodes in the network receive a broadcast about this block and one of the nodes (called miner) validates the block and broadcasts it to all nodes; after the block is added to the chain by the nodes when they verify that it has a hash number to connect with the previous block [9]. To solve the reliability issue of verifying a new block in a network, varied consensus algorithms. The proof of work (PoW) algorithm is considered the most widely used where each node has to perform some of complex computation in mining a new block and proving it's validation as a member of the network. Blockchain technology is categorised into two categories: Permissionless (public) and Permissioned Blockchains (private, consortium). In Permissionless, every user has a right to join the network and to become a node. Also, it considers more secure because of the numbers of nodes so, it is complicated for the malicious nodes to collude on the network. However, because increasing the number of nodes with their transactions the Permissionless consumes more time during transaction processing. Also, the size of each block makes the Permissionless blockchain facing a severe scalability issue where each block has size of only 1MB to include seven transactions per second.

In the other side, permissioned blockchain tends to address this limitation by restricting the access to the peer to peer network which reduces the number of nodes in the network. However, this way for addressing the limitation could affect the security level where malicious nodes have more opportunities to consist. Choosing the best category depends on the nature of the proposed system. For our system, we chose the Permissioned Blockchain category specifically, private blockchain (Ethereum) so, as we mentioned in the previous scenario that Alice could need medical recommendations in timely manner so, the speed of transactions matters in our case. Another reason is that our system has predefined parties which includes doctors and patients.

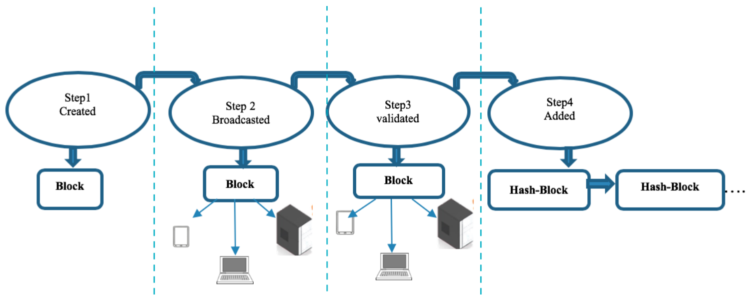


Figure 6.1: Exemplary blockchain process[9].

6.3.2 Ethereum and Smart contract

Ethereum is an open-source platform of blockchain and was introduced in 2014 to address several limitations of public blockacin. One of the main issues that Ethereum addressed is the scalability where comparing with Bitcoin, the Ethereum can process around 11 transactions per second which means over one million transactions in 24 hours. It aims to be an alternative protocol that helps in constructing decentralized applications. Also, the consensus algorithm in the Ethereum network tackles the problem of stale blocks when some number of blocks have heavy computing power for mining more than the others that leads to centralization issue. The Greedy Heaviest Observed Subtree(GHOST) [235] protocol in Ethereum addresses this issue by removing the centralization where the stal block continues receiving a reward after mining a block however, only 87.5% and the remaining reward will be receiving by the nephew of the stal block. In addition, Ethereum supports smart-contract that gives users more possibilities of creating their own rules, formatting transactions, stating transition function, and interacting with parties that have no trust relationship. The smart contract [236] is an object connected with their blockchain account which includes programs for storing data, making decision, and communicating with other smart contract. The smart contract is constructed as a new layer on the top of blockchain by the owner and can be executed by Ethereum without any commitment of other party or losing time.

6.3.3 Blockchain oracle

The smart contacts have a limitation where they cannot access or even resides data from outside the blockchain. Oracles are considered as a solution which can feed the smart contact with external data. Blockchain oracle is a bridge that connects blockchain with smart contracts to the outside world. It plays a vital role: in providing a link between on chain and off chain also in querying, verifying, and authenticating external data sources. Blockchain oracles can be categorized into three main categories. Firstly, source of data where oracles can interact with online data sources (software) such as (1) databases, servers, websites ,etc to provide the required information; (2) getting the data from some sensors or any other reading devices as it is in our scenario; or (3) dealing with specialized of human who has knowledge/skills in a specific field. The second category is data orientation to define the data direction if it is from the data provider to smart contract(inbound) or from smart contract to the data provider(outbound). The last category is trust which includes centralized and decentralized oracles. However, centralized oracle face more security issue (single point of failure) where it gets the data only from one source. Unlike centralized oracle, decentralized oracles address the single point of failure and distribute the trust among many participants. One example of decentralized oracles is chainlink which will be explained with details in the following section.

6.3.4 Chainlink

Chainlink is considered a decentralized oracle and it plays a vital role in gathering the required data from outside world to execute the smart contract. It aims to gather the dynamic data from various data providers outside blockchain that makes smart contact more efficient and powerful. In Chinalink, there is a fee that is paid to an oracle in form of a LINK token to get and to process the required data. For our system, chainlink is an ideal solution to feed the smart contract with the off-chain data. As we mentioned in Alice scenario, monitoring her health needs to gather massive sensory data per half second. The huge amount of data that our system is produced cannot be stored in blockchian directly; however, storing these data safely off chain and making it available to access using chainlink is the best solution to this issue. In the following points, we summarize the chainlink process:

- Matching oracles with blockchain operator to provide the required information. A service level agreement (SLA) software is used here to provide the required information based on the conditions and terms that are determined in the smart contract. For example, Alice's doctor may want to receive an emergency notification only if Alice has high blood pressure. So, chainlink will match the blockchain operator with only oracles that provide more accurate information that met the request.

- After selecting oracles, they will obtain the required data from each data provider (her blood pressure 140/90 mm Hg), the oracles will process this data and make it available to the smart contract.
- The extracted data of the previous step will be forwarded through chainlink to the operator's blockchain. It represents as an input for the smart contract. For example, if blood pressure rate is high above a normal level, the doctor will receive an emergence notification.

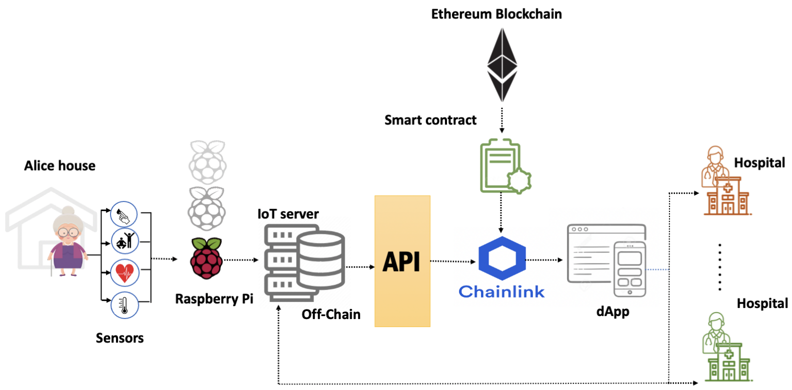


Figure 6.2: Overview of proposed framework

6.4 Proposed framework for reminder care system (RCS) based blockchain

This section discusses the main components of our framework as shown in figure 6.2.

- **Sensors** where we focus here only on medical sensors that collect health data about Alice such as temperature, blood pressure, heartbeat, and blood sugar. All the previous sensors are connected to Raspberry Pis

- **Raspberry pi** receives the sensory data and froward them to the server.
- **Server** is responsible for storing the data; so, the sensory data are produced every half second which considers massive data cannot be stored on the blockchain immediately but instead we store them off chain.
- **API** represents the interface that provides the required information by calling them from the sever; so, doctors and users have the access to send or receive data.
- **Chainlink** is considered as mediator between the blockchain and data from sensors where it provides the required data to execute a smart contract. each Oracle represents a node in the chainlink network, while a Job is a specific task that an Oracle can do. An Oracle might be able to do many jobs, so we have an Oracle ID to specify which Oracle we're using and we also specify the JobId to identity the particular job we want this oracle to do. An oracle specifies a fee to be paid for each job to be performed.

  For example, if the Job that is specified in the smart contract via a Job ID is an HTTP Get >INT256, then we're basically performing a routine that makes a GET request and returns an integer value as the following:

```
{
  "initiators": [
    {
      "type": "runlog",
      "params": {
        "address": "0x9c0383de842a3a0f403
        b0021f6f85756524d5599"
```

```
      }
    }
  ],
  "tasks": [
    {
      "type": "httpget"
    },
    {
      "type": "jsonparse"
    },
    {
      "type": "multiply"
    },
    {
      "type": "ethuint256"
    },
    {
      "type": "ethtx"
    }
  ]
}
```

- **Ethereum Blockchain** was chosen for our system based on previous reasons that were mentioned before. Furthermore, Ethereum is dealing with smart contact and supporting the dApp that we included in our system .
- **Smart contracts** is built to manage our system where it defines which kind of

data should be passed through the blockchain. For example, one of the smart contract is written to send an emergency notification of Alice's temperature to the doctor over the blockchain network only if it is over the threshold(>37); otherwise it will be stored on the off chain. The doctor can access Alice health data anytime and provides any health recommendations (more details in section).

- **dApp** is a decentralization application that acts like any conventional applications but it operates on decentralized network instead. It provides numbers of features such as decentralized, meaning it can work as an open platform without being controlled by any third party; it is available to be accessed at anytime no downtime; it is secluded where any error on the smart contact can not affect the blockchain network, and finally it provides data integrity where the storage is incontrovertible and immutable on the blockchain.

