Office of Graduate Programs
School of Engineering and Applied Science
Thornton Hall A108
20200609

Thesis/Dissertation Cover and Approval Pages

* Complete the following Form Variables and then click UPDATE

Form Variables

Document Title	Adaptive Mobile Sensing Using Reinforcement Learning Framework
Student Name	Lihua Cai
Document Type	Dissertation
Degree	
Semester/Year Degree Awarded	December 2020
Advisor 1	Laura Barnes
Advisor 2	Mehdi Boukhechba
Committee Member 1	Peter Beling
Committee Member 2	James Lambert
Committee Member 3	Philip Chow
Committee Member 4	
Committee Member 5	
Committee Member 6	

UPDATE

A

Doctoral Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Lihua Cai

August, 2020

APPROVAL SHEET

This

doctoral dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Lihua Cai

Advisor: Laura Barnes

Advisor: Mehdi Boukhechba

Committee Member: Peter Beling

Committee Member: James Lambert

Committee Member: Philip Chow

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

August, 2020

Doctoral Dissertation

Adaptive Mobile Sensing Using Reinforcement Learning Framework

Author:

Lihua Cai

Advisor:

Dr. Laura Barnes

A dissertation submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

in the

Department of Engineering Systems and Environment

University of Virginia

February, 2020

Dissertation Committee Members

- Dr. Peter Beling (Committee Chair, Engineering Systems and Environment)
- Dr. Laura Barnes (Advisor, Engineering Systems and Environment)
- Dr. Mehdi Boukhechba (Co-advisor, Engineering Systems and Environment)
- Dr. James Lambert (Committee Member, Engineering Systems and Environment)
- Dr. Philip Chow (Committee Member, Psychiatry and Neurobehavioral Sciences)

Abstract

Mobile sensing has created unprecedented opportunities to study human behaviors and serve users in diverse applications. The foundation to mobile sensing is data collection through both passive and active sensing. Successful mobile sensing applications require efficiently managing energy consumption in passive sensing using smartphone embedded sensors, and compliance in active sensing such as mobile Ecological Momentary Assessments (EMAs). To date, there is a lack of a unified framework that can enable adaptive mobile sensing in a personalized and adaptive manner to address both of these challenges. This dissertation leverages the most recognizable general purpose artificial intelligence framework, reinforcement learning (RL), to model both passive and active sensing as sequential control problems, and adapt the sensing tasks to the users' contexts. We design both adaptive passive and active sensing strategies under the RL framework with different problem formulations to improve energy efficiency in passive sensing, and user compliance in active sensing. Performance of the proposed RL strategies are evaluated in simulations using real data collected by continuous mobile sensing in mental health studies. Results from simulations and predictive models show that our approaches, when compared to various baseline methods, consistently achieve: 1) for passive sensing, more energy saving with comparable data utility; and 2) for active sensing, higher overall compliance. We implement and maintain a cross-OS mobile adaptive sensing platform, on which the proposed RL strategies will be evaluated in future studies, and point out future directions to advance mobile sensing technologies.

Acknowledgements

There are too many people to thank during the past 6 years at Virginia. My gratitude goes first to Dr. Matthew Gerber. He brought me to Virginia in fall 2014 and provided me funding until he chose to leave academia for industry last summer. During those years, he guided me to develop my research in adaptive active sensing, worked together with me for my qualification exam and dissertation proposal. In spring 2015, when I was forced to take a family leave, he provided all the supports he could. And when I was ready to come back, he did not hesitate to resume my funding. I learned from him how to conduct rigorous research, how to revise manuscript, and how to develop research questions that are significant in the mobile sensing area.

I am equally thankful to Dr. Mehdi Boukhechba for his mentoring. In my early years here, we worked a lot together on papers, and research ideas. He was always available when I needed research guidance or had doubts. I enjoy the most his flexibility in time and his generosity in mentoring me especially when I had no research experience at the start of my Ph.D. I still remember vividly our time playing ping-pong together and during those times our conversations about life in general and plan for future. Mehdi also helped me develop my research in adaptive passive sensing in this dissertation. I hope to continue to work with Mehdi and keep growing under his mentorship.

Dr. Laura Barnes became my academic advisor last summer and funded me for the rest of my time at Virginia. The prolonged process of Ph.D. research had planted a seed of depression in me. When I first knew that Dr. Gerber was going to leave Virginia, I was extremely stressed, depressed, and at a lost. It was Dr. Barnes who warmly welcomed me into her lab and reassured me not to worry about funding and time as my sixth year was approaching. She provided me all the accommodations and flexibility I needed in research. She was able to listen to me, and guide me to make important decisions on research and my dissertation. I can not express enough my gratitude to her. And in fact, I enjoy working with her so much that I decide to stay and continue a post-doc position under her supervision.

I also want to extend my gratitude towards all my colleagues in the previous PTL (Lab) with Dr. Gerber, including Mohammad Al Boni, Peter Wu, Mark Rucker, Kiana Jafari Meimandi, Nikesh Kapadia, and at the time visiting Ph.D. student Ristea Mihaela Alina for all the time we spent together hiking, dinning, discussing, and working together on research and papers. In the past one and a half year, Anna Baglione, Sanjana Mendu, Mawulolo Ameko, Sonia Baee, Navreet Kaur, Debajyoti Datta (Debo) and many others from the Barnes Health and Sensing Lab provided me the nutrients to continue thriving. And most importantly, their constant supports and encouragements rescued me away from the verge of depression, ironically a symptom that we have all been so familiar with as our research topic in mental health.

Lastly, I want to thank my committee for their time, supports, and flexibility for each Ph.D. milestone I accomplished along this journey. I want to thank Dr. Greg Gerling for his human factor courses which taught me how to create figures for my research, Dr. James Lambert for his advice and enthusiasm on my research, Dr. Peter Beling for his expertise in reinforcement learning, and Dr. Philip Chow for his time mentoring me on our collaborations in the EIM project.

There are many friends that I want to make special shout-outs to: Junrui Xu, Xiaomin Lin, Yang Wang, who shared their experiences from their doctorate studies at Virginia; Kai Dong knew me since day one at Virginia, and has since become great friend; Wanjia Zhao, Minghui Sun and other roommates over the years living together at Eagles Landing; Yong Lin, Zhijie Wang, Simeng Wu, and Yinhui Li, who are like my big brothers/sisters and always think of me and support me to carry on. This list goes on and can not be inclusive.

I leave this special place to my wife Grace Yuying Ma, my daughter Virginia Yuanyuan Cai, my parents, and my two sisters. Family is my lifeblood and greatest strength to carry on and fight. Without all of you, life becomes meaningless. You all have been waiting for this day to come in the past six years, and I can not say enough but hope that you can share my joy in accomplishing this great achievement in life. Hope to have many more such moments in the coming future with all of you.

To my beloved wife Grace Ma, daughter Virginia Cai.

Contents

List of Figures

List of Tables

List of Abbreviations

CHS	**C**yber **H**uman **S**ystems
EMA	**E**cological **Momentary** **A**ssessment
MCS	**M**obile **C**rowd **S**ensing
RL	**R**einforcement **L**earning
JITAI	**J**ust-**I**n-**T**ime **A**daptive **I**ntervention
HAR	**H**uman **A**ctivity **R**ecognition
MDP	**M**arkov **D**ecision **P**rocess
TD	**T**emporal **D**ifference
DP	**D**ynamic **P**rogramming
SVM	**S**upport **V**ector **M**achine
SMOTE	**S**ynthetic **M**inority **O**versampling **T**echniqu**E**
QLADE	**Q**-learning with **L**inear **A**pproximation and **D**ecaying **E**xploration
FFT	**Fast** **F**ourier **T**ransformation
DWT	**D**iscrete **W**avelet **T**ransformation
DBSCAN	**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
OPTICS	**O**rdering **P**oints **T**o **I**dentify **C**luster **S**tructure
PCA	**P**rinciple **C**omponent **A**nalysis
LDA	**L**atent **D**irichlet **A**llocation
ECDF	**E**mpirical **C**umulative **D**istribution **F**unction
LASSO	**L**east **A**bsolute **S**hrinkage and **S**election **O**perator
HAR	**H**uman **A**ctivity **R**ecognition
CNN	**C**onvolution **N**eural **N**etwork
HMM	**H**idden **M**arkov **M**odel

Chapter 1

Introduction

1.1 Motivation

Cyber Human Systems (CHS) is a broad area of research and development that aims to enhance human capabilities and foster well being by optimizing the human-technology interface.[1] Smartphone has become the most prominent CHS in the past decade due to their increasing accessibility (e.g. affordability and mass production), and improved hardware (e.g., computing power and rich embedded sensors), and the wider coverage of wireless networks. These advancements create pervasive opportunities to both passively and actively sense human contexts, understand users' contextual states in real time, and based on this understanding, provide timely, automatic, and relevant services to them. For instance, the application of mobile sensing to improve public access to scarce health care resources led to the development of mobile health [200]. Psychologists and behavior scientists leverage mobile sensing to study individual and social behaviors [229, 208], while environmentalists apply mobile sensing to understand large scale complex systems such as air pollution, traffic management, and energy consumption in smart city [70, 41].

Successes in mobile sensing applications rely heavily on both passive and active sensing management. Passive sensing refers to continuous sensing using smartphone embedded sensors to collect various data streams for construction and mining of user's contextual states, while requires no active user involvements during the data collection process. [200] The major challenge in passive sensing lies in the limited energy in smartphones due to

[1] https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=504958

the small battery capacity. Continuous sensing without interruption will lead to quick drainage of mobile device's battery, interfering with user's normal usage of the device for other, potentially more important, daily purposes. [112, 144]

To complement passive sensing, active sensing (e.g., mobile ecological momentary assessment or EMA) becomes a necessity when it is impossible to uncover key determinants of human performance and well being through passively sensed data alone. [200] In contrast to passive sensing, active sensing requires users to expend resources (e.g., time, energy, and cognitive efforts) to satisfy sensing demands (e.g. taking surveys through smartphones). Typical mobile sensing users and participants will have limited time and mental resources to comply with active sensing tasks, especially when demands are placed in inopportune moments such as during exercising, in meetings, or when being stressful. This results in low compliance, and therefore absence of critical information required in the targeted application. For instance, the authors in [201] reported an overall mean compliance rate 68.2%($SD = 16.9\%$) in mobile EMAs for a one week study in 461 adolescent smokers. In [217], a four week mobile EMA study reported a 30% overall compliance rate from 17 college students.

To date, there is no unified framework that aims to systematically solve both energy efficiency challenge in passive sensing, and compliance challenge in active sensing. In this dissertation, we leverage the reinforcement learning (RL) framework to address these fundamental challenges in both passive and active sensing within mobile sensing applications. We aim to construct key state factors that can capture the underpinning differences in each sensing task, and effective feedback signals that can shape the learned policies towards improving the ultimate sensing outcomes (e.g., higher energy efficiency in passive sensing, and response compliance in active sensing).

Problem Statement

Passive Sensing: Continuous operation of embedded sensors poses tremendous energy stress on personal smartphones; however, reducing sampling time through simple techniques such as duty cycling may lead to compromised data utility (i.e., missing data when critical information is needed for mobile sensing applications). This dilemma in balancing energy efficiency and data utility leads to our first research question:

RQ1: ***How do we optimize the timing of sensor deployment in order to maintain a practical balance between energy efficiency and data utility?***

Active Sensing: Active sensing provides critical information about mobile sensing users that can not be passively sensed through smartphone embedded sensors (e.g., GPS, accelerometer, and microphone) for a wide range of applications (e.g., Just-in-time Adaptive Interventions and environmental monitoring). Unfortunately, active sensing requires frequent and oftentimes burdensome user responses that can lead to high perceived burden and noncompliance. One source of low response compliance and poor user experience comes from active sensing in the wrong moments of user's routine lives when they are unavailable because of engagement with other more prioritized tasks (e.g. during a class). In particular, users may forget to follow up with these untimely active sensing triggers or can be interrupted from their current tasks. In order to reduce the proportion of these mistimed active sensing tasks and upkeep an acceptable level of response compliance as well as providing a good user experience, trigger timing of active sensing needs to be adaptive to the users' changing contexts. This leads to our second research question:

RQ2: ***How do we optimize the timing and contexts of active sensing triggers in order to maximize long term response compliance?***

To address these two research questions, we propose to leverage the RL framework, which can unify both passive and active sensing under the same adaptive sensing framework.

1.2 Applying Reinforcement Learning in Adaptive Sensing

Benefits of Applying RL in Adaptive Sensing

Implementing adaptive sensing using the RL framework has numerous benefits. The most prominent one roots in its underlying principle 'trial and error'. In many real world settings such as the ones we are facing in mobile sensing, we do not have prior knowledge on how to optimally manage both passive and active sensing with respect to energy efficiency and user compliance. Learning through balancing exploitation and exploration can lead to best long term results.

Another important reason of selecting the RL framework as our approach is due to the unpredictability in human behaviors. Although existing research has discovered a high level routine pattern in human daily lives [77], people's routine patterns evolve over time. To capture these within subject variability, continuous learning is necessary to keep up with the changes in human behaviors. In addition, between subject variability requires learning individual models for different subjects. The RL framework provides us both of these conveniences to model human behavior dynamics in an adaptive and personalized manner.

The RL framework is a natural fit to transform static mobile sensing into adaptive sensing by converting the raw sensing data into critical contexts that can better inform sensing decision. The Markov Decision Process (MDP) formulation enables this context-aware through state characterization, treats sensing decision as action selection, and develops a constantly updated policy via a carefully designed feedback (reward) signal. Under this formulation, both adaptive passive and active sensing can be unified using the RL framework. Hosts of mobile sensing applications need only to design the state characterization, decide the action space, and define the reward signal. These decisions will then influence what RL algorithms to choose for learning a policy.

The idea of RL can be dated back to the mid-19th century in animal behavior studies.

Since then, RL has undergone over a century of developments in several different threads including trial and error and optimal control [203]. Modern RL has become tightly integrated with statistics, optimization, and other mathematical subjects. Many RL research has focused on topics related to efficient learning [241] and delayed rewards [57]. All these more recent developments will provide a solid methodological foundation to our topics in this dissertation, that is adaptive passive and active sensing.

Challenges of Applying RL in Adaptive Sensing

Applying RL in applications involved human behaviors and their situated contexts as the interacted environment warrants several technical challenges. First and foremost, unlike many physical (e.g., balancing an inverse pendulum) and virtual (e.g., adversarial games) problems, learning effective state representation of human behaviors and its surrounding environment is extremely challenging. There is no definitive models that are both sufficiently specialized and flexible to accommodate the between and within subject variability in people's behaviors. Regarding passive sensing, since our goal is to enhance energy efficiency by reducing sensor deployments at moments that the users are not active, we can characterize the state using features that capture signal variance in the collected raw data stream. However, given the infinite many ways of measuring signal variance, we do not yet know what the best practice is. Regarding active sensing, we need to characterize users' interruptibility and availability in the state representation. Numerous researches in interruption management and human computer interaction [164, 146] find that some contextual factors such as current task [165] and psychological traits [147] are significantly correlated with users' response compliance. These results provide initial guidance to our design of state representation for understanding users' active sensing response compliance. However, no existing studies have applied them in the active sensing setting as we do in this dissertation.