The previous framework is not limited to Alice scenario, but for demonstrating the benefits of the framework. So, our framework can be generalised to a wider range of application domains, such as smart homes, to enhance the deployment of privacy-preserving IoT services. However, to exemplify our goals, we use this scenario in the rest of the article. As mentioned previously, we used a IoT devices that collect the required data to monitor and to manage Alice's health. More specifically, measuring temperature, blood pressure, heart rate and blood sugar and all the sensors connected to raspberry pi. To manage the data that are flooded into the system, by dividing the data into two categories: normal and critical data. The normal data includes all sensor reading that matches the normal threshold will be store in off chain(see figure 6.3) and Alice's doctor can access them anytime using the API. The critical data includes the sensors reading that do not match the normal threshold such as a temperature's sensor reading around 39 C° so, this kind of data needs to be sent to Alice's doctor without any tempering and with high data integrity protection. Consequently, Alice's doctor decides which medical advice recommendations should be provided for her at this moment. Figure 6.4 presents the process of our framework when the sensor produces a critical data. Here, the smart contact is executed when the required data sources are provided. So, Alice's temperature (39 C°) passes through the chainlink using number of oracles to ensure the integrity of the data. The oracles connect the blockchain with the data provider; when the transaction is received on the blockchain, it will be immutable and decentralized so, the doctor can receive the emergency data (following the red lines in figure 6.4). After Alice's doctor received the data, the medical advice recommendations can be provided immediately to Alice passing through the same process as shown with the green lines in the figure 6.4.

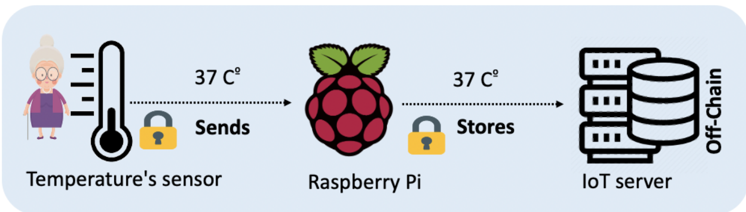


Figure 6.3: The processes of storing the normal sensory data off chain. Here, the data from temperature sensor is sent to the Raspberry Pi every half second and then it's stored in IoT server to be available for accessing at anytime

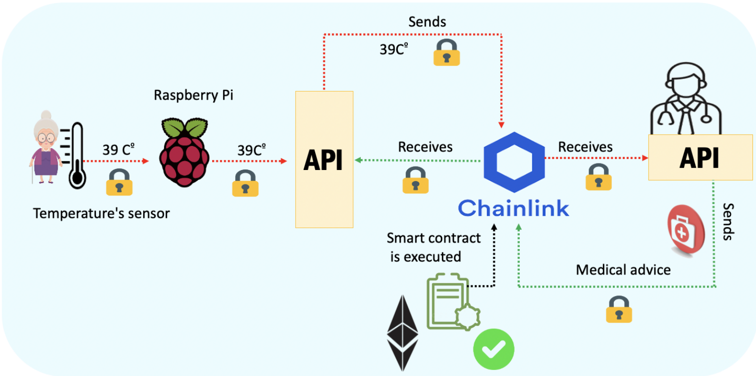


Figure 6.4: The processes of sending the critical sensory data on the blockchain.

6.4.1 Managing the critical data using smart contract and Ethereum

This section discusses how the critical data can be managed by smart contract using Ethereum.

One of the most significant feature of Ethereum is that supports the use of smart contracts. The Ethereum brought the benefits of the smart contract to enable the developers gain more advantages of blockchain by writing their own program and executing it on the top of blockkchain. In [237], the smart contract concept was introduced. Smart contact can be written in different programming languages namely Solidity, Pact, Liquidity [238]. The solidity language [239] is widely used and considered as the primary choice to write smart contracts and it has similar syntax to JavaScript. For our system, we designed four smart contacts that meet the goal in our case. One contract is to send an emergency notification to Alice's doctor when her health measurements increase the normal threshold. The second to enable her doctor of accesses her own health information that are stored off-chain. The health data aggregation to call the other smart contracts that collect specific health data. The last contact is to push medical advises to Alice during her day. However, we notice that building smart contracts can be the main challenge. All smart contacts should be predefined and written on the top of the blockchain to investigate the goals of a system and the requirements that needs to be met. In addition, the content of each smart contact could affect the cost for each transaction as we will explain in the following section 6.5.

1. **Emergency Notification Contract**.

   IoT devices send value readings to a Raspberry Pi around every half second. More precisely, each Raspberry Pi is connected to the internet and can send the values through an API to the smart contract which monitors the sensor's reading, If it increases above a certain threshold, an alert is created. The raspberry pi sends the following: value (Ex. 39), public Key, and signature first to the API, then ChainLink and finally the Ethereum Blockchain. As seen below an example for a temperature emergency contract to send an emergency notification to Alice's doctor about her higher temperature. Notice, figure 6.5 shows only an example of emergency notification contact for temperature sensor but the same can be done for different kind of sensors. The temperature value is an input the alert function and it the temperature value is above a certain threshold, the Report event is emits the user address and the current temperature reading. This also helps to keep Alice anonymous since the only data we know is her Ethereum address and a temperature reading

2. **Health Data Access Contract**.

   As we mentioned in our scenario, medical health sensors collect all health information about Alice so her doctor can access them at any time using the smart contact. At the beginning, the doctor can request specific information

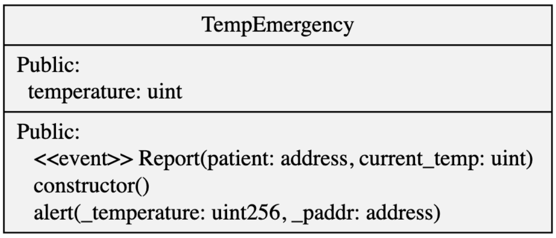


Figure 6.5: Emergency Notification Contract.

using his API, then the request will be executed as smart contact on the top of Ethereum. ChainLink is responsible for allowing the data that is sent to the doctor accessible through the blockchain. Figure 6.6 and figure 6.7 shows smart contacts that responsible to provide data from two kinds of sensors: temperature sensor and blood pressure sensor.

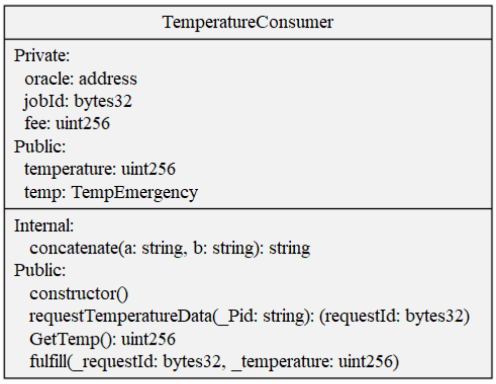


Figure 6.6: Temperature Consumer Contact.

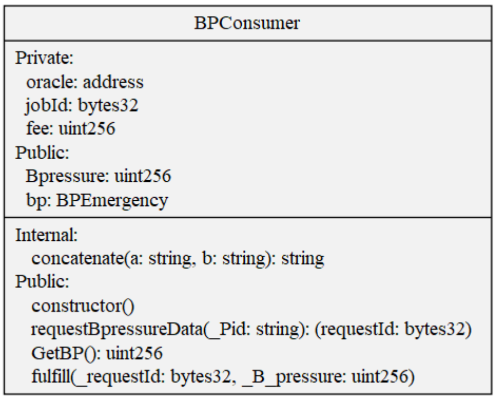


Figure 6.7: Blood Pressure Consumer Contact.

3. **Health data aggregation contract**.

   In some cases, Alice's doctor needs to access her health information to monitor her health condition or check her health status. An aggregate data contract calls the other smart contracts such as TemperatureConsumer, and BPConsumer contracts that fetch specific data and return all the data at once.

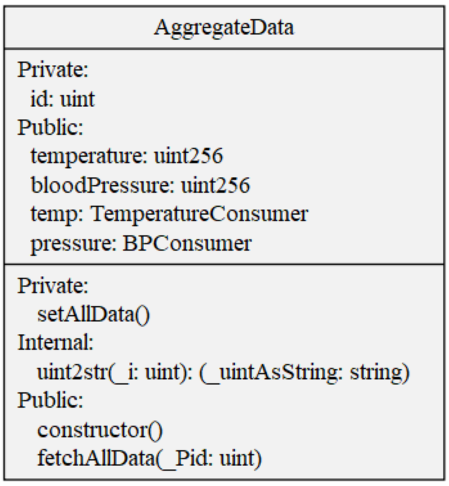


Figure 6.8: Aggregate Data Contact .

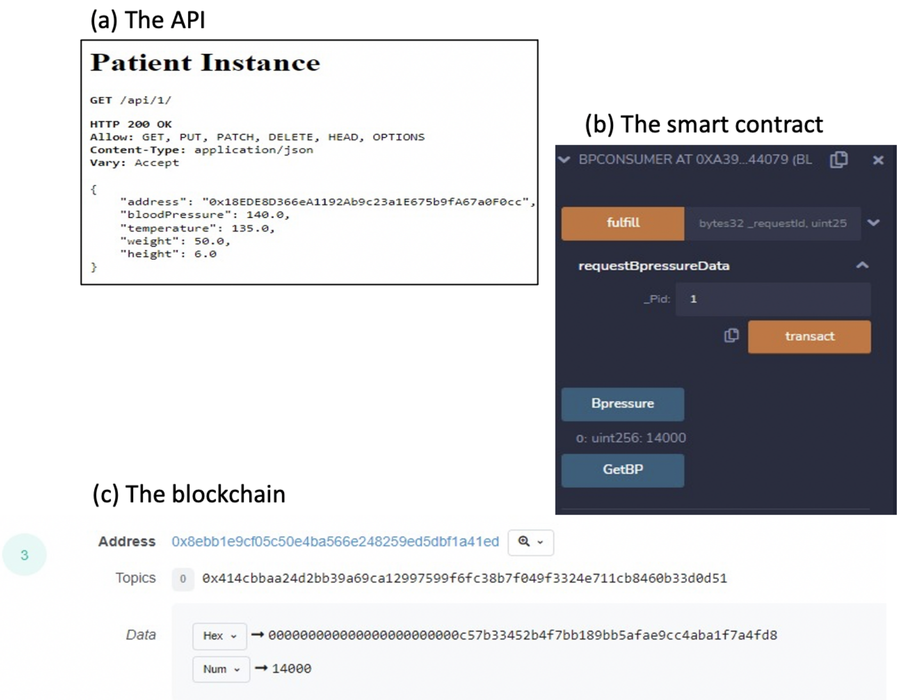


Figure 6.9: An example shows the data integrity for a patient's blood pressure.

Figure 6.8 shows a smart contract that aggregates data from temperature contract and blood Pressure contract. The Aggregate Data Contract Calls the other smart contract that provides specific health data and returns the values of all the health data at once. To ensure the data integrity, the data should be the same since it is sent till to be received. Here, we provide an example of sending the blood pressure for a patient using health data aggregation contract. The blood pressure on the API is 140 (see figure 6.9(a)), then querying the API using data aggregation contact is created (see figure 6.9(b)). finally, the receiver will get the same blood pressure on blockchain as shown in figure 6.9(c) .

4. **Medical Advice Contract**.

   After the emergency contract reports an emergency and emits the user address and current health reading, the doctor takes the address and uses the advice contract to send a text based advice over the blockchain to the user. Figure 6.10 shows an advice contract that can send a health recommendation advice for Alice. For example, when Alice's doctor finds her temperature is more than 37, a medical advice recommendation can be sent to her immediately such as taking two bills of paracetamol.

6.5 Evaluation

In order to evaluate the performance of our smart contracts for the proposed framework, we used two testing blockchain environment that are maintained by Ethereum: Kovan [240] and Rinkeby [241] Test Nets. Both proposed to tackle a denial of service attack(DoS) issue in Ropsten Test Net by utilizing Proof of Authority (PoA)

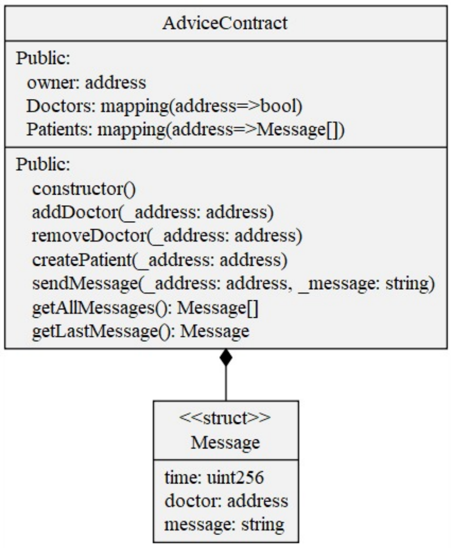


Figure 6.10: Advice Smart Contract.

consensus engine instead of Proof of Work(PoW) that removes mining process that could occur by malicious actors. Moreover, Kovan and Rinkeby have shorter time in adding a new block and less cost to maintenance their network. Both provide an environment to test smart contracts in the similar way of main Public Ethereum network. The main advantages of using the two Test Nets is that there is no financial required as in the main Ethereum network. As we mentioned before there is no mining process in both networks. Consequently, in our evaluation, we focus on two main aspects: execution time for each smart contact and fee (ETH) for each transaction. Here, the API system is hosted on a cloud based on AWS m3.xlarge node, and its specifications are as the following: 15 GiB of memory, 4 vCPUs, 64-bit platform. The program is granted a 1 GB Disk quotation and our API was written with Django and Python 3.9.

In table 6.1, we present a qualitative explanation about transaction cost for smart contracts in both test networks which are considered as a baseline of how the contracts will operate on the main networks.

Smart contract	Kovan Test Network	Rinkeby Test Network
DataAggregate	0.00065201	0.00064623
TempData	0.00049531	0.00049531
BPData	0.00049521	0.00049521
TempEmergency	0.00049531	0.00046421
BPEmergency	0.00049509	0.00045391

Table 6.1: Transaction Cost Comparison

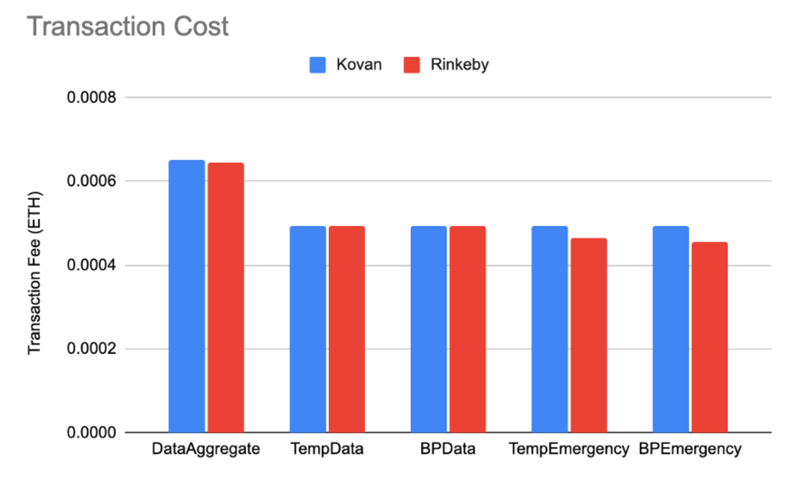


Figure 6.11: Transaction Cost in Kovan and Rinkeby.

Figure 6.11 shows how smart contracts content can affect the transaction fee on both networks. There are slightly differences in the three contracts: TempEmergency, DataAggregate and BPEmergenc. However, TempData and BPData have the same ETH cost. For the execution time as shown in figure 6.12, We can observe that Kovan has better performance compared with Rinkeby. For example, the smart contract to aggregate data about Alice health status executed in 12500 milliseconds in a Kovan and in 15000 milliseconds in Rinkeby. Notice that all the smart contracts of each Test Net have the same execution time; the reason is that on Test Net, blocks are mostly created by the same miner.

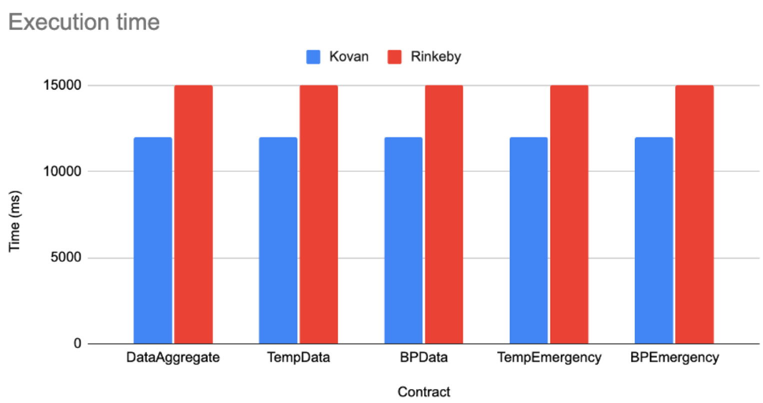


Figure 6.12: Smart Contracts performance in Kovan and Rinkeby.

6.6 Conclusion

The combination of IoT and blockchain has demonstrated the ability to build recommender system that not only aims to meet the user preferences but also to ensure the integrity of data. Blockchain provides decentralized system that allows peers to communicate in trust and resilient manner. Here, smart contracts enable us to gain more advantage of the blockchain by writing our own program and executing it on the top of blockkchain. Thus, this chapter proposed a reminder care system (RCS) based on the blockchain, exemplifying the usability of blockchain in order to ensure the data integrity. We created smart contacts based on requirements that we need to investigate. We tested the smart contracts on two Test Net.

Chapter 7

Conclusion

In conclusion, we have summarized a few research challenges in RSIoT and surveyed the recent state-of-the-art the recommender system for the internet of things techniques and its applications in a variety of domains. The DeepConvLSTM and ontology are utilized to detect elementary and complex activities which represent the input for our system. Two main challenges of RSIoT are received intensive attentions in this dissertation: the dynamicity with the ability to learn the user pattern and updating the system automatically without needing feedback from users. Deep reinforcement learning particular DQN algorithm is adapted to address the first issue. The reward function of the system is formulated to help in updating the system automatically without needing feedback from the user by checking the status of items after a period of time. To provide an accurate recommendation in various scenarios, contextual bandit approach is explored. It outperforms the deep reinforcement learning approach for main reason which is considering the context as a feature helps the system to learn the various scenarios from the user. Because the data integrity was considered a main concern for our system, blockchain is presented. We designed four smart contacts that serve our goal in providing data integrity during data transmission without dealing with third party services.