The second challenge arises from the unpredictability of human behaviors. It is easy to understand the state of a chess game, but rather difficult to understand the state of a human user. The changes of behaviors and thoughts in a person is not solely determined by the applied action (e.g., active sensing demands) but also their exposure to the surrounded social and physical environments, which is unpredictable and beyond our control. Because of this unpredictability, human environment is dynamic and non-stationary. Although environmental model is not a required component for solving a RL problem, as we do not know how human user will behave, the learned policy do not have theoretical convergence guarantee. Even worse is that if a user's behaviors and environment have weaker routine patterns or are changing faster than the agent can adapt to, the learned policy may result in sub-optimal long term rewards.

Limited learning samples is another significant challenge in most human behavior problems. Data collection in human studies are complicated and costly. Thus we have to improve sample efficiency using various RL techniques given our limited data. However, it is not clear what the best way to achieve higher sample efficiency is. Possible solutions include resampling, and integrating planning with learning. Meanwhile, we have delayed or unknown immediate rewards due to choosing the 'placebo' action. In the context of adaptive sensing, when we choose not to deploy a certain sensor or trigger an active sensing task, we either do not have the required data to compute the reward for passive sensing, or do not know the actual response if the sensing task is instead triggered for active sensing.

Last but not least, unlike supervised learning, in which ground truth knowledge from a supervisor is available for guiding learning, we have to design a reward signal to guide our learning in a RL setting towards achieving the ultimate goal of the targeted application. In the case of adaptive sensing, we want to improve energy efficiency and response compliance in passive and active sensing, respectively. What are the guidelines to design proximal reward signals that can shape the policy towards long term improvements in energy efficiency and response compliance remains research question to be answered.

1.3 Overview of the Dissertation

The rest of the dissertation is organized as the following: Chapter 2 provides a review of embedded hardware sensors in today's consumer smartphones, research in mobile sensing applications, and a brief summary about existing challenges in mobile sensing. A brief introduction to RL and technical details on several RL topics are then presented.

In Chapter 3, we propose a feature extraction framework based on reviews in existing mobile sensing works. This proposed framework will guide state feature design in the RL adaptive sensing strategies in the subsequent chapters.

Chapter 4 and 5 present our adaptive passive sensing framework. We develop adaptive sensing algorithms using the RL framework, present our design methods on state representation and reward, and how we handle the delayed reward challenge. We show two different problem formulations to turn continuous sensing into adaptive sensing. Simulation results imply that energy efficiency can be improved without adversely affecting the data utility in predicting daily negative affect and social anxiety.

Chapter 6 discusses adaptive active sensing. We present our formulation to the adaptive active sensing problem, our design on the state representation and reward signal, and how we handle the unknown (and delayed) reward challenge. In particular, we propose a k-routine mining algorithm to discover frequent routine patterns, and present some learned k-routines using real mobile sensing data. These learned k-routines are adopted to represent a user's high-level routine state. Simulation results show that adaptive active sensing method can lead to improvements in various response compliance metrics (e.g. response latency, and overall response rate).

Chapter 7 demonstrates our ongoing efforts in the development of a comprehensive adaptive sensing framework built on top of an existing mobile crowdsensing platform (Sensus) [236]. Chapter 8 discusses future direction and challenges in mobile sensing system implementation, while Chapter 9 concludes the current dissertation by summarizing all the above works and laying out our future directions.

Chapter 2

Background

In this chapter, we first review the existing hardware sensors in smartphones, state-of-the-art mobile sensing applications, and several important challenges that hinder their further progress. Then we briefly introduce reinforcement learning, and provide technical details on several RL topics that will serve as the foundation for our proposed adaptive sensing algorithms.

2.1 Mobile Sensors

Smartphones have become an integral part to our lives, serving us for many purposes including but are not limited to communication, information needs, study and work, entertainment, navigation, banking, and safety. All these smartphone applications are made possible by several technological advancements including the high speed wireless network, the access to hundreds of thousands of mobile apps distributed through app stores, and hardware improvements (e.g., larger and higher resolution display, larger battery, smaller form factor). [112] Among the many hardware improvements is the increasingly enriched set of embedded sensors, which can be leveraged in a variety of mobile sensing applications. These sensors can be roughly classified as motion, positioning, ambient, device usage, and physiological sensors based on their capabilities in measuring different aspects of human behavior, and can be selected and configured according to different application requirements [82].

Motion Sensors

The main motion sensor applied in most existing mobile sensing applications and studies is accelerometer, an eletromechanical instrument that measures acceleration along three orthogonal axes relative to the center of the earth. [89] Another motion sensor is gyroscope, which measures the angular rotation rate along three axes of the device. The integration of the gyroscope in smartphone has allowed for more accurate recognition of movement within a 3D space than accelerometer alone. However, accelerometer is oftentimes sufficient for recognizing motion characteristics associated with various activities (e.g., walking, running, sitting/being still), and is adopted widely alone without gyroscope in activity recognition tasks [121, 180]. However, due to the uncontrolled position and orientation of smartphones in users' daily usage, combining the two may lead to improved performance in activity recognition.

Positioning Sensors

Global Positioning System (GPS) leverages the delay times among broadcast signals from a number (at least three) of satellites to measure the geographical position on earth using the longitude and latitude coordinate system. [89] As long as the required satellite signals are accessible, we can measure the position of the smartphone user anywhere on earth. In contrast, Bluetooth, a wireless technology standard for exchanging data between fixed and mobile devices over short distances using short-wavelength UHF radio waves, registers users' position by the known referenced position of the fixed Bluetooth device. Thus, Bluetooth technology can only capture user's position when his/her mobile device enters the broadcasting range of the referenced fixed Bluetooth device (e.g., a static Bluetooth enabled computer). In the same vein, WiFi-enabled devices (e.g., smartphones) can scan surrounding Wireless local area networks with registered addresses, and log their media access control addresses (MAC) or service set identifiers (SSID). The user's position can then be identified using this information. [89] A fourth positioning method utilizes cellular signal from the referenced cell tower to identify the smartphone user's position at a cell tower level.

Ambient Sensors

Ambient sensors, including but are not limited to microphone, light, and barometer sensors, can capture the surrounding contexts of the smartphone users. Specifically, microphone is an acoustic transducer that records the sound signals from both human speakers and other sources of sounds; light sensor measures the intensity of illumination on the front-facing surface of the mobile device; and barometer sensor collects the environmental pressure, which can then be converted into estimates of variation in altitude. Due to the unknown placement of the smartphone, the measurement accuracy of these sensors can be compromised. For example, if the phone is placed within the pant pocket, the recorded sound maybe muffled, and the light level may not reflect the true lighting condition. Plenty of existing works have leveraged these sensors in mobile sensing applications for context-aware computing [133, 47, 189]. Other ambient sensors include temperature sensor (thermometer) and humidity sensor etc.

Device Usage Sensors

Device usage sensors refer to software sensors that log the running and usage of different apps installed on the smartphone. Of important interests to us are battery levels, communication records (e.g., calls and text messages), screen on/off, and social media logs (e.g., Facebook and Twitter messages). Battery levels can be used to control the deployment of various other sensors, aiming to avoid interference of smartphone users' normal usage from mobile sensing applications. [55, 89] Communication records and social media app usages reflect users' remote and virtual social interactions with others, while face-to-face social interactions can be approximated by Bluetooth encounters [233, 234]. Lastly, screen on/off directly signals the starts and endings of phone usage episodes, and can be used to understand users' phone usage patterns, which is very interesting by itself as many studies [59, 71, 31] have shown statistically significant connections between phone usage patterns and various mental health conditions.

Physiological Sensors

Although our focus in this work is on mobile sensing using personal smartphones, wearable devices such as smartwatches and wrist bands are increasingly used for health tracking among smartphone users, and can be directly connected with mobile phones for monitoring various physiological signals such as heart rate, skin conductance, respiration, and body temperature. [24] Typical wearable device embedded sensors include galvanic skin response (GSR) sensor, photoplethysmography (PPG) sensor, accelerometer, and temperature sensor etc. One advantage of these wearable devices over smartphones is their fixed position on the wrist or other parts of the body, as compared to the uncontrolled orientations and positions of smartphones. This may lead to better performance in context recognition tasks. Also, when users are wearing them, they are effectively measuring users' physiological signals and other motion signals without interruption; while smartphones may be more likely left away from owners' whereabouts, making the collected data irrelevant to their real contexts. These wearable devices can be considered as extended components to the associated personal smartphones, therefore when we talk about mobile sensing, we do not make such differentiation unless necessary.

These sensors, although grouped by different aspects of information they are capable of collecting, generate longitudinal signals with different formats (e.g., numeric and categorical sequences) and intensities (e.g., different sensing rates). Certain combinations of these information can be applied to form understanding of the complex contexts users situate, but it requires building effective computational models.

2.2 Mobile Sensing Applications

Equipped with such a diverse set of sensors, smartphone has become the test bed for many innovative sensing applications. In this section, we survey some of the relevant mobile sensing applications that leverage smartphone's passive and active sensing capabilities.

Activity Recognition and Context-aware Computing

Activity recognition and context-aware computing refer to the usage of raw sensing data collected by various sensors to understand users' ambient and physical state. Table 2.1 summarizes some representative works in this domain using mobile sensing. The applied sensors range from accelerometer, GPS, microphone, Bluetooth, barometer, gyroscope, light sensor, WiFi, cell tower signal, to phone usage logs. The activities can be categorized as ambulatory activities (e.g., walking, running, and being still), transportation mode (e.g., walking, biking, on a bus, and driving), mobility (e.g., next place prediction), social activities (e.g., studying, dining, and partying), ambient environments (e.g., human conversation, quiet and noisy environment, indoors vs. outdoors), sleeping (e.g., sleeping duration and quality), and phone usage (e.g., app preference).

TABLE 2.1: Physical contexts. Sensors: Accelerometer – acc, Microphone – mic, Barometer – br, Bluetooth – bt, Gyroscope – gs, Phone usage – pu, Communications (calls and sms) – comm, Compass – cp, cell tower – ct

Category	Behavior Details	Sensor(s)	Ref(s)
Ambulatory activities	15 activities (e.g., walking, running, vacuuming).	acc,br,mic	[101]
	10 activities (e.g., sitting, walking, jogging).	acc,mic,light,br,cp	[117]
	Static (e.g., standing, sitting, lying), and movement (e.g., walking, running, ascending stairs).	acc	[121]
	6 activities (e.g., sitting, standing, walking).	acc,gs	[180]
	11 activities (e.g., working, meeting, resting, home and talking)	acc,GPS,mic	[227]
Transportation mode	walking, biking, taking a bus, and driving a car.	GPS	[126]
	stationary, walking, running, biking, or in motorized transport.	GPS,wifi,ct	[182]
	idle, walking, and vehicle.	br	[189]
	walking, taking a bus, riding a bike, and driving.	GPS	[246]
Mobility	predict next place	GPS	[127]
	predict next place	GPS	[204]
Social/daily activities	5 activities (e.g., meeting in the hall, working indoors, watching a movie in the cinema).	bt	[46]
	buying food and drink	acc,wifi,mic	[44]
	activity changes in an academic term (e.g., levels of activity and sociability)	acc,mic	[80]
	7 acitivities (e.g., Shopping, transportation, housekeeping, work, sleeping, watching TV).	acc,mic	[67]
Ambient environments	voice, music, and ambient sound.	mic	[131]
	5 sounds (e.g., opening polymer packet, crushing packet).	mic	[209]
sleeping	sleep duration, onset	mic	[2]
	sleep duration	mic,light,pu,acc	[47]
	sleep	acc,mic,light, pu	[150]
Phone usage	app usage	acc,mic,GPS,pu	[237]

Mobile Crowd Sensing

Mobile crowd sensing (MCS) is a community sensing paradigm that leverages pervasive mobile devices owned by individual users to collect data for many urban scale applications. [70] Applications of MCS include but are not limited to air quality monitoring [83], traffic monitoring [151, 60], emergency management [104, 63], road surface monitoring [38], place of interest recommendation [94], urban wifi characterization [62]. Depending on the level of participant involvement, MCS can be opportunistic or participatory. Opportunistic sensing requires minimal user involvement (e.g., automatic location sampling), while participatory sensing requires active user involvement (e.g., taking a photo of a parking garage). Since participants in MCS expend resources (e.g., time, money, efforts) to complete the assigned tasks, most of the research problems in MCS revolve around user participation. The major challenges in MCS include recruitment and incentives [244], data quality (e.g., coverage, latency, and confidence) [128] and transmission [135], task assignment [220], power efficiency [54], and privacy [70].

Mobile Ecological Momentary Assessment

Ecological Momentary Assessment (EMA) is an intensive surveying approach that allows subjects to repeatedly report their experiences in real-time and in situ. Comparing to retrospective self-reports, EMA has become the new norm of data collection for a wide range of research areas (e.g., clinical assessment [78, 16], psychology/cognitive process and their mechanisms [229, 208, 13], mobile health [99, 116]) focusing on gaining a better understanding of the dynamics in human behavior and experience over time and across situations [196]. Mobil EMA leverages the mobility of smartphone users and their closed proximity to these mobile devices to ecologically collect data. With the passive sensing capabilities, these active data can be further augmented by context, making the EMAs context-aware.

Applying mobile EMA, many researchers have studied technology utility experiences to inform better product/service design [93, 65, 43], clinical assessment and research (e.g., schizophrenia [78], suicide ideation [16], major depressive disorder [153, 206], HIV progression [61]), mental health [163, 214, 85, 160], health behaviors (e.g., alcohol use [232], binge eating [5], behavioral medicine [186]), psychology/cognitive process and their mechanisms [229, 208, 13], academic emotions [75], organizational research [33, 106, 14, 90, 192], tourism and live event experiences [20, 138, 198, 230], and mobile health (mHealth) [99, 116, 159, 108].

Mobile Health and Intervention

Mental state and health behaviors include various aspects of people's mental well being (e.g., stress, depression, anxiety, mood/emotion, happiness), personal and relatively enduring characteristics (e.g., personality, attributional style), health symptoms and diseases (e.g., cancer symptoms, Parkinson's disease), and health related behaviors (e.g., drinking, smoking). Table 2.2 summarizes some representative works in mental and health behavior inference using mobile sensing. A similar set of sensors were applied when compared to those in physical activity recognition in Table 2.1. Most of these researches attempted to establish the links between passively sensed human contexts and health outcomes (e.g., stress [3], schizophrenia [222], and psychotic relapse [18]). They indicate that interventions and recommendations informed by passive sensing may effect positive behavior changes in applications such as regulating anxiety [51] and suggesting personalized physical activities [176].

Mobile intervention aims to lower barriers to treatments for a wider range of populations and ultimately improve certain health outcomes, especially chronic diseases [200]. Researchers have investigated several engineering frameworks to implement efficient and effective mobile interventions. Timms et al. proposed to use control engineering framework to model mobile interventions [111, 207], while Kelly et al. propose to use RL to facilitate the delivery of intelligent real time treatment (iRTT) with continuous self-reports through EMAs [100]. Both of these proposed methods are dynamic and leveraging active sensing to obtain real-time understandings from the users, which in turn will serve as the controlled variable in the control engineering framework or reward signal in the RL framework, to determine intervention regime and timing of delivery.