To develop full potential of RS in the IoT, some future research directions are worthy of further investigation. The challenges that are summarized in this thesis can be stimulated as future directions. Here, devoted effort is provided to address RSIoT challenges; however, there are still not fully explored. Despite the current research studies not provide a reliable and comprehensive solutions for the issues, they present concrete foundations and lay guidance for future directions.

Here we detail future work directions to give opportunity for this research to be extended in this dissertation.

- ***Dynamicity in RSIoT.*** During this dissertation the dynamicity is considered one of the main challenges. The key issue is that human activity can be done in various way not only the activities but also the order items that are included in each activity and the time that could be consumed for each recommended item. A promising way is to learn features that helps the system to constructing accurate recommendations is required. We assume the reward delay period to be used in the system updating. However, LSTM [242] is a suitable to define the exact time for each recommended item.

- ***Security.*** In this dissertation, data integrity was one of our goals particularly when the system deals with critical data such as Aris's health information. Ensuring the integrity of these data helps the system in conducting accurate medical advice recommendation to patients. Blockchain were adapted to ensure the data transmission between two parties and to eliminate third party service. However, ensuring the data integrity during the data collection were not covered. In [121], a conceptual framework was proposed to protect the data during its total life cycle using three main techniques: secret sharing, blockchain and fog computing.

- ***Multi-Source Contextual Correlations.*** As mentioned before, accurate recommendations need quality of features to be feed into the system as input. Here, in Aris's scenario, we focus on specific kinds of sensors that met our goal. However, expanding in collecting more data about the user helps in improving the service recommendations. For example, the system can not only recommend an item to the user but also can provide a healthy diet by counting the required information about patients using additional wearable sensors.

- ***New activities identification.*** During the first stage of our system, only the complex activity that have a ground truth will be identified. As a result, a reliable approach should be proposed which provides the required features to detect a new activity that has never seen before. In [243], a new approach was introduced which focuses on learning mid-level features representation to depict activities. Also, in [244], a multi-head CNN framework was presented which helps in learning discriminant features.

- ***Contextual bandit for IoT Recommendation***

  Contextual bandit has been extensively used for various tasks where the context plays main role to take a decision including our system. Contextual bandit emerges and draws increasing attention on various occasions for two main reasons. First, including the context as feature helps the system to conduct accurate recommendation to the user in timely manner. It could also be impossible to depend only on the history data as features where the human has various pattern for each activity and previous information about the user is not enough to decide the correct item for the current situation. Another reason is that contextual bandit treats each state independently unlike other approaches that each state affects the next.

- ***The automatic feedback.*** Receiving the feedback about recommendations automatically was one of the main goals of our system. We introduced the Reward Delay Period parameter which helps our system to decide if the recommended item has been used or not. However, we adjusted this parameter randomly. Finding a technique that can adjust this parameter based on the nature of items is crucial to improve the quality of recommendations.

- ***User Friendly.*** As we mentioned in Aris scenario, the patient does not need to hold any smart device or phone. However, the system still needs to be improved in term of the way of conducting recommendations to the user. Voice recommendations are considered the ideal way for Alzheimer's patients.

- ***Brain Activity Recognition.*** In our system, we consider Alzheimer's patient in their first stage as use case. However, the system can be improved to handle advanced stages by considering the brain activity recognition [245, 246].

- ***The state-of-the-art of a unified RSIoT framework.*** While numerous of works have been investigated in RSIoT, there lacks a unified RSIoT framework for fair comparison. There is quite variation from paper to paper in various perspective such as data sources, experiments sitting and evaluation metrics. As a part of the evaluation, RSIoT heavily relies on data sources in constructing the high quality recommendations. However, sufficient public datasets that includes the required features for RSIoT were not accessible. Therefore, it is noteworthy that having a testbed that is created based on the goal of the system is a sufficient solution not only to collect the required data but also to test the system in the real time.

List of Publications

This dissertation contains parts of contexts of the following publications:

[1] B. Yong, Z. Xu, X. Wang, L. Cheng, X. Li, X. Wu, and Q. Zhou, "Iot-based intelligent fitness system," *Journal of Parallel and Distributed Computing*, vol. 118, pp. 14–21, 2018.

[2] H. Yoo and K. Chung, "Mining-based lifecare recommendation using peer-to-peer dataset and adaptive decision feedback," *Peer-to-Peer Networking and Applications*, vol. 11, no. 6, pp. 1309–1320, 2018.

[3] A. Hassani, P. D. Haghighi, S. Ling, P. P. Jayaraman, and A. Zaslavsky, "Querying iot services: A smart carpark recommender use case," in *Internet of Things (WF-IoT), 2018 IEEE 4th World Forum on*, pp. 619–624, IEEE, 2018.

[4] A. Yavari, P. P. Jayaraman, and D. Georgakopoulos, "Contextualised service delivery in the internet of things: Parking recommender for smart cities," in *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*, pp. 454–459, IEEE, 2016.

[5] Y. Salman, A. Abu-Issa, I. Tumar, and Y. Hassouneh, "A model and prototype of a proactive multi-type context-aware recommender system," in *Intelligent Systems Conference (IntelliSys), 2017*, pp. 297–303, IEEE, 2017.

[6] Y. Saleem, N. Crespi, M. H. Rehmani, R. Copeland, D. Hussein, and E. Bertin, "Exploitation of social iot for recommendation services," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 359–364, IEEE, 2016.

[7] M. Amoretti, L. Belli, and F. Zanichelli, "Utravel: Smart mobility with a novel user profiling and recommendation approach," *Pervasive and mobile computing*, vol. 38, pp. 474–489, 2017.

[8] K. Matsui and H. Choi, "A recommendation system with secondary usage of hems data for products based on iot technology," in *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6, IEEE, 2017.

[9] M. Banerjee, J. Lee, and K.-K. R. Choo, "A blockchain future for internet of things security: a position paper," *Digital Communications and Networks*, vol. 4, no. 3, pp. 149–160, 2018.

[10] S. Zhou, C.-H. Chu, Z. Yu, and J. Kim, "A context-aware reminder system for elders based on fuzzy linguistic approach," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9411 – 9419, 2012.

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[12] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," *Transaction on IoT and Cloud computing*, vol. 3, no. 1, pp. 11–17, 2015.

[13] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.

[14] R. Singh, A. Rani, *et al.*, "A survey on the generation of recommender systems.," *International Journal of Information Engineering & Electronic Business*, vol. 9, no. 3, 2017.

[15] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[16] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*, pp. 325–341, Springer, 2007.

[17] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[18] C. Sarkar, S. A. U. Nambi, R. V. Prasad, and A. Rahim, "A scalable distributed architecture towards unifying iot applications," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 508–513, IEEE, 2014.

[19] L. Yao, Q. Z. Sheng, A. H. Ngu, X. Li, and B. Benattalah, "Unveiling correlations via mining human-thing interactions in the web of things," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 5, pp. 1–25, 2017.

[20] L. Yao, Q. Z. Sheng, A. H. Ngu, and X. Li, "Things of interest recommendation by leveraging heterogeneous relations in the internet of things," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 2, p. 9, 2016.

[21] A. Gupta, R. Christie, and P. Manjula, "Scalability in internet of things: features, techniques and research challenges," *Int. J. Comput. Intell. Res*, vol. 13, no. 7, pp. 1617–1627, 2017.

[22] A. Heidari, M. A. Jabraeil Jamali, N. Jafari Navimipour, and S. Akbarpour, "Internet of things offloading: Ongoing issues, opportunities, and future challenges," *International Journal of Communication Systems*, vol. 33, no. 14, p. e4474, 2020.

[23] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the internet of things: A systematic review of the literature and recommendations for future research," *Journal of Network and Computer Applications*, vol. 97, pp. 23–34, 2017.

[24] Z. Ghanbari, N. J. Navimipour, M. Hosseinzadeh, and A. Darwesh, "Resource allocation mechanisms and approaches on the internet of things," *Cluster Computing*, vol. 22, no. 4, pp. 1253–1282, 2019.

[25] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "Iot middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2016.

[26] L. Yao, Q. Z. Sheng, A. H. Ngu, H. Ashman, and X. Li, "Exploring recommendations in internet of things," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 855–858, ACM, 2014.

[27] D. Massimo, M. Elahi, and F. Ricci, "Learning user preferences by observing user-items interactions in an iot augmented space," in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pp. 35–40, ACM, 2017.

[28] Y. Salman, A. Abu-Issa, I. Tumar, and Y. Hassouneh, "A proactive multi-type context-aware recommender system in the environment of internet of things," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 351–355, IEEE, 2015.

[29] M. S. Altulyan, C. Huang, L. Yao, X. Wang, S. Kanhere, and Y. Cao, "Reminder care system: An activity-aware cross-device recommendation system," in *International Conference on Advanced Data Mining and Applications*, pp. 207–220, Springer, 2019.

[30] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big iot data analytics: architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[31] Y. L. Sun and Y. Yang, "Trust establishment in distributed networks: Analysis and modeling," in *Communications, 2007. ICC'07. IEEE International Conference on*, pp. 1266–1273, IEEE, 2007.

[32] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the iot world: Present and future challenges," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2483–2495, 2018.

[33] S. Misra, M. Maheswaran, and S. Hashmi, *Security challenges and approaches in internet of things*. Springer, 2017.