A special type of behavior interventions, namely Just-in-time Adaptive Intervention (JITAI), provides supports when the users are most in need through adaptation in the delivery timing and intervention regime. Frameworks guiding the Design of JITAIs with mobile technologies have been proposed by Nahum et al. [154]. However, implementation of adaptive and interaction systems that deliver JITAI remains very challenging, and most existing works either involved only conceptual developments and/or proof-of-concept, or were not truly adaptive in trigger timings (e.g., using recipient chosen timings, or when help/support is requested through EMAs [84]).

Table 2.2: Mental and health behaviors. Sensors: Accelerometer – acc, Microphone – mic, Barometer – br, Bluetooth – bt, Gyroscope – gs, Phone usage – pu, Communications (calls and sms) – comm, Compass – cp, cell tower – ct.

Category	Behavior	Sensor(s)	Ref(s)
Stress	daily stress level (e.g., non-stressed vs. stressed).	bt, comm	[21]
	stress (e.g., stressed speech, and neutral speech).	mic	[133]
	stress (e.g., below and above median).	GPS	[149]
	momentary stress (e.g., stressed vs. not-stressed).	acc, GPS, mic, bt	[233]
	stress (e.g., momentary stress with two levels stressed and non-stressed; daily stress on 7-point scale).	GPS, bt	[234]
Depression	daily depression level based on PHQ-8 using cut points (e.g., mild, moderate, moderately severe and severe).	GPS	[39]
	depression behavior using CES-D survey instrument.	mic, light, gps, acc, pu, comm	[55]
	daily depression using PHQ-8 (0 and 1 representing absence and presence of depressive symptom).	GPS	[143]
	post-traumatic stress disorder (PTSD) and depression using SCID (0 and 1).	GPS,mic,comm	[169]
	depression using PHQ-9 and 5 as cut point (0 and 1).	GPS, pu	[188]
	depression using PHQ-9; anxiety using GAD-7 (both with cut-off point of 10 as binary classification task).	acc, mic, light, GPS, comm, wifi, pu	[187]
	depression using PHQ-8 (term, cut point 10) and PHQ-4 (weekly,cut point 3)	acc, GPS, light, mic, pu	[225]
Anxiety	social anxiety using SIAS, depression using DASS-21, daily posive and negative state affect (0-100).	GPS	[48]
	social anxiety using SIAs.	acc,comm	[76]
Mood/emotion	emotion in two dimensions: pleasantness and arousal (activation).	acc	[69]
	negative emotion in three dimensions using visual analogue scales with 1/3 as cut points: depression, stress, and anxiety	pu, comm	[91]
	mood in Circumplex pleasure and activeness dimensions on 5-point scale.	pu, GPS, comm	[123]
	daily mood in three dimensions on 5-point scale: displeasure (overall), tiredness and tensity.	acc, mic, GPS, comm	[136]
	daily mood on a affect grid in two dimensions: valence and arousal.	acc, mic, GPS, comm	[195]
Diseases and Symptoms	daily Schizophrenia symptom level using positive and negative EMA question items.	acc, GPS, light, mic, pu, comm	[221]
	various health symptoms (e.g., Sad, Depressed, Stressed; Sore-Throat, Cough, Runny-Nose, Congestion, Sneezing; Flu and Fever Symptoms.)	bt, comm	[7]
Others	Happiness using affect grid with two dimensions: valence and arousal.	acc	[115]
	Personality using Big Five personality trait questionnaire.	acc, mic, pu, GPS	[226]
	Trust	bt,comm	[197]
	academic performance (i.e., predict GPA)	acc, GPS, mic, light, pu, wifi, bt	[223]

2.3 Challenges in Mobile Sensing

There are many challenges in mobile sensing that are beyond the scope of this dissertation. For example, in passive sensing, training activity recognition and behavior inference models requires a large quantity of human labeled data, which is extremely expensive (e.g., time, costs, and efforts) to obtain. Much work has been done in these areas as are shown in Table 2.1 and 2.2. However, the definitions of outcome metrics are not consistent, making it hard to compare performances, and reuse existing results. In active sensing, data quality may be compromised by careless and dishonest submissions from participants, which necessitates certain format of data authentication. [145, 217, 184]. Privacy issue is another concern in handling the collected sensing data, which may breach participants' personal information.

This dissertation focuses on the energy efficiency challenge in passive sensing, and the user compliance challenge in active sensing. Using the RL framework, we are able to unify these two disparate challenges under adaptive sensing by leveraging users' sensed contexts. With the understanding of users' contexts, on one hand, we can optimize the timing and contexts of sensor deployments to save energy while preserve sufficient data utility; on the other hand, we can determine when the best moments are for task triggers to obtain higher response compliance.

User compliance is a multifaceted challenge in active sensing. Although users' contexts play a critical role in their response behaviors (e.g., when they are unavailable or unattended to the triggers), it is not the only cause for low compliance. One biggest challenge mobile sensing research faces is the law of attrition [174]. Users usually lose their interest and motivation over time, resulting in failure of maintaining long term compliance and intervention efficacy. In order to solve this challenge, adaptive sesing needs to be coupled with other measures such as incentives [244], gamification [177, 218], and feedbacks (i.e., making participants' efforts rewarding and useful to them) [147, 108] to sustain long term compliance. In this dissertation, we estimate user's motivation using a moving window response compliance, and apply it as a state feature in our adaptive sensing methods.

2.4 Reinforcement Learning

What is the RL framework?

Reinforcement learning is a subdivision of machine learning that centers on the idea of learning through interactions in a given environment [203]. In contrast to supervised learning, in which learning is guided under knowledge of correct 'actions', learning in RL is based upon 'trial and error'. To characterize the interactions between a learning agent and its environment, RL follows the formal framework of Markov decision processes (MDP) using states, actions, and rewards. State refers to the representation that captures the key characteristics of the environment at a given moment relevant to the learning task. An action will be chosen in a state (of an environment) at each decision point, and a reward signal, typically a proximal performance metric directly generated by the environment in receipt of the selected action, will provide feedback to either strengthen or weaken the connection between the state and the chosen action in that state. This law of effect, as was derived by Edward Thorndike back in 1911 [203] in his observations from animal experiments, can be regarded as a basic learning principle underlying many human behaviors and artificial intelligence problems.

The main task in RL is to learn an optimal policy, a mapping between states and actions that can guide any future action selection in any given state and maximize the long term cumulative rewards. This is achieved through estimating a value function of the states. While the reward signal indicates the immediate and intrinsic desirability of the environmental states, the value of a state (as reflected by the value function) indicates their long-term desirability by taking into account the states that are likely to follow. In learning the optimal policy, we always seek actions that generate the highest values, instead of highest rewards, in those states.

This dissertation considers the RL framework in a broad sense, that is learning through interaction in a given environment, regardless of how the learning output is represented (i.e., be it a policy resulted from classical RL formulations, a simple decision rule, or a supervised model) to guide action decision. In this sense, many methods under control

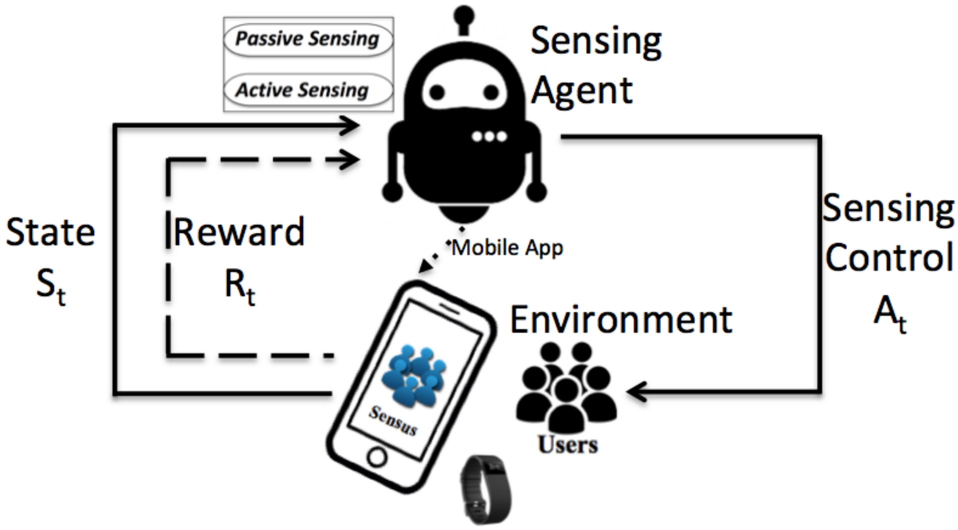

FIGURE 2.1: Reinforcement learning in adaptive sensing.

engineering that use a feedback mechanism can be considered as RL methods as well. Methods that leverage supervised models for action decision in an online fashion can also be considered as falling under the RL framework. Figure 2.1 illustrates the RL framework in the context of adaptive sensing.

In the following sections, we provide basic technical details on RL that are applied in the later expositions. We will only focus on the control problem rather than the prediction problem in RL due to its relevancy and constraints in space. Specifically, the prediction problem in RL is where a policy is supplied, and the goal is to measure the value of the policy; and the control problem in RL is where the policy is not fixed, and the goal is to find the optimal policy.

Markov Decision Process

Markov Decision Process (MDP) is a classical formalization of sequential decision making problems, which simplify the problem of learning from interactions with the environment into a sequence of discrete decision points. At each decision point, the agent understands the environment through a state representation, and chooses an action based on this knowledge. As a consequence of its action, the agent receives a numerical reward that signals the immediate desirability of the chosen action given the current state.

Traditional MDP is defined as a five-tuple $M = \langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R}, \gamma \rangle$, where $\mathbf{S}$ is the state space, $\mathbf{A}$ is the action space, $\mathbf{T} \in (Ps)^{\mathbf{S}\times\mathbf{A}}$ is a transition function, $\mathbf{R} \in \mathbb{R}^{\mathbf{S}\times\mathbf{A}}$ is a reward function, and $\gamma \in (0, 1)$ is a discount factor. Among these five elements, $\mathbf{T}$ provides an environment model that designates the state transitions given the chosen actions. Depending on whether $\mathbf{S}$ is continuous or discrete, $T(s'|s, a)$ can be understood as a probability density or mass function describing the probability of reaching state s' if action a is taken in state s. However, the environment model is not required to solve a RL problem as in almost all practical problems, the true environment model will be unknown and extremely challenging to accurately estimate.

We apply model-free RL methods, which requires no environment model, to solve the mobile sensing problems in this dissertation. Although the MDP formulation is not exactly followed, it does provide us useful abstractions for the mobile sensing problems, and the basic terminologies that we will follow in the rest of this dissertation, including the state and action space, and reward function.

Using the MDP formulation, a RL problem that characterizes the online interactions between the agent and environment proceeds according to the following protocol. At timestep $t = 1, 2, 3, \ldots$,

1. The agent perceives the current state $s_t \in \mathbf{S}$, and takes an action $a_t \in \mathbf{A}$.
2. In response, the environment sends an immediate reward r_t to the agent, and moves to the next state $s_{t+1} \in \mathbf{S}$.
3. $t \leftarrow t + 1$

Temporal Difference Methods and Q-Learning

The goal of RL is to search for an optimal policy that can result in maximal cumulative long term rewards. To achieve this goal, it leverages action value functions $q(s, a)$, which represent the expected cumulative future discounted rewards of taking a certain action a in a given state s. Let us denote q_* as the optimal action value function. According to the Bellman optimality equations, we have:

$$\begin{aligned} q_*(s,a) &= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a] \\ &= \sum_{s',r} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')] \end{aligned} \tag{2.1}$$

where $p(s', r|s, a)$ is the environment model that specifies state transitions and reward generation. There are two major issues with Equation 2.1. First, we do not know the environment model in almost all practical problems, and secondly it is not in an incremental format that can be implemented for policy searching. Fortunately, both of these problems become non-issues in the temporal difference (TD) learning methods.

TD learning is considered the most central and novel technique in RL. [203] It combines the dynamic programming (DP) ideas (e.g., learning a guess from a guess, or in another word, bootstrapping) with the Monte Carlo ideas (e.g., requiring no environmental model). Learning in Monte Carlo methods must wait until the end of an episode, while TD methods learn in every time step, thus making it much more efficient in continuing tasks or tasks that have long episodes. At each time step, the TD update is given by

$$V(S_t) \longleftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \tag{2.2}$$

This TD method is called $TD(0)$, or one-step TD, a special case of the TD(λ) when using eligibility-trace. We will defer eligibility-trace as a later topic below.

Of all the TD methods, we focus on an off-policy TD control algorithm known as Q-learning. Its per time step update is given by

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_{a} Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{2.3}$$

The Q-learning algorithm is off-policy because its update policy and behavior policy are different. Algorithm 1 [203] shows the Q-learning algorithm for learning the optimal policy. For proofs of policy improvements and value function convergence, please refer to [203] for more details.

Algorithm 1 Q-learning for estimating optimal policy π_*

Initialize $Q(s,a)$, for all $s \in S$, $a \in A(s)$, arbitrarily, and $Q(\text{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of an episode):
    Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_{A'} Q(S',A') - Q(S,A)]$
    $S \leftarrow S'$
  until S is terminal

Linear Approximation

When the state space $\mathbf{S}$ is continuous, it is impossible to encounter all possible state values in the learning process. Thus the learned policy needs to be able to generalize to unseen state values. To achieve this goal, instead of the classical tabular RL methods, we apply a technique called function approximation. Let us denote the parametric action value function as $q(s,a,\mathbf{w})$, where $\mathbf{w} \in \mathbb{R}^d$ is a finite dimensional parameter vector. The action value function can be either linear or non-linear. Non-linear functions include neural networks and possibly other supervised methods such as Support Vector Machin (SVM). We will only focus on linear approximation in this dissertation.

The update for the one-step Q-learning method with linear approximation is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha[R_{t+1} + \gamma \max_{A_{t+1}} q(S_{t+1}, A_{t+1}, \mathbf{w}_t) - q(S_t, A_t, \mathbf{w}_t)]\nabla q(S_t, A_t, \mathbf{w}_t). \tag{2.4}$$

With $q(S,A,\mathbf{w}) = \Phi(S,A)\mathbf{w}^A$, where $\Phi(S,A) \in \mathbb{R}^d$ is a d-dimensional feature vector, and $\mathbf{w}^A$ is the parameter vector associated with action A, Equation 2.4 can be rewritten as

$$\mathbf{w}_{t+1}^{A_t} = \mathbf{w}_t^{A_t} + \alpha[R_{t+1} + \gamma \max_{A_{t+1}} \Phi(S_{t+1}, A_{t+1})\mathbf{w}_t^{A_{t+1}} - \Phi(S_t, A_t)\mathbf{w}_t^{A_t})]\Phi(S_t, A_t). \tag{2.5}$$

Eligibility-trace

Eligibility trace is a basic and general mechanism in RL that can be combined with most TD methods such as Q-learning to improve learning efficiency. There are two different views on eligibility trace, the forward and backward views. The forward view provides a clear understanding on how eligibility trace actually enhance learning efficiency by proportionally weighting all the n-step updates according to λ^{n-1}. The resulting update is called λ-return, and is given by

$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_t \tag{2.6}$$

where $G_{t:t+n}$ is the n-step update given by

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n q(S_{t+n}, A_{t+n}, \mathbf{w}_{t+n-1}), 0 \leq t \leq T-n.$$

However, the forward view of eligibility trace depends on future rewards that are not available at the current moment, and is much more complex to implement. To address this challenge, the backward view provides an online, real time, and more computationally efficient alternative that does not depend on future rewards, but look backward to recently visited states using eligibility trace. The rough idea is to magnify the updating of the corresponding parameters in the most recent state. The trace-decay parameter $\lambda \in [0, 1]$ determines decaying rate of the trace. Let us denote the eligibility trace by $\mathbf{z}_t$, and the TD error by $\delta_t = R_{t+1} + \gamma q(S_{t+1}, A_{t+1}, \mathbf{w}_t) - q(S_t, A_t, \mathbf{w}_t)$. Two key equations for the TD(λ) algorithm are given by

$$\mathbf{z}_t = \gamma\lambda \mathbf{z}_{t-1} + \nabla q(S_t, A_t, \mathbf{w}_t), \quad 0 \leq t \leq T \tag{2.7}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \delta_t \mathbf{z}_t \tag{2.8}$$

Equation 2.7 is the eligibility update formula, while Equation 2.8 is the parameter update formula.