[34] V. Scuotto, A. Ferraris, and S. Bresciani, "Internet of things: Applications and challenges in smart cities: a case study of ibm smart city projects," *Business Process Management Journal*, vol. 22, no. 2, pp. 357–367, 2016.

[35] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[36] M. Altulyan, L. Yao, X. Wang, C. Huang, S. S. Kanhere, and Q. Z. Sheng, "A survey on recommender systems for internet of things: Techniques, applications and future directions," *The Computer Journal*, 2021.

[37] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, and S. Zhang, "Recommendations on the internet of things: Requirements, challenges, and directions," *IEEE Internet Computing*, vol. 23, no. 3, pp. 46–54, 2019.

[38] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-based systems*, vol. 46, pp. 109–132, 2013.

[39] P. Nagarnaik and A. Thomas, "Survey on recommendation system methods," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pp. 1603–1608, IEEE, 2015.

[40] M. Sharma and S. Mann, "A survey of recommender systems: approaches and limitations," *International Journal of Innovations in Engineering and Technology*, vol. 2, no. 2, pp. 8–14, 2013.

[41] M. Sridevi, R. R. Rao, and M. V. Rao, "A survey on recommender system," *International Journal of Computer Science and Information Security*, vol. 14, no. 5, p. 265, 2016.

[42] Y. Dou, H. Yang, and X. Deng, "A survey of collaborative filtering algorithms for social recommender systems," in *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)*, pp. 40–46, IEEE, 2016.

[43] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[44] E. Ashley-Dejo, S. Ngwira, and T. Zuva, "A survey of context-aware recommender system and services," in *2015 International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–6, IEEE, 2015.

[45] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? a survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, 2019.

[46] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," 2013.

[47] B. Mobasher, R. Burke, and J. J. Sandvig, "Model-based collaborative filtering as a defense against profile injection attacks," in *AAAI*, vol. 6, p. 1388, 2006.

[48] S. Asiri and A. Miri, "An iot trust and reputation model based on recommender systems," in *Privacy, Security and Trust (PST), 2016 14th Annual Conference on*, pp. 561–568, IEEE, 2016.

[49] A. Salis, R. Bulla, G. Mancini, P. Cocco, and J. Jensen, "Anatomy of a fog-to-cloud distributed recommendation system in airports," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pp. 272–277, IEEE, 2018.

[50] S. Chakraverty and A. Mithal, "Iot based weather and location aware recommender system," in *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 636–643, IEEE, 2018.

[51] K. Gao, X. Yang, C. Wu, T. Qiao, X. Chen, M. Yang, and L. Chen, "Exploiting location-based context for poi recommendation when traveling to a new region," *IEEE Access*, vol. 8, pp. 52404–52412, 2020.

[52] S. D. Sawant, K. V. Sonawane, T. Jagani, and A. N. Chaudhari, "Representation of recommender system in iot using cyber physical techniques," in *Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of*, vol. 2, pp. 372–375, IEEE, 2017.

[53] J.-S. Lee and I.-Y. Ko, "Service recommendation for user groups in internet of things environments using member organization-based group similarity measures," in *Web Services (ICWS), 2016 IEEE International Conference on*, pp. 276–283, IEEE, 2016.

[54] I. Mashal, O. Alsaryrah, and T.-Y. Chung, "Performance evaluation of recommendation algorithms on internet of things services," *Physica A: Statistical Mechanics and its Applications*, vol. 451, pp. 646–656, 2016.

[55] I. Mashal, O. Alsaryrah, and T.-Y. Chung, "Testing and evaluating recommendation algorithms in internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 6, pp. 889–900, 2016.

[56] D. Zhang, Q. Zou, and H. Xiong, "Cruc: Cold-start recommendations using collaborative filtering in internet of things," 2013.

[57] N. S. Nizamkari, "A graph-based trust-enhanced recommender system for service selection in iot," in *Inventive Systems and Control (ICISC), 2017 International Conference on*, pp. 1–5, IEEE, 2017.

[58] F. Jabeen, M. Maqsood, M. A. Ghazanfar, F. Aadil, S. Khan, M. F. Khan, and I. Mehmood, "An iot based efficient hybrid recommender system for cardiovascular disease," *Peer-to-Peer Networking and Applications*, vol. 12, no. 5, pp. 1263–1276, 2019.

[59] W. Li, X. Zhou, S. Shimizu, M. Xin, J. Jiang, H. Gao, and Q. Jin, "Personalization recommendation algorithm based on trust correlation degree and matrix factorization," *IEEE Access*, vol. 7, pp. 45451–45459, 2019.

[60] D. Margaris and C. Vassilakis, "Exploiting internet of things information to enhance venues' recommendation accuracy," *Service Oriented Computing and Applications*, vol. 11, no. 4, pp. 393–409, 2017.

[61] S. Wu and F. Crestani, "Data fusion with estimated weights," in *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 648–651, Citeseer, 2002.

[62] X. Yang, B. Zimba, T. Qiao, K. Gao, and X. Chen, "Exploring iot location information to perform point of interest recommendation engine: Traveling to a new geographical region," *Sensors*, vol. 19, no. 5, p. 992, 2019.

[63] S. Rossi, F. Barile, D. Improta, and L. Russo, "Towards a collaborative filtering framework for recommendation in museums: from preference elicitation to group's visits," *Procedia Computer Science*, vol. 98, pp. 431–436, 2016.

[64] Z. Cui, X. Xu, F. Xue, X. Cai, Y. Cao, W. Zhang, and J. Chen, "Personalized recommendation system based on collaborative filtering for iot scenarios," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 685–695, 2020.

[65] S. P. Erdeniz, I. Maglogiannis, A. Menychtas, A. Felfernig, and T. N. T. Tran, "Recommender systems for iot enabled m-health applications," in *IFIP International conference on artificial intelligence applications and innovations*, pp. 227–237, Springer, 2018.

[66] A. Felfernig, S. Polat-Erdeniz, C. Uran, S. Reiterer, M. Atas, T. N. T. Tran, P. Azzoni, C. Kiraly, and K. Dolui, "An overview of recommender systems in the internet of things," *Journal of Intelligent Information Systems*, vol. 52, no. 2, pp. 285–309, 2019.

[67] N. Koubai and F. Bouyakoub, "Myrestaurant: A smart restaurant with a recommendation system," *International Journal of Computing and Digital Systems*, vol. 8, no. 02, pp. 143–156, 2019.

[68] B. Srisura and V. Avatchanakorn, "Periodical mobile recommendation toward parking conflict reduction," in *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, pp. 397–402, IEEE, 2019.

[69] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*, pp. 73–105, Springer, 2011.

[70] M. Tu, Y.-K. Chang, and Y.-T. Chen, "A context-aware recommender system framework for iot based interactive digital signage in urban space," in *Proceedings of the Second International Conference on IoT in Urban Space*, pp. 39–42, ACM, 2016.

[71] E. Kim, "User space customized recommendation service platform system in mobile edge environment," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 622–624, IEEE, 2018.

[72] K. G. HamlAbadi, A. M. Saghiri, M. Vahdati, M. D. TakhtFooladi, and M. R. Meybodi, "A framework for cognitive recommender systems in the internet of things (iot)," in *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 0971–0976, IEEE, 2017.

[73] A. M. Saghiri, M. Vahdati, K. Gholizadeh, M. R. Meybodi, M. Dehghan, and H. Rashidi, "A framework for cognitive internet of things based on blockchain," in *2018 4th International Conference on Web Research (ICWR)*, pp. 138–143, IEEE, 2018.

[74] R. Sharma and S. Rani, "A novel approach for smart-healthcare recommender system," in *International Conference on Advanced Machine Learning Technologies and Applications*, pp. 503–512, Springer, 2020.

[75] I. Hwang, M. Kim, and H. J. Ahn, "Data pipeline for generation and recommendation of the iot rules based on open text data," in *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on*, pp. 238–242, IEEE, 2016.

[76] S. Kumar, M. G. Kibria, S. Ali, M. A. Jarwar, and I. Chong, "Smart spaces recommending service provisioning in woo platform," in *Information and Communications (ICIC), 2017 International Conference on*, pp. 311–313, IEEE, 2017.

[77] A. Varfolomeyev, D. Korzun, A. Ivanovs, H. Soms, and O. Petrina, "Smart space based recommendation service for historical tourism," *Procedia Computer Science*, vol. 77, pp. 85–91, 2015.

[78] B. Di Martino, A. Esposito, S. Liguori, S. A. Maisto, and S. Nacchia, "A fuzzy prolog and ontology driven framework for medical diagnosis using iot devices," in *Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 875–884, Springer, 2017.

[79] A. Gyrard and A. Sheth, "Iamhappy: Towards an iot knowledge-based cross-domain well-being recommendation system for everyday happiness," *Smart Health*, vol. 15, p. 100083, 2020.

[80] F. Ali, S. R. Islam, D. Kwak, P. Khan, N. Ullah, S.-j. Yoo, and K. S. Kwak, "Type-2 fuzzy ontology–aided recommendation systems for iot–based healthcare," *Computer Communications*, vol. 119, pp. 138–155, 2018.

[81] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of library and information systems*, vol. 69, no. Supplement 32, pp. 175–186, 2000.

[82] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human–Computer Interaction*, vol. 16, no. 2-4, pp. 97–166, 2001.

[83] Q. Yuan, G. Cong, K. Zhao, Z. Ma, and A. Sun, "Who, where, when, and what: A nonparametric bayesian approach to context-aware recommendation and search for twitter users," *ACM Transactions on Information Systems (TOIS)*, vol. 33, no. 1, p. 2, 2015.

[84] A. Zhou, Y. Feng, P. Zhou, and J. Xu, "Social intimacy based iot services mining of massive data," in *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*, pp. 641–648, IEEE, 2017.

[85] P. D. Kaur *et al.*, "A context-aware recommender engine for smart kitchen," in *Smart Innovations in Communication and Computational Sciences*, pp. 161–170, Springer, 2019.

[86] F. Casino, C. Patsakis, E. Batista, O. Postolache, A. Martínez-Ballesté, and A. Solanas, "Smart healthcare in the iot era: A context-aware recommendation example," in *2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI)*, pp. 1–4, IEEE, 2018.

[87] M. Hong, J. J. Jung, F. Piccialli, and A. Chianese, "Social recommendation service for cultural heritage," *Personal and ubiquitous computing*, vol. 21, no. 2, pp. 191–201, 2017.

[88] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *international conference on Ubiquitous Computing*, pp. 116–122, Springer, 2001.

[89] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[90] Z. Chen, R. Ling, C.-M. Huang, and X. Zhu, "A scheme of access service recommendation for the social internet of things," *International Journal of Communication Systems*, vol. 29, no. 4, pp. 694–706, 2016.

[91] C. Ren, J. Chen, Y. Kuo, D. Wu, and M. Yang, "Recommender system for mobile users," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4133–4153, 2018.

[92] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[93] A. Forestiero, "Multi-agent recommendation system in internet of things," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 772–775, IEEE Press, 2017.

[94] S. Di Martino and S. Rossi, "An architecture for a mobility recommender system in smart cities," *Procedia Computer Science*, vol. 98, pp. 425–430, 2016.

[95] B. Twardowski and D. Ryzko, "Iot and context-aware mobile recommendations using multi-agent systems," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 33–40, IEEE, 2015.

[96] D. M. Jiménez-Bravo, J. Pérez-Marcos, D. H. De la Iglesia, G. Villarrubia González, and J. F. De Paz, "Multi-agent recommendation system for electrical energy optimization and cost saving in smart homes," *Energies*, vol. 12, no. 7, p. 1317, 2019.

[97] J. T. de Souza, G. A. L. de Campos, C. Rocha, E. Werbet, L. F. d. Costa, R. T. de Melo, and L. V. Alves, "An agent program in an iot system to recommend activities to minimize childhood obesity problems," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 654–661, 2020.

[98] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, p. 1, 2008.

[99] R. H. Güting, "Graphdb: Modeling and querying graphs in databases," in *VLDB*, vol. 94, pp. 12–15, 1994.

[100] G. Palaiokrassas, I. Karlis, A. Litke, V. Charlaftis, and T. Varvarigou, "An iot architecture for personalized recommendations over big data oriented applications," in *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*, vol. 2, pp. 475–480, IEEE, 2017.

[101] L. Noirie, M. Le Pallec, and N. Ammar, "Towards automated iot service recommendation," in *Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conference on*, pp. 103–106, IEEE, 2017.

[102] P. Wang, "Iot service recommendation scheme based on matter diffusion," *IEEE Access*, vol. 8, pp. 51500–51509, 2020.

[103] T. Mitchell, *Machine Learning, McGraw-Hill Higher Education.* 1997.

[104] I. El Naqa and M. J. Murphy, "What is machine learning?," in *Machine Learning in Radiation Oncology*, pp. 3–11, Springer, 2015.

[105] M. Sewak and S. Singh, "Iot and distributed machine learning powered optimal state recommender solution," in *Internet of Things and Applications (IOTA), International Conference on*, pp. 101–106, IEEE, 2016.

[106] Y. Guo, C. Yin, M. Li, X. Ren, and P. Liu, "Mobile e-commerce recommendation system based on multi-source information fusion for sustainable e-business," *Sustainability*, vol. 10, no. 1, p. 147, 2018.

[107] S. Asthana, A. Megahed, and R. Strong, "A recommendation system for proactive health monitoring using iot and wearable technologies," in *AI & Mobile Services (AIMS), 2017 IEEE International Conference on*, pp. 14–21, IEEE, 2017.

[108] K. Rasch, "An unsupervised recommender system for smart homes," *Journal of Ambient Intelligence and Smart Environments*, vol. 6, no. 1, pp. 21–37, 2014.

[109] S. Valtolina, M. Mesiti, and B. Barricelli, "User-centered recommendation services in internet of things era," in *CoPDA2014 workshop. Como, Italy*, 2014.

[110] S. R. Rizvi, S. Zehra, and S. Olariu, "Aspire: An agent-oriented smart parking recommendation system for smart cities," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 48–61, 2018.

[111] D. Ayata, Y. Yaslan, and M. E. Kamasak, "Emotion based music recommendation system using wearable physiological sensors," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 196–203, 2018.

[112] P. Mishra, S. K. Gudla, A. D. ShanBhag, and J. Bose, "Alternate action recommender system using recurrent patterns of smart home users," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, IEEE, 2020.

[113] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[114] S. H. Hashemi and J. Kamps, "Exploiting behavioral user models for point of interest recommendation in smart museums," *New Review of Hypermedia and Multimedia*, vol. 24, no. 3, pp. 228–261, 2018.

[115] Y. Han, Z. Han, J. Wu, Y. Yu, S. Gao, D. Hua, and A. Yang, "Artificial intelligence recommendation system of cancer rehabilitation scheme based on iot technology," *IEEE Access*, vol. 8, pp. 44924–44935, 2020.

[116] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

[117] R. Fakhfakh, A. B. Ammar, and C. B. Amar, "Deep learning-based recommendation: Current issues and challenges," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 12, pp. 59–68, 2017.

[118] D. Massimo, "User preference modeling and exploitation in iot scenarios," in *23rd International Conference on Intelligent User Interfaces*, pp. 675–676, ACM, 2018.

[119] N. Gutowski, T. Amghar, O. Camp, and S. Hammoudi, "A framework for context-aware service recommendation for mobile users: A focus on mobility in smart cities," 2017.

[120] R. O. Oyeleke, C.-Y. Yu, and C. K. Chang, "Situ-centric reinforcement learning for recommendation of tasks in activities of daily living in smart homes," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 317–322, IEEE, 2018.

[121] M. Altulyan, L. Yao, S. S. Kanhere, X. Wang, and C. Huang, "A unified framework for data integrity protection in people-centric smart cities," *Multimedia Tools and Applications*, vol. 79, no. 7, pp. 4989–5002, 2020.

[122] Z. Ji, I. Ganchev, M. O'Droma, L. Zhao, and X. Zhang, "A cloud-based car parking middleware for iot-based smart cities: Design and implementation," *Sensors*, vol. 14, no. 12, pp. 22372–22393, 2014.

[123] S. Cha, M. P. Ruiz, M. Wachowicz, L. H. Tran, H. Cao, and I. Maduako, "The role of an iot platform in the design of real-time recommender systems," in *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*, pp. 448–453, IEEE, 2016.

[124] C. Bin, T. Gu, Y. Sun, L. Chang, and L. Sun, "A travel route recommendation system based on smart phones and iot environment," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[125] R. Kamal, J. H. Lee, C. K. Hwang, S. I. Moon, and C. S. Hong, "Autonomic inferring of m2m-iot service-usage from user-emotion and environmental information," in *Proceedings of the Korea Information Science Society*, pp. 1034–1036, 2013.

[126] G. F. Savari, V. Krishnasamy, J. Sathik, Z. M. Ali, and S. H. A. Aleem, "Internet of things based real-time electric vehicle load forecasting and charging station recommendation," *ISA transactions*, vol. 97, pp. 431–447, 2020.

[127] V. Bhatnagar and R. Chandra, "Iot-based soil health monitoring and recommendation system," in *Internet of Things and Analytics for Agriculture, Volume 2*, pp. 1–21, Springer, 2020.

[128] M. Manca, L. Boratto, and S. Carta, "Design and architecture of a friend recommender system in the social bookmarking domain," in *2014 Science and Information Conference*, pp. 838–842, IEEE, 2014.

[129] A. Felfernig, S. P. Erdeniz, M. Jeran, A. Akcay, P. Azzoni, M. Maiero, and C. Doukas, "Recommendation technologies for iot edge devices," *Procedia Computer Science*, vol. 110, pp. 504–509, 2017.

[130] S. Haller and S. Hodges, "The need for a universal smartsensor network," 2002. Auto-ID Center, White Paper CAMAUTOID-WH-007.

[131] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of things: Survey and open issues of mqtt protocol," in *2017 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–6, IEEE, 2017.

[132] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014. Orlando: The Internet Engineering Task Force–IETF.

[133] N. Kushalnagar, G. Montenegro, C. Schumacher, *et al.*, "Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals," 2007.

[134] I. Fette, "The websocket protocol," 2011.

[135] C. C. Aggarwal, N. Ashish, and A. Sheth, "The internet of things: A survey from the data-centric perspective," in *Managing and mining sensor data*, pp. 383–428, Springer, 2013.