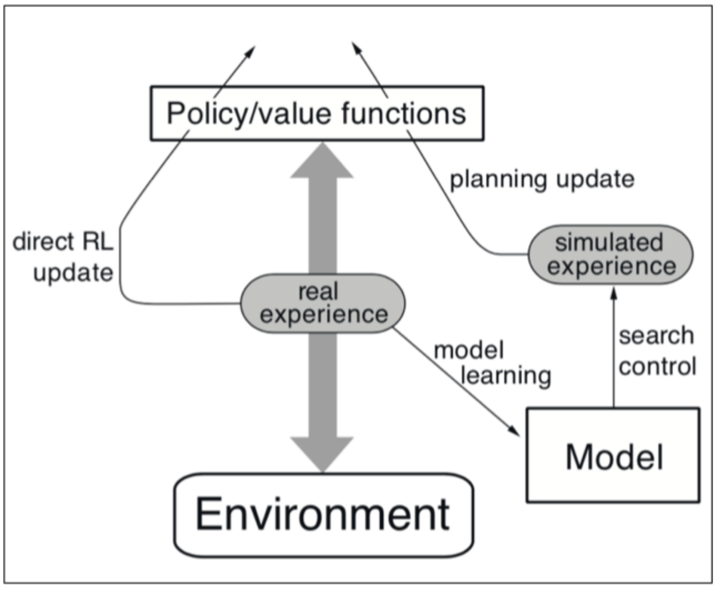

FIGURE 2.2: Dyna-Q Framework from [203].

Dyna-Q

One of the biggest challenges we face in applying RL in solving practical problems such as those in mobile sensing is the shortage of real learning experiences. One way to address this challenge is to augment the limited real learning samples by integrating planing with learning. Dyna-Q is a simple architecture that integrates planning with learning in an online agent. Figure 2.2 shows this architecture in Dyna-Q and Algorithm 2 shows one way of implementing it.

Algorithm 2 Tabular Dyna-Q [203]

Initialize $Q(s, a)$ and $Model(s, a)$ for all $s \in S$, $a \in A(s)$
Do forever:
(a) $S \leftarrow$ current (nonterminal) state
(b) $A \leftarrow \epsilon-$greedy(S,Q)
(c) Execute action A; observe resultant reward, R, and state, S'
(d) $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
(e) $Model(S, A) \leftarrow R, S'$ (assuming deterministic environment)
(f) Repeat n times:
$S \leftarrow$ random previously observed state
$A \leftarrow$ random action previously taken in S
$R, S' \leftarrow Model(S, A)$
$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$

The main issue in applying Dyna-Q in the adaptive active sensing problem in particular, and in other practical problems in general, with limited learning experiences, is about creating the environment model for generating simulated experiences. In tabular Dyna-Q, simulated experiences can be generated by randomly sampling past real experiences in the format of $S_t, A_t \rightarrow R_{t+1}, S_{t+1}$. In this way, the model will never be queried with state action pairs that it has no information from the past. However, the downside of this approach is that the sample model can not generate new unseen samples for planning. In supervised learning, there is a technique called Synthetic Minority Oversampling Technique (SMOTE) for oversampling imbalanced class. The idea of this technique is to generate a new sample from those in the minority class using a random sample and its k nearest neighbors. It is possible to adapt this approach to overcome this generalization issue in bootstrapping.

Chapter 3

Feature Extraction for Behavior Modeling in Mobile Sensing Data

In this chapter, we propose a feature extraction framework for mobile sensing data to guide the extraction of state features based on systematic reviews of existing works. We close this chapter with a presentation of an online location mining algorithm that will be used throughout this dissertation, and some summary statistics from applying it in real GPS data.

3.1 Introduction

Recent advances in mobile and passive sensing technologies have provided a new avenue for researchers to understand and model human behaviors. Passive sensing, unobtrusively capturing fine-grained behavioral data via smartphone and wearable embedded sensors, can provide fine-grained objective data about users' contexts and behaviors with minimal burden; while mobile EMAs provide subjective data with minimal recall and expectancy biases, and maximal ecological validity. These multimodal data can be combined to understand more complex behaviors such as choice of transportation, sleep, social activities, and mental health [18, 10, 182, 126].

In order to create valid models from mobile sensing data, researchers must go through a process of transforming the raw sensor outputs into meaningful explanatory variables that represent aspects of behavior (i.e. engineered features). These engineered features can be subsequently aggregated or mined for patterns that may be associated with more complex behaviors, such as those found in the domain of mental health (e.g., depression and anxiety) [27, 48, 39].

There are significant challenges in this feature extraction process. The relevance of features should be informed by behavioral domain knowledge (e.g. behavior theories of change [148], models of psychopathology [73]), which may not always be available. Even when available, their impacts can be limited because many prominent behavior models and theories do not account for the temporal aspect of behavior (e.g., duration of social isolation [48]). [185, 202] Furthermore, sensing data can be processed in many ways by statistical and mathematical functions, leading to poorly understood features. Finally, automatic feature extraction methods, such as deep learning, have become more prevalent in recent years. [24, 143] However, these methods require large quantity of training data, which are not available in many human-centric problems.

In addition to the above challenges, an obstacle of successful behavior modeling, is the absence of standards in feature extraction. There is a lack of systematic reviews targeting feature extraction techniques in mobile sensing data [171]. To date, feature extraction has been mostly conducted in a case by case, application-dependent basis. Although many reviews in mobile sensing have provided perspectives on challenges and outlooks in this field, none of these works propose a feature extraction framework. As a result, a method that works well for one study can not be assumed to work well in a different study. An overarching framework is urgently needed to guide feature extraction for the development of machine learning models from raw sensor data.

In this chapter, we survey representative works that leverage mobile sensing to understand and model human behaviors, and summarize their techniques in feature extraction. While prior works have focused on the general design of mobile sensing applications and their accompanying challenges, this chapter mainly focuses on the feature extraction process and emphasizes the methods that are used to encode and extract features from preprocessed raw sensing data. Based on these reviews, we propose a computational framework to guide the design and extraction of features from raw sensor outputs to support human behavior modeling.

3.2 Related Work

Lane et al. published a paper on mobile phone sensing in 2010, recognizing the emerging sensing technologies at the time, along with their enormous potential for a wide range of applications (e.g., transportation, social networking, environmental monitoring, and health) at different levels (e.g., individual, group, and community). [112] Their proposed sensing architecture has been influential in more recent mobile sensing works. Specifically, the 'learning' phase of their framework discussed challenges in interpreting different sensor data; however, they did not explained feature extraction methods in them for behavior modeling using machine learning.

While [112] presented various opportunities and challenges within general purpose mobile sensing, [50, 152, 231] focused on personal sensing in health and well-being. In [50], the authors systematically reviewed works on smartphone sensing for health and well-being, identifying 35 papers and summarizing them based on their health problem, study goal, principal findings, adopted sensors, sample size, and study duration. Learning from these summaries, they provided a brief discussion on the various challenges and limitations in mobile sensing. They emphasized the importance of choosing sensing strategy and adopting personalized models, and proposed future research directions including application domains, policy and privacy etc.

In [152], the authors proposed a hierarchical model for translating raw sensor data into markers of mental health. Within this framework, raw sensing data are distilled into information, which is further combined to form behavior markers for mental well-being monitoring and detection. They surveyed existing works that leverage different platforms (e.g., mobile phones, wearable devices, social media, and computers), and grouped them according to different types of behavioral markers (e.g., sleep, social context, mood and stress) and clinical disorders (e.g., depression, bipolar disorder, schizophrenia). They then discussed broadly general machine learning methodologies such as supervised and active learning, existing challenges, and potential applications in personal sensing.

Woodward et al. provided an overview of traditional and newer sensing technologies for clinical assessment, monitoring, and intervention. [231] Specifically, mHealth apps and 'tangible interfaces' (i.e., digital interfaces embedded in physical environments) in various applications were reviewed to understand how they were leveraged in mental health monitoring, and how virtual and augmented reality can be used to improve mental well-being. Challenges in these newer technologies, such as data collection, privacy, and battery life, were also discussed. The authors recognized the growing opportunities in the applications of sensing and feedback technologies to develop robust clinical decision support systems for the diagnosis and intervention of mental disorders.

These reviews did not cover feature extraction in depth, and therefore can not provide practical guidance to feature extraction for behavior modeling in mobile sensing data.

Shmueli et al. surveyed mobile sensing works based on the types of the targeted human behaviors (e.g., personal vs. inter-personal, and short-term vs. long-term). [197] Personal and short-term behaviors include emotional states and activities; personal and long-term behaviors include personality, health and wellness; inter-personal and short-term behaviors include roles in meetings, and short-term group interactions; and inter-personal and long-term behaviors include community structure and organizational effectiveness. However, They did not focus on low-level derived features via different sensing modalities, such as physical motion patterns using accelerometer data.

Harari et al. reviewed mobile sensing works in social interactions, daily activities, and mobility patterns, and describe specific features extracted from each smartphone embedded sensor. [82] However, they did not discuss the methods used to extract them. In [81], the same authors summarized more concrete features extracted for similar groups of behaviors including physical movement, social interactions, and daily activities. However, neither feature extraction framework, nor summary of feature extraction techniques was presented to guide behavior inference using mobile sensing data.

Hoseini et al. surveyed existing researches and approaches towards implementation of systems in context recognition using mobile phones. [89] In data preprocessing, the authors categorized features into heuristic, time and frequency domain features under different context recognition tasks (e.g., recognizing physical activity, detecting social interactions, and sensing ambient environment). They considered the feature extraction process as a computational process with preprocessed sensing data within a selected window as input, and the distilled information (i.e., more computationally efficient and lower-dimensional forms called features) as output. We share the same view with this work on feature extraction as an **input-featurization-output** computational process, and build on this foundation to propose a feature extraction framework that can guide feature extraction with various techniques.

3.3 A Feature Extraction Framework

Raw data from mobile sensing applications are messy and high dimensional, and contain a large amount of noise that interferes with the behavior modeling process. Feature extraction is the process of cleaning, organizing, and reducing the dimension of raw data for behavior modeling. While some authors call it data preprocessing, and others feature engineering or representation learning, no one really formalizes it in the contexts of behavior modeling using mobile sensing data. In this section, we propose a computational framework for feature extraction towards behavior modeling. Figure 3.1 shows the proposed framework. The input component includes time series data collected from various embedded sensors in mobile sensing; the feature extraction component includes preprocessing (e.g., imputation, normalization, and transformation), encoding, and featurization; and the output component includes the generated features for behavior modeling. For the rest of this section, we review existing mobile sensing works based on this proposed framework.

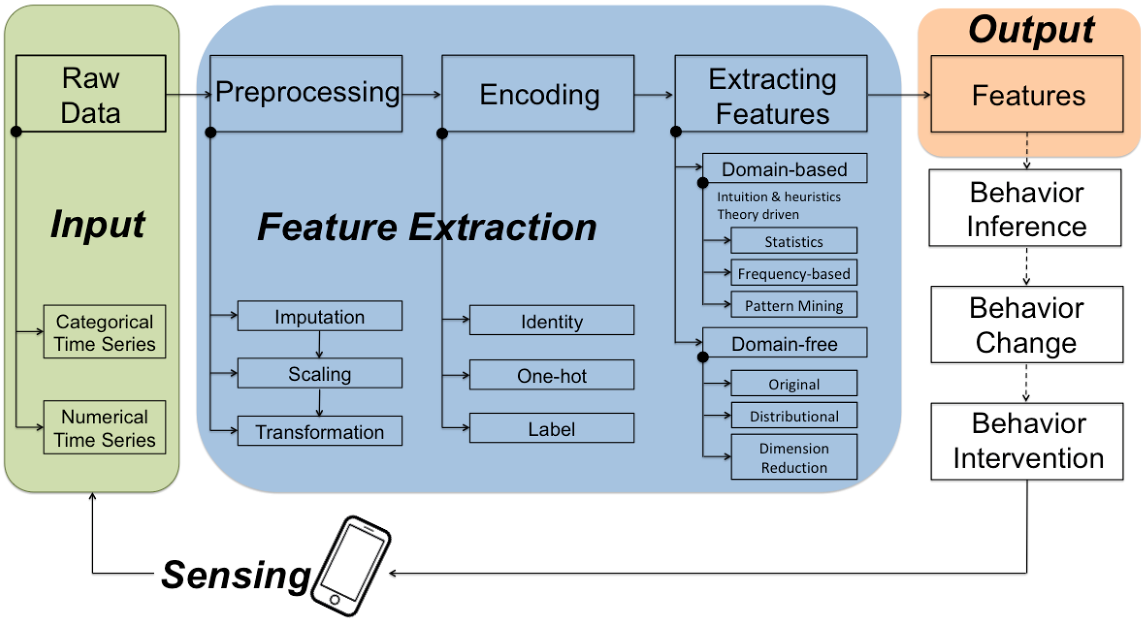

FIGURE 3.1: A Computational Framework for Feature Extraction.

3.3.1 The Input

The raw data collected from various smartphone embedded sensors can be grouped based on their data formats, including continuous (e.g., accelerometer and GPS data) and categorical (e.g., app usage logs and screen on/off). Due to their different formats, feature extraction is conducted using different methods.

Continuous Time Series. Continuous time series can be aggregated using numeric-based statistics such as mean and standard deviation [227, 126], or transformed using Fast Fourier Transformation (FFT) [117, 171, 209] and Wavelet Transformation (WT)[110, 121]. Density-based methods have also been applied to estimate distribution of continuous time series, and percentiles of these empirical distributions can be extracted as features for behavior inference (e.g., recognizing physical activities [171]).

Categorical Time Series. Categorical time series apply a rather different sets of methods for feature extraction, including frequency-based and pattern mining methods [39, 82, 130, 92]. Counts of each category within a designated time window (e.g., number of visits to certain places [39, 82]) and patterns of categorical sequences (e.g., visiting a local elementary school before going home [130]) are among some of the most popular features.

Timestamped Data. The raw sensing stream is timestamped, and provides a rich space for many different techniques to extract feature. We can extract the duration of certain events such as distribution of time spent at different places, time intervals between different consecutive place visits, and proportion of time being active each day [48, 36].