[136] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "A collaborative approach to in-place sensor calibration," in *Information Processing in Sensor Networks*, pp. 301–316, Springer, 2003.

[137] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 588–599, VLDB Endowment, 2004.

[138] C. C. Aggarwal, *Managing and mining sensor data*. Springer Science & Business Media, 2013.

[139] A. Gupta and M. Srivastava, "Developing auto-id solutions using sun java system rfid software," 2004. SUN Corporation.

[140] Q. Ni, A. García Hernando, and I. de la Cruz, "The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development," *Sensors*, vol. 15, no. 5, pp. 11312–11362, 2015.

[141] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[142] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing/IEEE Computer Society [and] IEEE Communications Society*, vol. 9, no. 1, p. 48, 2010.

[143] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A pattern mining approach to sensor-based human activity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1359–1372, 2010.

[144] C. Sutton, A. McCallum, *et al.*, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[145] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," in *International conference on Ubiquitous computing*, pp. 483–500, Springer, 2007.

[146] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *International conference on pervasive computing*, pp. 158–175, Springer, 2004.

[147] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2006.

[148] N. Yamada, K. Sakamoto, G. Kunito, Y. Isoda, K. Yamazaki, and S. Tanaka, "Applying ontology and probabilistic model to human activity recognition from surrounding things," *IPSJ Digital Courier*, vol. 3, pp. 506–517, 2007.

[149] J. Debattista, S. Scerri, I. Rivera, and S. Handschuh, "Ontology-based rules for recommender systems.," in *SeRSy*, pp. 49–60, 2012.

[150] Z. Li, C.-H. Chu, W. Yao, and R. A. Behr, "Ontology-driven event detection and indexing in smart spaces," in *2010 IEEE Fourth International Conference on Semantic Computing*, pp. 285–292, IEEE, 2010.

[151] Y. Bai, J. Yang, and Y. Qiu, "Ontocbr: Ontology-based cbr in context-aware applications," in *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, pp. 164–169, IEEE, 2008.

[152] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

[153] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[154] V. Oleshchuk, "Internet of things and privacy preserving technologies," in *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, pp. 336–340, IEEE, 2009.

[155] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "Iot security: ongoing challenges and research opportunities," in *2014 IEEE 7th international conference on service-oriented computing and applications*, pp. 230–234, IEEE, 2014.

[156] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[157] S. Görmüş, H. Aydın, and G. Ulutaş, "Security for the internet of things: a survey of existing mechanisms, protocols and open research issues," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 33, no. 4, pp. 1247–1272, 2018.

[158] R. Mitchell and R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, 2014.

[159] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *International conference on wireless algorithms, systems, and applications*, pp. 685–695, Springer, 2015.

[160] L. Liu, Y. Peng, M. Liu, and Z. Huang, "Sensor-based human activity recognition system with a multilayered model using time series shapelets," *Knowledge-Based Systems*, vol. 90, pp. 138–152, 2015.

[161] Z. Yan, D. Chakraborty, A. Misra, H. Jeung, and K. Aberer, "Sammple: Detecting semantic indoor activities in practical settings using locomotive signatures," in *2012 16th International Symposium on Wearable Computers*, pp. 37–40, Ieee, 2012.

[162] Z. Yan, D. Chakraborty, S. Mittal, A. Misra, and K. Aberer, "An exploration with online complex activity recognition using cellphone accelerometer," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 199–202, 2013.

[163] L. Peng, L. Chen, M. Wu, and G. Chen, "Complex activity recognition using acceleration, vital sign, and location data," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1488–1498, 2018.

[164] L. Peng, L. Chen, X. Wu, H. Guo, and G. Chen, "Hierarchical complex activity representation and recognition using topic model and classifier level fusion," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 6, pp. 1369–1379, 2016.

[165] L. Yao, Q. Z. Sheng, B. Benatallah, S. Dustdar, X. Wang, A. Shemshadi, and S. S. Kanhere, "Wits: an iot-endowed computational framework for activity recognition in personalized smart homes," *Computing*, vol. 100, no. 4, pp. 369–385, 2018.

[166] F. Amirjavid, A. Bouzouane, and B. Bouchard, "Intelligent temporal data driven world actuation in ambient environments: Case study: Anomaly recognition and assistance provision in smart home," in *Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on*, pp. 287–293, IEEE, 2013.

[167] K.-Y. Lam, N. W.-H. Tsang, S. Han, J. K.-Y. Ng, S.-W. Tam, and A. Nath, "Smartmind: Activity tracking and monitoring for patients with alzheimer's disease," in *Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on*, pp. 453–460, IEEE, 2015.

[168] F. Latfi, B. Lefebvre, and C. Descheneaux, "Ontology-based management of the telehealth smart home, dedicated to elderly in loss of cognitive autonomy.," in *OWLED*, vol. 258, 2007.

[169] K. A. Alaghbari, M. H. M. Saad, A. Hussain, and M. R. Alam, "Activities recognition, anomaly detection and next activity prediction based on neural networks in smart homes," *IEEE Access*, vol. 10, pp. 28219–28232, 2022.

[170] L. G. Fahad and S. F. Tahir, "Activity recognition and anomaly detection in smart homes," *Neurocomputing*, vol. 423, pp. 362–372, 2021.

[171] I. U. Khan, S. Afzal, and J. W. Lee, "Human activity recognition via hybrid deep learning based model," *Sensors*, vol. 22, no. 1, p. 323, 2022.

[172] A. Hayat, M. Dias, B. P. Bhuyan, and R. Tomar, "Human activity recognition for elderly people using machine and deep learning approaches," *Information*, vol. 13, no. 6, p. 275, 2022.

[173] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.

[174] L. Liu, L. Cheng, Y. Liu, Y. Jia, and D. S. Rosenblum, "Recognizing complex activities by a probabilistic interval-based model.," in *AAAI*, vol. 30, pp. 1266–1272, 2016.

[175] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, p. 40, ACM, 2012.

[176] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[177] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[178] B. Das, D. J. Cook, M. Schmitter-Edgecombe, and A. M. Seelye, "Puck: an automated prompting system for smart environments: toward achieving automated prompting–challenges involved," *Personal and ubiquitous computing*, vol. 16, no. 7, pp. 859–873, 2012.

[179] Y. Sahaf, *Comparing sensor modalities for activity recognition.* PhD thesis, Washington State University, 2011.

[180] M. S. Altulyan, C. Huang, L. Yao, X. Wang, and S. Kanhere, "Deep reinforcement learning for dynamic things of interest recommendation in intelligent ambient environment," in *Australasian Joint Conference on Artificial Intelligence*, pp. 393–404, Springer, 2022.

[181] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.

[182] L. M. De Campos, Fernández-Luna, and et al., "Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks," *International journal of approximate reasoning*, vol. 51, no. 7, pp. 785–799, 2010.

[183] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for page-wise recommendations," in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 95–103, ACM, 2018.

[184] X. Zhao, L. Zhang, Z. Ding, L. Xia, J. Tang, and D. Yin, "Recommendations with negative feedback via pairwise deep reinforcement learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1040–1048, ACM, 2018.

[185] G. Zheng, F. Zhang, Zheng, and et al., "Drn: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176, 2018.

[186] C. Huang and L. Yao, "Active object estimation for human-robot collaborative tasks," in *International Conference on Neural Information Processing*, pp. 750–761, Springer, 2020.

[187] I. Munemasa, Y. Tomomatsu, K. Hayashi, and T. Takagi, "Deep reinforcement learning for recommender systems," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, pp. 226–233, IEEE, 2018.

[188] Z. Zhao, X. Chen, Z. Xu, and L. Cao, "Tag-aware recommender system based on deep reinforcement learning," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[189] X. Zhao, C. Gu, H. Zhang, X. Yang, X. Liu, H. Liu, and J. Tang, "Dear: Deep reinforcement learning for online advertising impression in recommender systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 750–758, 2021.

[190] S. H. Oh, S. J. Lee, and J. Park, "Precision medicine for hypertension patients with type 2 diabetes via reinforcement learning," *Journal of Personalized Medicine*, vol. 12, no. 1, p. 87, 2022.

[191] S. S. Shuvo and Y. Yilmaz, "Home energy recommendation system (hers): A deep reinforcement learning method based on residents' feedback and activity," *IEEE Transactions on Smart Grid*, 2022.

[192] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[193] H. Alemdar, H. Ertan, O. D. Incel, and C. Ersoy, "Aras human activity datasets in multiple homes with multiple residents," in *7th IEEE International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, pp. 232–235, IEEE, 2013.

[194] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods of information in medicine*, vol. 48, no. 05, pp. 480–485, 2009.

[195] M. S. Altulyan, C. Huang, L. Yao, X. Wang, and S. S. Kanhere, "Contextual bandit learning for activity-aware things-of-interest recommendation in an assisted living environment.," in *ADC*, pp. 37–49, 2021.

[196] M. S. Altulayan, C. Huang, L. Yao, X. Wang, and S. Kanhere, "Contextual bandit learning for activity-aware things-of-interest recommendation in an assisted living environment," in *Databases Theory and Applications: 32nd Australasian Database Conference, ADC 2021, Dunedin, New Zealand, January 29–February 5, 2021, Proceedings 32*, pp. 37–49, Springer International Publishing, 2021.