Conversion between Data Formats. Raw sensing data can be converted from one format into another one. For example, some researchers derived features (e.g. distance traveled) directly from GPS coordinate sequences [39], other researchers convert the numeric GPS coordinate time series into categorical place visit time series using clustering algorithms to learn semantic places of users [97, 125].

It is important to understand what data format of a sensing stream is, what meta data associated with the data stream are available, and whether they can be converted into another data format to extract more important features for behavior modeling.

3.3.2 Feature Extraction

Data Preprocessing

Data preprocessing refers to various cleaning and preparation steps including imputing missing data, normalizing data in different scales, and transforming data from one format to a different format. There are no universal rules to follow in these steps. To illustrate, we provide some typical examples in each of them.

Imputation. Missing data can result from various conditions (e.g., app failure, drained battery, and interference from other apps) during the data collection process. Imputation is required for certain subsequent steps in the preprocessing pipeline (e.g., transform the data from time domain to frequency domain using FFT). Depending on the format of the time series data, we can apply different imputation methods such as moving average for continuous data, and last value carried forward for categorical data. When temporal relation is weaker (e.g., hourly step counts on a given day), the average value in the corresponding hour across different days for continuous data, or the most encountered value for categorical data, can be applied to impute missing values. Many imputation methods are available in the literature and on the web [1]. It is important to select an appropriate imputation strategy based on the context of the data, amount of missingness, and their missing mechanism.

Scaling. Scaling data is required when multi-modal sensing data are applied in behavior modeling. Typical scaling strategies include 0-1 scaling using minimum and maximum values, and normalization using mean and standard deviation. The former strategy results in values ranging from 0 to 1, while the later strategy converts the original scale onto number of standard deviations below or above the mean value without enforcing lower and upper bounds.

Transformation. Transforming data is a preparation step for extracting certain features. One example is to transform the data from time domain to frequency domain for frequency domain features. Popular transformation algorithms include FFT [117, 171, 209] and Wavelet transform (WT) [110, 121].

A second example is to convert the GPS coordinate sequences into place visit trajectories using various clustering algorithms [9, 247, 97, 240, 125, 40]. In [9], the authors proposed to recognize significant places as any logged GPS coordinate with an interval of 10 minutes between the previous point and itself. These found significant places are clustered by initially picking a point and a radius of half a mile, using the mean of all points that fall within the radius to the initial point as a new center, and repeating this process until the center does not change any more. This center is chosen as the learned location. Subsequently, all these included points are removed from the list, and the process is repeated until the list is empty. In [247], a variation of DBSCAN clustering algorithm with temporal filtering is applied to learn semantic locations. In [97], the authors proposed a tempo-spatial clustering algorithms that takes the GPS coordinate trajectories and two parameters (a distance threshold and a time threshold) as input, and learn significant clusters as the places the user visited. In [240], the authors proposed a two level place detection algorithm. In the first level, stay points are first detected given a distance and

[1] https://medium.com/@Cambridge_Spark/tutorial-introduction-to-missing-data-imputation-4912b51c34eb

a time threshold parameter. In the second level, these stay points, which are considered visits to different places, are clustered based on a DBSCAN algorithm called OPTICS to form the final learned unique places. Cao et al. proposed a merging method that combined DBSCAN learned clusters based on similarity scores derived from various visit metrics from GPS data across all users before reverse geocoding them to obtain the semantic labels of different significant places. [40] Although these clustering algorithms can learn significant places from GPS coordinate trajectories, we do not know semantic labels such as home and office to these places. Heuristics can be used to discover home and work places for most people (e.g., the place that people spent most time everyday from 10pm to 6am can be considered as home with very high confidence). To obtain semantic labels for other places, external database such as Four-square location service, Google Maps API, and OpenStreetMap API can be leveraged.

There are several challenges in learning location clusters and obtaining semantic labels for them. First, most of the experiments in these studies collect high frequency GPS data with specialized device, which may not take into consideration battery drainage problem. In reality, when the rate of GPS data has to be reduced to save battery consumption, the parameters of these proposed methods can be completely different, or worse, the algorithms may not work due to the sparsity of the GPS coordinate trajectories. Second, external database may not always be available, thus discovering semantic labels may not be possible. Lastly, what semantic labels to choose is very application dependent, and reliability of methods to associate labels to the location clusters can not be guaranteed (e.g., choosing the semantic label of the nearest building as the label of the learned location cluster), and thus it requires substantial experiments to fine tune the results. In any case, the more complicated the method is, the more difficult they are to be generalized across real world applications.

Data Encoding

Data encoding (or data representation) is the process of representing the data stream in a certain format to work with computational models. Data encoding is particularly needed for categorical data stream such as place visit trajectories because categorical sequence can not be consumed by many feature extraction techniques directly. One example is given in the work of Eagle and Pentland [58]. In order to study "eigenbehaviors" of users, location trajectories were represented as a D by 24 two-dimensional array, where D is the number of days in data, and 24 is the hours of a day. Each cell in the matrix contains a location label (e.g., home, work, others, no signal). This data matrix is then binary encoded to become a D by H two-dimensional array, where H is 24 times the number of location labels (in this case, number of location labels is 5). This encoding scheme is shown to be generalizable to other time series data including phone usage and number of people in proximity. With such encoding, the data matrix can be transformed into "eigenbehaviors" using principle component analysis (PCA) to identify most significant behavior structure in smartphone users' routine lives. A similar encoding procedure is also applied in [212] to represent hourly activity level, step counts, and phone usage times across different days as data matrix, and a technique called robust PCA was used to decomposed these data matrices into components that represent the underlying routine patterns, and variations of these routines.

Another example is by Wu et al. who proposed a bag-of-word representation in bluetooth encounter data for feature extraction to recognize stress levels [234]. The same authors also proposed to construct encounter networks in different graphs for feature extraction to recognize current stress level or predict future stress level [233].

A number of properties for good encoding have been identified: smoothness, temporal and spatial coherence, sparsity, natural clustering, consistency (i.e., similarity in the representation space should reflect the similarity of the corresponding concepts), independence (i.e., easy to obtain in the absense of external information), imputable (i.e., possible to fill-in when missing based on the observed ones). [19, 12] When consider which encoding schemes to apply, the above criteria, especially temporal and spatial coherence, should be taken into consideration, given that mobile sensing data stream are inherently temporal, and can be placed into its spatial contexts if GPS data is available.

We have identified three different popular encoding schemes: identity, one-hot, and label encoding. In identity encoding, the original sensing stream is kept as it is and fed into any feature extractors. For example, in [24], the preprocessed PPG signals were applied as the input to deep learning models for activities recognitions. In one-hot encoding, categorical data stream such as place visits is converted into 0 and 1 representations [58] for feature extraction. Label encoding is similar to one-hot encoding, but converts categorical data into more than two numeric values. Numeric data sequence usually applies identity encoding such as those in [24, 143], while categorical data sequence uses one-hot and label encoding instead [58]. However, it is not always clear how to preprocess data before encoding them. For example, in [24], the original PPG signal was first decomposed into cardiac, respiration, and motion artifact signals, and the concatenation of the three components in their identical form was used as the input for deep learning models; while in [143], location displacements and time spent in significant places were used as input for auto-encoder to learn compact feature representation.

Featurization

After preprocessing and encoding, the resulted data will undergo the featurization process to generate the desired features for behavior modeling. There are two broad categories of methods for featurization: 1) domain-based methods and 2) domain-free methods according to whether domain knowledge is required during the feature extraction process.

Domain-based Methods. Domain-based methods usually require specific domain knowledge (e.g., health and behavior theories, and mathematics) on the targeted behaviors (e.g., understanding the impacts of physical activity on stress, or the motion characteristics of ambulatory activities) being modeled to guide the design and extraction of features. We group domain-based methods into theorized and heuristic methods, statistical methods, frequency-domain methods, and pattern mining methods based on the analytical techniques involved in the feature extraction process.

Theorized and Heuristic Methods. One set of features can be time of day, day of week, and month of year. Intuitively, these temporal features definitely have certain impacts on our behaviors as our biological/circadian rhythm governs our daily behaviors (e.g., waking up and sleeping at certain times of a day) [1], and our social rhythm determines our broader behaviors in weekly and even monthly/yearly cycles (e.g., work during week days and off work on weekends, study during the semester and on breaks in between semesters). [224] However, There are more than one way to represent these features, and choosing among the options can depend on the behaviors one is modeling. For example, time of day can be represented as the hour of day (0-23) or as several time buckets (morning/noon/afternoon/evening/night). Another example is from the work of Canzian et al. [39]. They proposed eight different features based on GPS coordinate sequences from participants, including total distance, maximum distance between two locations, radius of gyration, standard deviations of displacements, maximum distance from home, number of different places visited, number of different significant places visited, and routine index. In [223], basic contextual information are combined to form heuristic features that represent more social behaviors. For example, being in study areas or classrooms and static with no phone usage can be considered as being concentrated on studying or paying attention in class.

Some features are extracted based on existing human behavior theories, including those related to social and health behaviors. [155] A few examples include social cognitive theory, planned behavior theory, and health belief model [73]. In mental health, manifestations of disease symptoms such as depression, social anxiety, bipolar disorder, and schizophrenia have been extensively studied. This knowledge can guide feature extraction. For example, researchers in [225] mapped their feature extraction to the major depressive disorder symptoms defined in the standard mental disorders diagnostic manual (DSM-5), and proposed a set of features that approximate the depression symptoms for college students using mobile sensing data. These features include sleep, phone usage, place visits, physical activities, and sociability.

Statistical Methods. Statistical methods extract features from segmented time series by applying various statistical functions directly on the data. For example, mean and standard deviations are computed for continuous data to describe the centrality and disperse of the segmented data. Other popular statistics for continuous data include but are not limited to percentiles, minimum, maximum, skewness, kurtosis, auto-correlation, correlation and regression coefficients. [56] When the input data segments are categorical (including binary), frequency of each category is the usual choice of method (e.g., [123, 187]).

Frequency-domain Methods. Frequency-domain methods transform segmented continuous time series into their frequency domain using FFT or WT, and extract features related to the various frequency components. For example, in [121], the authors applied WT to extract features from acceleration sequence to predict ambulatory activities (e.g., being static, walking, running, ascending stairs, descending stairs, cycling, jumping). In [187], FFT was applied to extract frequency components that maximized the amplitude as features from audio data to infer the semantic label of location. FFT and WT are mathematical transforms that are powerful to interpret numeric time series data, in which the frequency content is more informative than the original signal. It is applicable to sensing data from several popular smartphone embedded sensors including accelerometer and microphone. Note that both FFT and WT are computationally expensive, and thus may not be practical for real time or resource constrained applications. [66] provides some detailed formulas of frequency domain features for accelerometer data, and most of these features can be generalized to other numeric sensing sequences.

Pattern Mining Methods. Most behavior inference works apply supervised learning algorithms to infer physical activities or mental states, and then use these inferred activities and mental states as features to understand and infer more complex behavioral constructs. In contrast, pattern mining methods apply unsupervised learning techniques to understand and infer significant behavior patterns, which can also serve as features for inference of more complex behaviors. In [92], the authors proposed to apply Latent Dirichlet Allocation (LDA), a topic modeling technique, to infer activity patterns in people's daily life. Specifically, the LDA algorithm takes as input discrete labels that are generated by K-means clustering on the feature vectors extracted from continuous sensor data, and outputs the topic models that are able to provide the activation of a given learned topic at a certain time of day. In this way, high-level routines such as commuting, working in the office, having lunch or dinner etc. can be inferred throughout the time of a day. Ye et al. proposed an individual life pattern mining framework that can learn frequent life patterns using daily place visitation sequences. [240] Liu et al. applied association rule mining technique to learn complex high-level activities (e.g., relaxing, coffee time, cleaning) from low-level actions. [130]

Domain-free Methods. In contrast to domain-based methods, domain-free methods require no feature design using domain knowledge. Instead, engineers only need to choose a data representation to encode the data, and feed them into some automatic feature extractors (e.g., PCA and auto-encoder) to learn certain informative features. This process entails two steps, an encoding step, and a feature extracting step. The encoding step follows the data encoding section.

In the feature extracting step, automatic feature extractors such as PCA and auto-encoder can be applied to obtain a more compact representation of the data as features for behavior modeling. These automatic feature extractors are indeed dimension reduction techniques that can both reduce the size of the data representation and discover features (i.e., latent factors) that can be both predictive and interpretable of the targeted behaviors. Existing works applied various dimensional reduction techniques such as PCA [58, 171], linear dynamical system [76], and auto-encoder [143, 171]. In [58], the authors applied PCA on the tempo-spatial binary encoded location data to learn "eigen-behaviors" that reveal repeating routine structures in daily lives. These eigen-behaviors can be applied as features to both understand and infer other higher level constructs such as personality and lifestyle. In [171], the ECDF representations of frame-based accelerometer time series were provided to the PCA algorithm to obtain the top 30 eigenvectors as features. The authors also evaluated using auto-encoder method to learn a low-dimensional feature representation from the same data for activity recognition. They showed that the ECDF+PCA method outperformed all other methods. Mehrotra et al. provided as input three different representations of GPS coordinate trajectories including displacement, change in displacement, and significant place, to train an encoder and decoder for each of these three representations, and extract features using the trained encoders to predict depressive state in study participants. [143] A more recent work from Gong et al. proposed to use Linear Dynamical System to extract features from frame-based raw accelerometer time series to study social anxiety symptoms during phone calls and text messaging. [76]

For accelerometer data, a class of distribution-based feature extraction methods were proposed to represent accelerometer time series in the traditional sliding window procedure for human activity recognition (HAR) tasks [243, 79, 110]. These methods can also be viewed as dimension reduction methods as they choose a set of percentiles to represent the encoded data sequences. Specifically, the authors in [243] discretized the preprocessed time series into a selected number (B) of bins, converted the original time series values into the discrete values for normalization, and use the B frequency counts from the resulted histogram as features to represent the original signal. This approach was proved to be effective in reducing the dimensionality and improving the generalizability of the feature vector. Plotz and Kwon et al. proposed a method called empirical cumulative distribution function to extract structure-based features to infer human activities. [79] Specifically, the cumulative density function is calculated for each frame of raw sensing data, and the N quantiles are calculated as an approximation for the distribution. The inverse mapping of these quantiles are then extracted as the final predictive features. To further preserve the temporal structure information within each frame of sensing data, the authors proposed several structure embedded techniques for the ECDF representation, and showed additional improvements in predicting various human activities. [110]

In recent years, artificial neural networks (ANN) or deep learning (DL) models have been successfully applied in many domains such as vision and image recognition, natural language processing (NLP), robotics, and adversarial games. [119] However, these domains often provide large amount of labeled data for training. In behavior modeling, labels are extremely expensive to obtain and error-prone. Some activity recognition works such as [170, 24] have applied deep learning techniques to recognize ambulatory activities. In [180], a multimodal deep learning model was proposed as a domain-free method. However, the effectiveness of these techniques requires verification in tasks beyond HAR. The unified deep learning framework proposed in [239] have shown some effectiveness in mobile sensing tasks and its feasibility in implementation on mobile devices.

3.3.3 The Output

The generated features are usually low-level, and can be combined to represent high-level characteristics in human behaviors. When ground truth labels are available, machine learning algorithms are often adopted to develop supervised models for activity and context recognition [101, 126] or mental health prediction (e.g., stress and social anxiety) [226, 48, 187].

The target behaviors we attempt to model have several implications on feature extraction. First, we need to choose the most effective sensing modality, and collect data from it to model the target behavior. Data from different sensors will require different feature extraction techniques. Second, the target behavior determines what features from a sensing stream are most effective. For example, different feature sets from GPS would be extracted for transportation mode inference and for depression prediction. Last and most importantly, different behaviors have different temporalities, and thus require different segmentation and aggregation methods in feature extraction. For example, ambulatory activities such as walking and being still, and transportation modes such as driving a car and biking are both heavily dependent on momentary features that characterize physical states of the user (e.g., mean and standard deviation of acceleration, speed). In modeling these behaviors, how many places you visit yesterday or a week ago may not add much predictive power to the model. In contrast, how much time one spent at home on the day or in the past week has been proven significant in predicting daily mood [48] or social anxiety [28].

In summary, the target behavior determines the selection of sensing modality, the features , the segmentation and aggregation methods in the feature extraction process. We propose to follow the below steps using our feature extraction framework while considering the target behavior to model: 1) choose segmentation methods; 2) follow the proposed

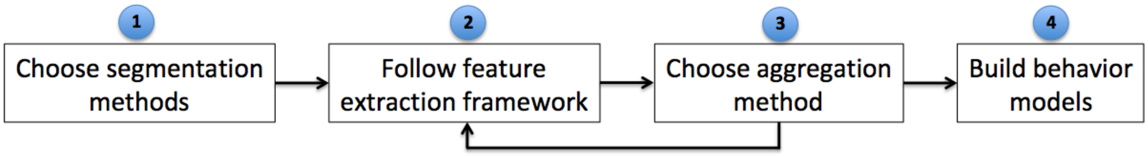

FIGURE 3.2: Behavior Modeling.

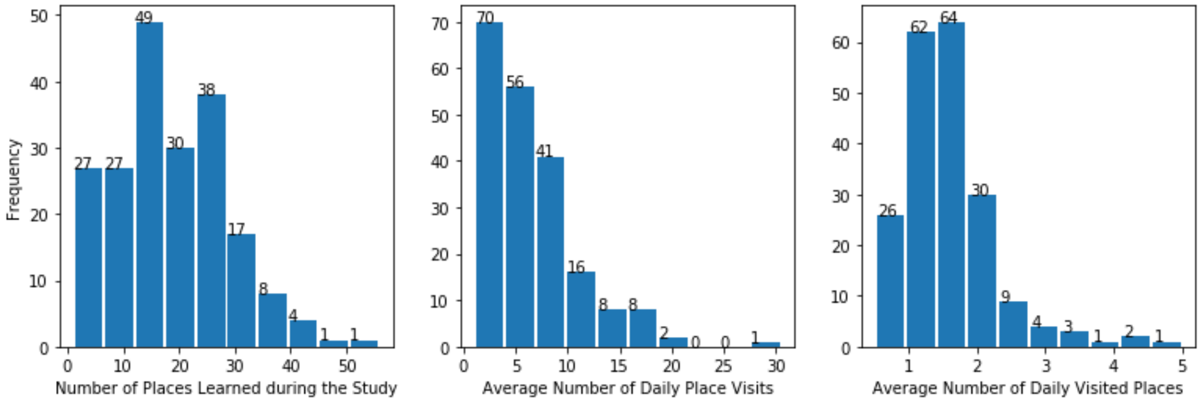

FIGURE 3.3: Distributions of (a) number of places, (b) average number of daily place visits, and (c) average number of daily visited unique places.

feature extraction framework; 3) determine aggregation method; and 4) build behavior models. Figure 3.2 illustrates this process.

3.4 Online Algorithms for Location Learning

Extracting location information from raw GPS data provides spatial contexts to model user's behaviors. We conduct a comprehensive review on various location clustering algorithms in this chapter. Among them, we choose Kang's temporal spatial clustering algorithm [97] for our works in this dissertation, and adapt it into an online version for simulations. Algorithm 3 describes this algorithm. $\boldsymbol{GPS}$ are the set of newly available raw GPS data, $\boldsymbol{Places}$ contains all the learned unique places that are being continuously updated over time. When a new place visit (i.e., a new cluster that meets both the time and distance thresholds) has been detected, $\boldsymbol{Places}$ will be scanned and updated. Either a new place that has never been visited before will be added to $\boldsymbol{Places}$ or a new visit will be added to an existing place in $\boldsymbol{Places}$. We apply it to a real GPS dataset from 200 college students in 10 minutes increments. Figure 3.3 visualizes the distributions on number of places, average number of daily place visits, and average number of daily visited unique places.

Algorithm 3 Online Temporal-spatial Clustering Algorithm adapted from [97]

Input: $\boldsymbol{Places}$, $\boldsymbol{cluster}$, $pgps$, $\boldsymbol{GPS}$, $t_{threshold}$ $d_{threshold}$.
Output: $\boldsymbol{Places}$, $\boldsymbol{cluster}$, $pgps$.

for $gps \in \boldsymbol{GPS}$ **do**
  if $\boldsymbol{cluster} \neq$ None or $dist(\boldsymbol{cluster}, gps) \leq d_{threshold}$ **then**
    $\boldsymbol{cluster}.update(gps)$
    $pgps =$ None
  else
    if $pgps \neq$ None **then**
      if $\boldsymbol{cluster}.duration \geq t_{threshold}$ **then**
        $\boldsymbol{Places}.update(\boldsymbol{cluster})$
      end if
      $\boldsymbol{cluster} =$ None
      $\boldsymbol{cluster}.update(pgps)$
      if $dist(\boldsymbol{cluster}, gps) \leq d_{threshold}$ **then**
        $\boldsymbol{cluster}.update(gps)$
        $pgps =$ None
      else
        $pgps = gps$
      end if
    else
      $pgps = gps$
    end if
  end if
end for
return $\boldsymbol{Places}$, $\boldsymbol{cluster}$, $pgps$.

3.5 Conclusion

Feature extraction in mobile sensing is a critical step towards modeling users' contexts and behaviors. In this chapter, we review existing works, and based on their methodologies distill a feature extraction framework that helps the mobile sensing community standardize this pre-modeling process and better understand what they should be considering in each step along the behavior modeling pipeline.

Chapter 4

Energy Efficient Adaptive Mobile Sensing Using Q-learning with Linear Approximation and Decaying Exploration (QLADE)

Smartphone embedded sensors have created unprecedented opportunities to study human behavior in natural conditions through continuous mobile sensing. However, continuous mobile sensing poses critical energy challenge to smartphone's daily usage. There is an urgent need to enhance energy efficiency of mobile sensing applications while capture sufficient data to accurately predict user state. In this work, we propose an adaptive passive sensing framework to control low-level sensing cycles using an off-policy reinforcement learning algorithm called Q-learning with Linear Approximation and Decaying Exploration (QLADE). We formulate the adaptive sensing problem as a middle ground between continuous sensing and duty cycle, and investigate the trade-off between energy efficiency and activity coverage. Our simulations using real continuous mobile sensing data from 220 participants for more than 2 weeks show consistently better performances for the proposed QLADE algorithm when compared to the random and learning automata baselines for both accelerometer and GPS. We also investigate whether the adaptive sensing strategies have significant impacts on the utility of data. Given the original data

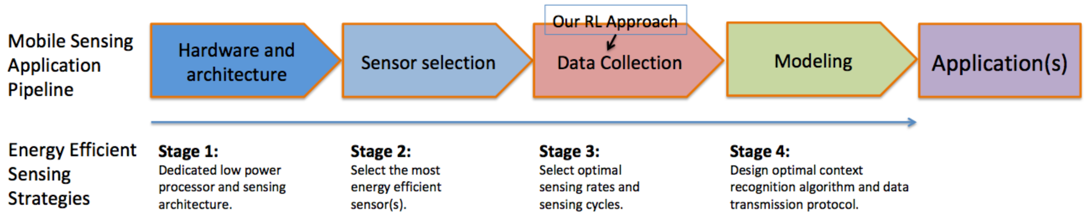

FIGURE 4.1: Mobile sensing pipeline for energy efficiency.

were collected to understand and predict mental health states from passive sensing data, results suggest that the proposed approach do not compromise the performance of our predictive models.

4.1 Introduction

Smartphones have become an integral part of our routine lives. According to the Pew Research Center [205], 81% of North American adults owned a smartphone in 2019. With increasing access to wireless networks, better display technology, and improved computation power, smartphones are used to surf the internet, navigate during driving, pay bills, and socialize with friends etc. The addition of embedded sensors, such as accelerometers, light sensors, GPS, cameras, microphones, and gyroscopes, enables them to capture fine-grained digital footprints of users, making them a popular tool to study human behaviors and well-being [29, 28]. The context-awareness created by performing machine learning algorithms on the sensor data can be leveraged in a variety of intelligent applications, including location services (e.g., booking a taxi at the user's location), healthcare (e.g., monitoring physical exercises), safety (e.g., checking young kid's whereabouts for parents), and mobile crowd sensing (e.g., monitoring traffic congestion and air quality) [137].

Many of the aforementioned applications require continuous sensing, which quickly drain the smartphone battery, and interrupt other services on the device. [235, 112, 181] Given the competing demands for energy among an increasing number of services, and the power intensive nature of mobile sensing applications [144], it is critical to enhance energy efficiency in these mobile sensing applications.

Many approaches have been proposed to address the energy challenge posed by mobile sensing. These methods target energy expenditure arising from various stages during the mobile sensing application pipeline, which consists of a hardware and architecture design stage, a sensor selection stage, a data collection stage, and a modeling stage, as is shown in Figure 4.1.

- **Stage 1: Hardware and Architecture Design.** The authors in [173, 132, 175] designed dedicated hardware and system architecture that are energy efficient for continuous mobile sensing applications. Without using these dedicated hardware, the smartphone main processor and its associated components will remain active for an excessive period of time, leading to significant energy overhead. Dedicated low power processor for sensing and data processing with novel sensing architecture can significantly reduce the energy overhead, and prolong the battery life.

- **Stage 2: Sensor Selection.** Many smartphone embedded sensors with different energy rate can be leveraged to achieve a common task with different accuracy and under certain conditions latency trade-offs. For example, GPS and WiFi sensor can both be leveraged to recognize user location, but are best for localization in different environments. [124, 249, 189] Accelerometer and GPS can be combined or independently applied to recognize transportation mode (e.g., walking, in a Vehicle, biking). The right combination of sensors that are most energy efficient can be selected to meet the application requirements in accuracy and latency.

- **Stage 3: Data Collection.** Prior to collecting sensor data, we need to specify the sensing rate and sensing cycle for each chosen sensor. Higher sensing rate and longer sensing time result in denser data but consume more energy. Existing works such as those in [238, 102, 227, 29] propose methods to identify optimal sensing rate and reduce sensing time in an attempt to reduce energy consumption.

- **Stage 4: Modeling.** Once collected, sensor data are used to create predictive models for context recognition. Different algorithms can achieve different accuracy, latency, and energy consumption levels. Existing works aim to select the optimal algorithm that is most energy efficient [49, 113]. When data transmission is necessary prior to modeling, it is also critical to design an energy efficient data transmission protocol to minimize battery drain [137, 144]. For example, the authors in [219, 167, 144] proposed efficient data transmission frameworks for mobile crowd sensing (MCS) and personal sensing tasks.

Methods targeting different stages in the mobile sensing application pipeline have attained various levels of success, but often require offline tuning and training, and using external database in a well defined sensing task, making them hard to generalize across mobile sensing applications [11]. It has also been shown that hardware sensing consumes the majority of energy, with the only exception for applications that require transmission of large amount of data from local device to central server [144]. In view of these, an energy efficient sensing approach that is free of prior tuning and external dependency, easily generalizable, and efficient in hardware sensing is desired.

To meet these requirements, we propose an adaptive passive sensing framework, in which information obtained from small sensing window is leveraged to determine deployment of sensors in the following preset time window. In fact, existing works have also proposed to leverage certain sensed contexts (e.g., movement, current battery level, and being indoors) to activate power-consuming sensors (e.g., GPS) [15, 150, 30]. Unlike these works, we do not rely on a predefined state, upon which the decision on sensor deployment is made. Instead, we generalize the state into a state feature vector, and design a reinforcement signal to guide the action decision (i.e., turning the sensor on/off). This framework is known as reinforcement learning (RL) [203].

Our proposed reinforcement learning approach has several advantages. First, it can be combined with most existing approaches targeting energy challenges in different stages of the mobile sensing application pipeline to further mitigate the battery drainage problem in mobile sensing applications. For example, an energy efficient mobile sensing application could apply dedicated low power processor in a more efficient sensing architecture, select the most suitable sensing modalities, configure the sensing rates based on the needs of the applications, design an optimal data transmission framework, and deploy sensors adaptively using our proposed method. Second, the learned sensing strategies are personalized for each user. Under the RL framework, the sensing agent will learn a unique sensing protocol for each user using their own data. Third, the learned sensing strategies are adaptive over time. The RL agent continues to update each individual protocol as more data are being collected and made available for learning. Lastly, RL is a natural fit for adaptive sensing due to its learning through interaction with the environment. Static sensing protocol may be compromised due to changes in human behaviors. Adaptation to both within and between individual variability can potentially lead to long term energy efficiency in mobile sensing applications.

We formulate the adaptive sensing problem as a middle ground between continuous sensing and duty cycle, and propose an off-policy reinforcement learning algorithm called Q-learning with Linear Approximation and Decaying Exploration (QLADE) to investigate the trade-off between energy efficiency and activity coverage. Accelerometer and GPS are two highly representative smartphone embedded sensors, and have been applied in many existing mobile sensing applications. Continuous deployment of these sensors are both energy demanding and unnecessary, especially when the user is away from the phone or being static. Using accelerometer and GPS as examples, we implement two adaptive sensing schemes single and multi-modality QLADE, and compare them with two baseline strategies, including a random strategy and a strategy using learning automata technique [179]. The strategies are evaluated using four metrics: 1) energy saving (i.e., the percentage of time when there is no movement or displacement, and the sensors are turned off); 2) accuracy (i.e., the percentage of time when our proposed approach correctly predicted existence or absence of movement or displacement); 3) activity coverage (i.e., the percentage of time when movement or displacement is correctly predicted by our proposed approach); and 4) F-score (i.e. a comprehensive score reflecting the balance between prediction precision and recall). We also investigate whether the different sensing strategies (e.g., continuous sensing, duty cycle, and adaptive sensing) have significant impacts on the data utility (e.g., classification and regression accuracy), given the original data were collected for the purpose of understanding and predicting negative affect and social anxiety of college students [28, 52, 53].

The remainder of the chapter has the following structure. Section 4.2 reviews related work in energy-efficient mobile sensing. Section 4.3 formulates adaptive sensing. Section 4.4 explains the details of the proposed methodology. Section 4.5 provides the experimental setup while Section 4.6 presents performance evaluation of our proposed adaptive sensing algorithm QLADE. We discuss the limitations of our current work and lay out future work in Section 4.7, and make our conclusion marks in Section 4.8.

4.2 Related Works

Many strategies have been proposed to address energy challenges in various stages throughout the mobile sensing application pipeline illustrated in Figure 4.1. In this section, we focus mainly on works that are highly relevant to our proposed method. These works can be grouped into three categories based on their methodologies: 1) methods with pre-calibration in defined conditions; 2) static context-based methods; and 3) adaptive sensing methods.