[197] M. Altulyan, L. Yao, C. Huang, X. Wang, and S. S. Kanhere, "Context-induced activity monitoring for on-demand things-of-interest recommendation in an ambient intelligent environment," *Future Internet*, vol. 13, no. 12, p. 305, 2021.

[198] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, and S. Zhang, "Recommendations on the internet of things: Requirements, challenges, and directions," *IEEE Internet Computing*, vol. 23, no. 3, pp. 46–54, 2019.

[199] "Oasis standard, mqtt version 3.1.1 available from <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>." accessed 2019-07-16.

[200] Q. A. Ahmed and A. Q. Al-Neami, "A smart biomedical assisted system for alzheimer patients," in *IOP Conference Series: Materials Science and Engineering*, p. 012110, IOP Publishing, 2020.

[201] N. Armstrong, C. Nugent, G. Moore, and D. Finlay, "Developing smartphone applications for people with alzheimer's disease," in *The 10th International Conference on Information Technology and Applications in Biomedicine*, pp. 1–5, IEEE, 2010.

[202] L. Choon, "Helper system for managing alzheimer's people using mobile application," *Universiti Malaysia Pahang*, 2015.

[203] S. Alharbi, A. Altamimi, F. Al-Qahtani, B. Aljofi, M. Alsmadi, M. Alshabanah, D. Alrajhi, and I. Almarashdeh, "Analyzing and implementing a mobile reminder system for alzheimer's patients," *ALHARBI, S., ALTAMIMI, A., AL-QAHTANI, F., ALJOFI, B., ALSMADI, MK, ALSHABANAH, M., ALRAJHI, D. & ALMARASHDEH, I*, pp. 444–454, 2019.

[204] S. Aljehani, R. Alhazmi, S. Aloufi, B. Aljehani, and R. Abdulrahman, "icare: Applying iot technology for monitoring alzheimer's patients," in *1st International Conference on Computer Applications & Information Security*, IEEE, 2018.

[205] B. Steenwinckel, F. De Backere, J. Nelis, F. Ongenae, and F. De Turck, "Self-learning algorithms for the personalised interaction with people with dementia," in *Workshops of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 153–158, 2018.

[206] W. Intayoad, C. Kamyod, and P. Temdee, "Reinforcement learning based on contextual bandits for personalized online learning recommendation systems," *Wireless Personal Communications*, pp. 1–16, 2020.

[207] C. Zhang, H. Wang, S. Yang, and Y. Gao, "A contextual bandit approach to personalized online recommendation via sparse interactions," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 394–406, Springer, 2019.

[208] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

[209] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[210] J. Friedman, T. Hastie, R. Tibshirani, *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[211] D. Cortes, "Adapting multi-armed bandits policies to contextual bandits scenarios," *arXiv preprint arXiv:1811.04383*, 2018.

[212] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, pp. 1044–1048, 2009.

[213] A. H. Seh, M. Zarour, M. Alenezi, A. K. Sarkar, A. Agrawal, R. Kumar, and R. Ahmad Khan, "Healthcare data breaches: insights and implications," in *Healthcare*, vol. 8, p. 133, Multidisciplinary Digital Publishing Institute, 2020.

[214] "Bitcoin: A peer-to-peer electronic cash system.." accessed 2019-01-07.

[215] I. Zikratov, A. Kuzmin, V. Akimenko, V. Niculichev, and L. Yalansky, "Ensuring data integrity using blockchain technology," in *2017 20th Conference of Open Innovations Association (FRUCT)*, pp. 534–539, IEEE, 2017.

[216] L. Hang and D.-H. Kim, "Design and implementation of an integrated iot blockchain platform for sensing data integrity," *Sensors*, vol. 19, no. 10, p. 2228, 2019.

[217] T. Dey, S. Jaiswal, S. Sunderkrishnan, and N. Katre, "Healthsense: A medical use case of internet of things and blockchain," in *2017 International conference on intelligent sustainable systems (ICISS)*, pp. 486–491, IEEE, 2017.

[218] D. Ichikawa, M. Kashiyama, and T. Ueno, "Tamper-resistant mobile health using blockchain technology," *JMIR mHealth and uHealth*, vol. 5, no. 7, p. e7938, 2017.

[219] J. Zhang, N. Xue, and X. Huang, "A secure system for pervasive social network-based healthcare," *Ieee Access*, vol. 4, pp. 9239–9250, 2016.

[220] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Continuous patient monitoring with a patient centric agent: A block architecture," *IEEE Access*, vol. 6, pp. 32700–32726, 2018.

[221] Y. Ji, J. Zhang, J. Ma, C. Yang, and X. Yao, "Bmpls: Blockchain-based multi-level privacy-preserving location sharing scheme for telecare medical information systems," *Journal of medical systems*, vol. 42, no. 8, pp. 1–13, 2018.

[222] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, pp. 1–7, 2018.

[223] H. Kaur, M. A. Alam, R. Jameel, A. K. Mourya, and V. Chang, "A proposed solution and future direction for blockchain-based heterogeneous medicare data in cloud environment," *Journal of medical systems*, vol. 42, no. 8, pp. 1–11, 2018.

[224] M. S. Ali, M. Vecchio, G. D. Putra, S. S. Kanhere, and F. Antonelli, "A decentralized peer-to-peer remote health monitoring system," *Sensors*, vol. 20, no. 6, p. 1656, 2020.

[225] J. Ren, J. Li, H. Liu, and T. Qin, "Task offloading strategy with emergency handling and blockchain security in sdn-empowered and fog-assisted healthcare iot," *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 760–776, 2021.

[226] B. Alamri, I. T. Javed, and T. Margaria, "A gdpr-compliant framework for iot-based personal health records using blockchain," in *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, IEEE, 2021.

[227] K. Abbas, M. Afaq, T. Ahmed Khan, and W.-C. Song, "A blockchain and machine learning-based drug supply chain management and recommendation system for smart pharmaceutical industry," *Electronics*, vol. 9, no. 5, p. 852, 2020.

[228] R. Bosri, M. S. Rahman, M. Z. A. Bhuiyan, and A. Al Omar, "Integrating blockchain with artificial intelligence for privacy-preserving in recommender systems," *IEEE Transactions on Network Science and Engineering*, 2020.

[229] Y. M. Arif, H. Nurhayati, F. Kurniawan, S. M. S. Nugroho, and M. Hariadi, "Blockchain-based data sharing for decentralized tourism destinations recommendation system," *International Journal of Intelligent Engineering & System*, vol. 13, no. 6, pp. 472–486, 2020.

[230] B. Yan, J. Yu, Y. Wang, Q. Guo, B. Chai, and S. Liu, "Blockchain-based service recommendation supporting data sharing," in *International Conference on Wireless Algorithms, Systems, and Applications*, pp. 580–589, Springer, 2020.

[231] T.-Y. Yeh and R. Kashef, "Trust-based collaborative filtering recommendation systems on the blockchain," *Advances in Internet of Things*, vol. 10, no. 4, pp. 37–56, 2020.

[232] M. Arora, A. B. Chopra, and V. S. Dixit, "An approach to secure collaborative recommender system using artificial intelligence, deep learning, and blockchain," in *Intelligent communication, control and devices*, pp. 483–495, Springer, 2020.

[233] L. Lin, Y. Tian, and Y. Liu, "A blockchain-based privacy-preserving recommendation mechanism," in *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*, pp. 74–78, IEEE, 2021.

[234] D. Bobadilla and C. Lipizzi, "A blockchain-based collaborative filtering recommendation system based on trust," in *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 121–126, IEEE, 2021.

[235] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *International Conference on Financial Cryptography and Data Security*, pp. 507–527, Springer, 2015.

[236] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.

[237] M. Gord, "Smart contracts described by nick szabo 20 years ago now becoming reality," *Bitcoin Magazine*, 2016.

[238] R. M. Parizi, A. Dehghantanha, *et al.*, "Smart contract programming languages on blockchains: An empirical evaluation of usability and security," in *International Conference on Blockchain*, pp. 75–91, Springer, 2018.

[239] "Solidity." `https://solidity.readthedocs.io/en/develop/`. Accessed:2021-11-13.

[240] "kovan-testnet." https://kovan-testnet.github.io/website/, 2017. [Online; accessed 15-December-2021].

[241] "Rinkeby." https://www.rinkeby.io/#stats, 2017. [Online; accessed 15-December-2021].

[242] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional lstm networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[243] H. Nair, C. Tan, M. Zeng, O. J. Mengshoel, and J. P. Shen, "Attrinet: Learning mid-level features for human activity recognition with deep belief networks," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pp. 510–517, 2019.

[244] Z. N. Khan and J. Ahmad, "Attention induced multi-head convolutional neural network for human activity recognition," *Applied Soft Computing*, vol. 110, p. 107671, 2021.

[245] X. Zhang, L. Yao, D. Zhang, X. Wang, Q. Z. Sheng, and T. Gu, "Multi-person brain activity recognition via comprehensive eeg signal analysis," in *Proceedings of the 14th EAI international conference on mobile and ubiquitous systems: computing, networking and services*, pp. 28–37, 2017.

[246] X. Zhang, L. Yao, Q. Z. Sheng, S. S. Kanhere, T. Gu, and D. Zhang, "Converting your thoughts to texts: Enabling brain typing via deep feature learning of eeg signals," in *2018 IEEE international conference on pervasive computing and communications (PerCom)*, pp. 1–10, IEEE, 2018.