4.2.1 Methods with Pre-calibration in Defined Conditions

Yan et al. obtained a set of optimal configurations in sensing rate and classification features (e.g., time and frequency domain features) with respect to classification performance, and proposed an algorithm called A3R that works by transitioning between activity states based on the predicted confidence, while adopting the pre-calibrated optimal configuration in the corresponding predicted state [238]. Wang et al. proposed a framework called EEMSS that applied a state descriptor, which specified predefined states and state transitions, to deploy only the minimally required set of sensors in the corresponding user state. [227] They also used pre-tuned duty cycling intervals to further improve energy efficiency.

Kansal et al. proposed the LAB (latency, accuracy, and battery) abstraction that aims to provide a generalized framework for energy efficient mobile sensing application development [98]. Specifically, they implemented a set of context recognition algorithms, and evaluated their corresponding accuracy, latency and energy consumption performance. Algorithms that have no merits in all three measurements will be discarded. With the remaining algorithms, application developers just need to choose priorities in latency, battery, and accuracy, while the LAB framework will choose the optimal algorithm that meets these priority constraints for the developers. Another work by Cardone et al. designed the Mobile Sensing Framework (MSF) for adaptive duty cycling [42]. The adaptation is embedded inside the interaction layer, which acts on a strict event-action basis and supports arbitrary events. Users can leverage simple duty-cycle policy or design more complex policies that controls the sensing cycles of different sensors.

The above works share the same major drawback of being application dependent, and thus can not be easily generalized across different mobile sensing applications. In [238], the optimality of the sampling rate and classification features may not hold in a different classification task. In [227], the state descriptor will need to be redefined, while the pre-tuned duty cycling intervals do not warrant improvements in energy performance in a different application scenario. Similarly, Kansal's LAB abstraction [98] is entirely dependent on the context recognition tasks to obtain a set of strategies that trade-off among latency, accuracy, and energy efficiency; and the MSF framework [42] requires the design of events as triggers to control sensing cycles. In contrast, our proposed adaptive sensing scheme does not require pre-calibrations of energy performance in defined conditions that are application dependent.

4.2.2 Static Context-based Methods

Ben et al. presented the SenseLess system, in which they leveraged accelerometer to continuously provide contexts for the decision of activating GPS, thereby reducing energy consumption by more than 58% as compared to when GPS are continuously activated. [15]

Specifically, they chose a threshold for acceleration to indicate user movement, and followed the following rule: when the user's acceleration is beyond the chosen threshold, location sensing is turned on; when the user's acceleration is below the chosen threshold for three consecutive readings, the location sensing will be turned off. Similarly, Oshin et al. predicted users' mobility state using accelerometer to ensure activation of GPS only when the user is moving [161]. Their rules stated that 1) when the sum of the total peaks and troughs is greater than 3, and the difference of the max and min values is greater than 1.4, enable GPS; 2) else keep GPS sensor idle. They found a 27% energy saving in typical circumstances.

Li et al. proposed to leverage machine learning to predict the state of energy demanding sensors such as GPS [118] using light-weight sensors. The intuition is that data collected at the same moment from different sensors can be highly correlated. If the inference indicates that the energy demanding sensor is in a stable status, then the latest value can be carried forward without actually activating them, leading to reduction in energy consumption. Kim et al. proposed an adaptive WiFi scanning algorithms to conserve energy consumption through replacing mobile network usage with WiFi usage [103]. Specifically, accelerometer data are applied in a window of few seconds to classify movement activity (e.g., standing, walking, and running), and calculate movement distance. If distance is above a chosen threshold, then WiFi scanning will be triggered. This will improve WiFi usage while optimizing scanning rate.

Zhuang et al. built an adaptive location-sensing framework to improve the energy efficiency in location-based applications. [249] The proposed framework included four design principles, namely Substitution, Suppression, Piggybacking, and Adaptation, each of which aims to optimize sensing mechanism, sensing timing, sensing rate, and duty cycling intervals, respectively. In particular, the Suppression principle utilizes accelerometer sensor to provide contexts for invocation of location sensing. Paek et al. proposed RAPS, a rate-adaptive positioning system to reduce power consumption of location sensing by determining the GPS activation timing using: 1) duty-cycled accelerometer; 2) space-time history of velocity and its associated uncertainty from GPS data; 3) cell tower RSS blacklisting ; and lastly 4) position uncertainty reduction inferred by Bluetooth data. [162]

All works in this section proposed static rules in different formats using sensed contexts from certain smartphone sensors towards energy management in mobile personal sensing. In particular, [15, 161, 118, 103] leveraged a less power demanding sensor (e.g. accelerometer) to control activation of a power hungry sensor (e.g., GPS and WiFi). These static rules leverage predetermined acceleration thresholds or machine learning models, and can not account for between and within individual variability. The authors in [249, 162] proposed a suite of strategies that are designed only for location sensors such as GPS and WiFi sensor. In addition, some components in these approaches are also based on static rules. Compared with these works, our proposed RL approach is both personalized and adaptive.

4.2.3 Adaptive Sensing Methods

Lu et al. designed the Jigsaw engine, which consists of three sensor-specific pipelines for accelerometer, microphone, and GPS. [134] Of particular interests are the microphone and GPS pipelines, which adapt their duty cycling intervals based on various techniques. In addition, the GPS pipeline adjusts the sensing rate by treating it as a Markov Decision Process (MDP) using sensing duration, hardware status (e.g., remaining battery budget), and mobility as its state features. Similarly, Krause et al. proposed an entropy-based strategy that models the sensing problem as a MDP using the predicted activity as state to determine the next sensing timing [109]. Wang et al. investigated a deterministic sensing policy, which chooses different duty cycles based on different user states to conserve energy [228]. Specifically, they modeled the context recognition problem as a discrete time hidden Markov chain. At each time step, a sensor activation decision based on the recognized context is made, followed by a deactivation duration decision when the sensor is deactivated. In this case, the contextual state is estimated with uncertainty levels. These works applied model-based RL approaches, in which the transition model have to be estimated using annotated or predicted state labels. Since these approaches are

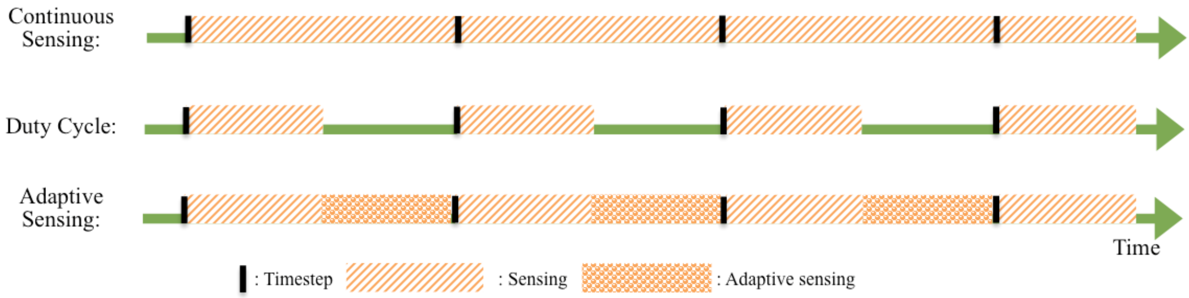

FIGURE 4.2: Continuous, duty cycle, and adaptive sensing.

dependent on availability of activity labels and/or sensing rates, we do not include them as our baseline comparisons.

Rachuri et al. proposed an adaptive sensing scheme based on a set of advance and back-off functions in controlling the sensing rate. [178] At each time step, data generated by the targeted sensor will be used to determine whether a 'missable' or 'unmissable' event is detected. When a missable event is detected, the selected advance function is applied to expand the sleep interval; when an unmissable event is detected, the back-off function is applied to shrink the sleep interval. Due to the different types of sensors, this method may not be applicable to sensors such as accelerometer and microphone. However, it can be generalized to controlling sensing cycles in a different adaptive sensing formulation. The authors investigated linear, quadratic, and exponential functions, and various combinations of them as both advance and back-off functions. The same authors also proposed an adaptive sensing strategy using learning automata to adaptively increase or decrease sensing probability at each time step [179]. More details about this algorithm will be provided in Section 4.5 as it is chosen as our baseline comparison. Comparing to our proposed approach, the learning automata algorithm does not leverage the knowledge about contextual state extracted from the collected sensor data.

4.3 Adaptive Sensing

Duty cycling is a widely adopted strategy to reduce energy consumption in mobile sensing applications by making the desired sensors operate and rest alternatively within the designated sensing and sleeping intervals. Figure 4.2 illustrates the duty cycle strategy and the continuous sensing strategy. By reducing the sensor operation time, duty cycle can save a significant amount of energy. However, important contexts may be missed when the sensors are "turned off" during the sleeping interval. For some applications, this critical information may lead to failure in providing necessary responses (e.g., triggering a recommendation to restaurants) to those contexts (e.g., walking in a business area around lunch time), rendering the applications less useful in serving their users. In some extreme cases, such as elderly and patient monitoring, the lack of responses due to the absence of data can not be tolerated. Therefore, we need to balance energy efficiency and application requirements by using less aggressive sensing strategies.

Adaptive sensing is the middle ground between continuous sensing and duty cycle as illustrated by Figure 4.2. It leverages the data collected during the sensing window to make sensor deployment decision in the adaptive sensing window. The intuition is that human activities oftentimes come in bouts and last for a certain duration. The signal from earlier times can provide information about the activities in the next few moments, therefore helping us determine sensor deployments. An 'off' decision will help us save energy, while an 'on' decision will help us capture the activities of interest, balancing energy consumption and application requirements.

Adaptive sensing can be treated as a sequential decision problem, and simplified using Markov decision process (MDP). Time is first discretized into steps, within which the sensor(s) operates for a configured window (e.g., sensing window), and a decision regarding sensor deployment is made for the remaining window (e.g., adaptive sensing window). The sensors will operate based on this decision during the adaptive sensing window, and the cycle repeats indefinitely. Within each time step, the action decision is made based on the state that is constructed using the sensed data collected within and prior to the current time step.

Based on the above problem formulation, a wide range of algorithms can be applied to implement adaptive sensing. In Section 4.2, we have surveyed numerous existing methods that fall into the adaptive sensing literature. In this work, we build upon our previous work [35], and show that our proposed adaptive sensing method using reinforcement learning has the potential to advance mobile sensing by reducing the operation time of various sensors, thereby saving energy, while maintaining equivalent application utility.

In general, our proposed adaptive sensing framework using reinforcement learning has the following contributions: 1) It can be generalized across different sensing modalities and applications; 2) it is complementary to most existing approaches (e.g., it can be easily combined with hardware-based or computationally efficient approaches); and 3) using reinforcement learning framework as the basis, the learned strategies are personalized and adaptive.

It is worth noting that although the current formulation posits our method as the middle ground between continuous sensing and duty cycle, it is possible to design a reinforcement learning adaptive sensing approach that can minimize the probability of not covering important contexts, while also become more energy efficient than duty cycle strategy. We will discuss more on this in Section 4.7. In the next section, we will provide the technical details about the implementation of our proposed reinforcement learning adaptive sensing algorithm, and the performance metrics for evaluation.

4.4 Methodology

In this section, we first describe how we apply reinforcement learning to adaptive sensing in two most representative smartphone embedded sensors accelerometer and GPS; then we present the RL algorithm, namely Q-learning with linear approximation and decaying exploration (QLADE) in two sensing schemes – single and multiple modality adaptive sensing; and lastly, we define the three RL components for adaptive sensing in both accelerometer and GPS, and the performance metrics that we apply to evaluate our proposed adaptive sensing strategy in the experiments.

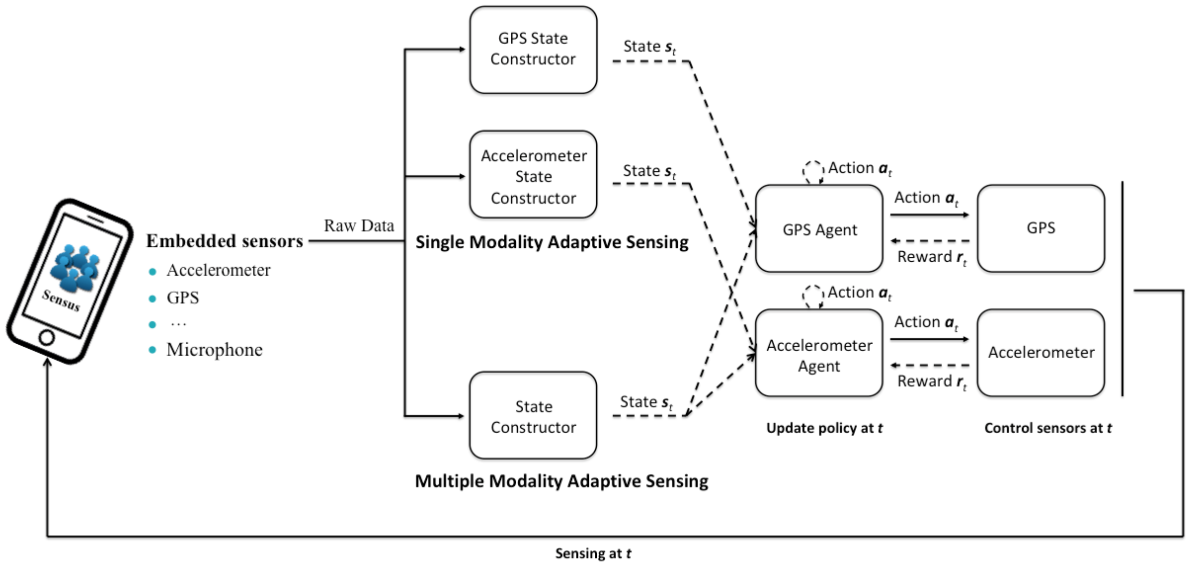

FIGURE 4.3: Single and Multiple Modality Adaptive Sensing Using Reinforcement Learning

4.4.1 Adaptive Sensing Using Reinforcement Learning

Figure 4.3 illustrates the workflow of our proposed adaptive sensing strategy. For each time step, the time window is divided into two sub-windows, a sensing window, and an adaptive sensing window, as shown in Figure 4.2. During the sensing window, the selected sensors are activated to provide raw sensor data for the state constructors to extract state features for sensing decision in the adaptive sensing window. There are two designs in the configuration of state constructors, single modality and multi-modality methods. In the single modality method, each sensor has its own state constructor, which leverages only data from the corresponding sensor to construct the state features. In contrast, the multiple modality method shares one state constructor for all sensors, and combines data from all the sensors to construct the state features. These state features are then sent to the various agents to inform sensing decision based on the most up to date policy. Once the sensing decision is made, the sensors will be deployed based on it, and a reward signal about the desirability of the sensing decision will be provided to the agents. At that point, the agents will leverage the state s_t, action a_t, and reward r_t information to update the policy. This cycle repeats for each time step. In the next section, we provide the technical details on our proposed RL algorithm, namely Q-Learning with Linear Approximation and Decaying Exploration (QLADE), which is applied to update the sensing policy in each time step.

4.4.2 Q-Learning with Linear Approximation and Decaying Exploration

There are two types of learning methods in RL, on-policy and off-policy methods. On-policy methods refer to learning algorithms that share the same policy for sample generation and learning, as compared to off-policy methods, in which the sample policy and the learning policy are different [203]. Q-Learning is an off-policy temporal difference (TD) control algorithm, defined by

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)],$$

where $Q(S_t, A_t)$ is the action value function that represents the policy. When all state features are discrete, $Q(S_t, A_t)$ is a high dimensional table that contains values for all state-action combinations. When some state features are continuous, the tabular format can not represent all possible state-action combinations. In this case, we apply function approximation to represent the action value function and enable generalization across different states. Specifically, we apply linear approximation to represent $Q(S_t, A_t)$ as a linear function given by

$$Q(S_t, A_t) = \Phi(S_t, A_t)\boldsymbol{w_t},$$

where $\Phi(S_t, A_t)$ is a feature vector, and $\boldsymbol{w_t}$ is the learned coefficients that encode the learned policy. [203]

In evaluation, continuous sensing stream is sliced into windows indexed by t. Each window will be further broken down into two sub-windows – the state feature window and the evaluation window. The state feature window corresponds to the sensing interval, while the evaluation window corresponds to the adaptive sensing interval plus the sensing interval in the next time step, which is necessary for estimating the reward when the sensor is turned off within the adaptive sensing interval. At each time step, s_t will be computed based on the raw sensing data from the state feature window.

Algorithm 4 shows our proposed ϵ-greedy Q-learning algorithm using linear approximation and decaying exploration ϵ_t with a decaying rate $d \in (0, 1)$ for adaptive sensing. At each time step, the Q-learning strategy will either randomly select one of the two sensing actions (switching sensor on and off) with a probability of ϵ_t (e.g., exploration) or choose the action that returns a higher Q-value based on the learned action value function $\Phi(S_t, A_t)\boldsymbol{w_t}$ (e.g., exploitation) with a probability of $1 - \epsilon_t$. Specifically, ϵ_0 is the initial exploration rate, α, γ, λ are the step-size (learning rate) parameter, the discount rate, and the eligibility trace-decay parameter, respectively.

Algorithm 4 Adaptive Sensing Using Q-learning with Linear Approximation and Decaying Exploration.

Input: $\boldsymbol{S}, \boldsymbol{A}, \gamma, \lambda, \alpha, \epsilon_0, d$.
Output: $\boldsymbol{w}^a$, $a \in \boldsymbol{A}$.

```
Initialize w^a and e^a for each a ∈ A.
for all t = 1, 2, ..., T do
    Observe current state s_t
    Take action a_t based on ε_{t−1}-greedy policy with regards to arg max_{a∈A} Φ(s_t, a)^T w^a.
    Observe reward r_t
    Transition to a new state s_{t+1}.
    Take action a_{t+1} based on ε_{t−1}-greedy policy with regards to arg max_{a∈A} Φ(s_{t+1}, a)^T w^a.
    e^{a_t} = e^{a_t} + Φ(s_t, a_t)
    δ_t = r_t + γΦ(s_{t+1}, a_{t+1})^T w^{a_{t+1}} − Φ(s_t, a_t)^T w^{a_t}
    for all a ∈ A do
        w^a ⟵ w^a + αδ_t e^a
        e^a ⟵ γλe^a
    end for
    if dε_{t−1} < 0.1 then
        ε_t ⟵ 0.1 {Maintain the exploration rate at 0.1 at minimum.}
    else
        ε_t ⟵ dε_{t−1} {Decay the exploration rate.}
    end if
end for
return w^a, for each a ∈ A
```

4.4.3 Adaptive Sensing for Accelerometer and GPS

Most existing sensors can be categorized into two groups based on their operation styles: 1) sensors that operate continuously for an optimal minimum sampling period to capture sufficient data for context recognition (e.g., accelerometer and microphone); 2) sensors that automatically switch to an idle state after the required sensing operation is completed (e.g., GPS, WiFi, and Bluetooth) [242]. Frequent activation and deactivation of sensors in the first group will generate overhead in energy consumption. In addition, both activation and deactivation take certain amount of time, creating limits in switching between 'On' and 'Off' states. Sensors in the second group are typically guided by dedicated communication protocols, which also create limits in their sensing rate. Accelerometer and GPS are two representative sensors from each of these two groups, and are chosen in this work to evaluate our proposed adaptive sensing strategies. We first design the state space S_t, the action space A_t, and the reward signal R_t in QLADE for both sensors, and then define the performance metrics for evaluations in our simulation experiments.

Table 4.1 shows the state features and their definitions. We divide the time throughout a day into 9 buckets based on intuition about people's daily routines. These include early morning, morning, noon, early afternoon, late afternoon, early evening, late evening, early night, and late night. This featurization of time maintains a reasonable amount of levels that is consistent with people's daily activities, while avoids creating too many levels for learning, which require many more learning samples. The accelerometer features characterize the potential motions (e.g., usage of the phone, movement within a confined space) within the sensing window at both time $t-1$ and t, while the GPS features represent the potential movements from one location to another during the sensing window at time t. In the single modality adaptive sensing scheme, we use the temporal and accelerometer features as the state features for accelerometer, and the temporal and GPS features as the state features for GPS; in the multiple modality adaptive sensing scheme, we combine all features as the state feature vector across different sensors. However, each type of agent will maintain their own policy, and use different reward signals defined below to update

TABLE 4.1: State features S_t for accelerometer and GPS.

group	feature	description
Time	early morning	binary, below the same. Between 6 and 8 am.
	morning	Between 8 and 11 am.
	noon	between 11am and 1 pm.
	early afternoon	between 1 and 4 pm.
	late afternoon	between 4 and 6 pm.
	early evening	between 6 and 8 pm.
	late evening	between 8 and 11 pm.
	early night	between 11 pm and 1 am on the next day.
	late night	between 1 and 6 am on the next day.
Accelerometer	current_avg_acc	the average acceleration from raw accelerometer data collected within the sensing window at time t.
	current_sd_acc	the standard deviation from raw accelerometer data collected within the sensing window at time t.
	prev_avg_acc	the average acceleration from raw accelerometer data collected within the sensing window at time $t-1$.
	prev_sd_acc	the standard deviation from raw accelerometer data collected within the sensing window at time $t-1$.
GPS	avg_distance	the average distance between all pairs of temporally consecutive GPS points collected within the sensing window at time t.
	std_distance	the standard deviation of distance between all pairs of temporally consecutive GPS points collected within the sensing window at time t.
	total_distance	the total distance traveled during the sensing window at time t.
	displacement	the distance that the participant moves from the starting location to the ending location during the sensing window at time t

the policies.

The action space includes two sensing actions: $A_t = \{\text{on,off}\}$. The reward signal is computed based on average acceleration and displacement within the adaptive sensing window at time step t and the sensing window at time step $t+1$ for accelerometer and GPS, respectively. Specifically, R_t^{acc} is given by:

$$r_t^{acc} = \begin{cases} 1 & a_t = \text{on} \ \& \ avg\_acc_t \geq th_{acc} \\ 1 & a_t = \text{off} \ \& \ avg\_acc_t < th_{acc} \\ -1 & a_t = \text{on} \ \& \ avg\_acc_t < th_{acc} \\ -1 & a_t = \text{off} \ \& \ avg\_acc_t \geq th_{acc} \end{cases},$$

where th_{acc} is a chosen acceleration threshold; and R_t^{GPS} is given by:

$$r_t^{GPS} = \begin{cases} 1 & a_t = \text{on} \ \& \ displacement_t \geq th_{GPS} \\ 1 & a_t = \text{off} \ \& \ displacement_t < th_{GPS} \\ -1 & a_t = \text{on} \ \& \ displacement_t < th_{GPS} \\ -1 & a_t = \text{off} \ \& \ displacement_t \geq th_{GPS} \end{cases},$$

where th_{GPS} is a chosen displacement threshold. We choose the acceleration and displacement thresholds to construct the reward signal because they indicate the magnitude of movements these sensors measure about their user. When the user is moving more actively, we want the sensors to be turned on and capture those activities. Thus under those conditions, we set the reward signal to be positive to reinforce the 'on' action. On the opposite, we set a negative reward to penalize the 'on' action when no or little movements are detected. Likewise, we will reinforce the 'off' action when the detected movement is below the chosen movement threshold, while penalize the 'off' action if the estimated movement is above the threshold.

Our performance metrics are defined based on four basic definitions: true positive (tp_t), true negative (tn_t), false positive (fp_t), and false negative (fn_t) at each time step t. They are defined in Table 4.2.

Table 4.2: Defintions for true positive, true negative, false positive and false negative.

true positive (tp_t)	$a_t = \text{on}$ &	$(avg\_acc_t \geq th_{acc}$ or $displacement_t \geq th_{GPS})$
true negative (tn_t)	$a_t = \text{off}$ &	$(avg\_acc_t < th_{acc}$ or $displacement_t < th_{GPS})$
false positive (fp_t)	$a_t = \text{on}$ &	$(avg\_acc_t < th_{acc}$ or $displacement_t < th_{GPS})$
false negative (fn_t)	$a_t = \text{off}$ &	$(avg\_acc_t \geq th_{acc}$ or $displacement_t \geq th_{GPS})$

The performance metrics given T time steps in the learning data are defined below:

- Accuracy is defined by $\frac{\#tp+\#tn}{T}$. Accuracy is intended to capture the percentage of time steps in which the action is correctly taken. A higher accuracy represents a better policy that the RL algorithm is able to generate.

- F-score is defined by $2\frac{precision*recall}{precision+recall}$, where $precision = \frac{\#tp}{\#tp+\#fp}$ and $recall = \frac{\#tp}{\#tp+\#fn}$. Precision reflects the percentage of steps in which the 'on' action is correctly taken (i.e., the actual movement level is beyond the chosen threshold) among all steps that 'on' action is taken, while recall reflects the percentage of steps in which the 'on' action is correctly taken among all steps that the 'on' action should be taken. The F-score reflects the balance between precision and recall.

- Percentage of battery saved is defined by $\frac{\#tn}{T}$. This metric refers to the percentage of time steps the 'off' action is taken when the actual movement level is below the chosen threshold. In continuous sensing, the energy consumed during these time steps would have been wasted, but instead would be saved by using our adaptive sensing policy. It also closely approximates the percentage of battery that can be saved when we assume the sensor will consume the same amount of power in each time step regardless of all other conditions in the device.

- Percentage of activity coverage is the same as recall. It measures the percentage of time steps, in which our proposed approach is able to capture the movements by turning on the sensor.

Existing works have leveraged mobile devices' own battery sensor to record energy consumption by specific apps and different types of phone activities; or when this is not possible, they resorted to benchmark the battery drainage in control experiments, and use the measured numbers to estimate battery consumption statistics. These methods highly depend on the similarity between the test scenarios and the phone usage conditions in the real world. Instead of adopting the same methods, we simply measure battery consumption based on sensors' operation time as a surrogate for performance metrics related to battery saving. Among these four metrics, we want to particularly focus on percentage of battery saved and percentage of activity coverage. When compared to duty cycle, the percentage of activity covered is our gain, while 1 minus the percentage of battery saved is our trade-off. Our goal is to find a strategy that gives us the maximum activity coverage and battery saved.

4.5 Experiments

4.5.1 Data

We conduct simulation experiments using continuous sensing data from a previous mobile sensing study that aimed to understand students' emotions and social anxiety over a two-week window [28]. In this study, we collected accelerometer, GPS, communication (e.g., text messages and phone calls), and Ecological Momentary Assessment (EMA) data of 220 students via the Sensus mobile application [236]. In particular, Sensus was configured to passively and continuously collect accelerometer data at 1 Hz and GPS coordinates every two and a half minutes for up to two weeks. We also collected daily affect scores via EMAs delivered at 10pm everyday, and social anxiety score using the SIAS scale at the start of the study. Part of our data has been made publicly available on the web [26].

4.5.2 Baseline Approaches

We implemented two baseline approaches: 1) a context-agnostic random strategy, which at each time step randomly decides whether to turn on the sensors or not; and 2) a learning automata strategy [179], which makes sensing decision based on probability p_{t-1} and adaptively increase or decrease the sensing probability at each time step based on the reward signal using the following formula:

$$p_t = \begin{cases} p_{t-1} & a_t = \text{off} \\ p_{t-1} - \alpha p_{t-1} & a_t = \text{on} \;\&\; (avg\_acc_t < th_{acc} \text{ or } displacement_t < th_{GPS}), \\ p_{t-1} + \alpha(1 - p_{t-1}) & a_t = \text{on} \;\&\; (avg\_acc_t \geq th_{acc} \text{ or } displacement_t \geq th_{GPS}) \end{cases}$$

where $\alpha \in (0, 1)$. In [179], α is chosen to be 0.5.

4.5.3 Experimental Settings and Research Questions

For accelerometer, We set the time window at each time step to be 300 seconds with the sensing and adaptive sensing window being 20 seconds and 280 seconds. Checking user's movement every five minutes seems reasonable for most mobile sensing applications, while a 20 second sensing window provides sufficient accelerometer data for constructing the state features to predict user's movement. For GPS, we choose an hourly time window with the sensing and adaptive sensing window being 20 minutes and 40 minutes, respectively. This decision is based on the sensing rate we used to collect the GPS data, which is one GPS point every two and a half minutes, and daily human routine mobility motifs [193], which discover that people visit only a few places on most days. The acceleration and displacement thresholds are chosen to be $0.2(g)$ and $0.1(km)$, respectively. In practice, these thresholds should be chosen based on the application requirement. For example, in fall detection, the acceleration threshold should be much higher than 0.2. In order to understand how the choices of these thresholds impact the simulation performance, we conduct sensitivity analyses by varying their values in Section 4.5.3.

Comparisons of Performance in Adaptive Sensing Strategies

There are four parameters in the RL algorithm that requires tuning. These include the initial exploration rate ϵ_0, the step-size parameter (or learning rate) α, the discount rate γ, and the eligibility trace-decay parameter λ. All four parameters take values between 0 and 1. We conduct various simulations using grid search with the multiple modality adaptive sensing scheme. The parameter grid is constructed using the following values in each parameter: 1) $\alpha = \{0.01, 0.05, 0.1\}$; 2) $\gamma = \{0.05, 0.1, 0.2\}$; 3) $\lambda = \{0.05, 0.1, 0.2, 0.5, 0.8\}$; and 4) $\epsilon_0 = \{0.1, 0.2, 0.5\}$. The exploration decaying rate is fixed to be $d = 0.999$. The optimal parameter setting from the grid search based on saved energy will be selected for the RL strategies. Two research questions we want to answer are: 1) what is the performance of the proposed adaptive sensing approach compared against the two baseline strategies? 2) which adaptive sensing scheme is better, the single modality or the multi-modality methods?

Data Utility Using Various Sensing Strategies

We investigate the impacts of using these adaptive sensing strategies on the utility of the collected data with respect to their applications. Specifically, how does the RL adaptive sensing strategies impact the data utility for predicting negative affect and social anxiety? We implement both regression and classificaiton tasks using data collected in three sensing scenarios: 1) using multi-modality RL strategy; 2) using duty cycle; and 3) using continuous sensing. In classification, we convert continuous scaled negative affect and social anxiety into binary values using cutoffs 50 and 34, respectively. Performances are evaluated using accuracy for classification, and mean square error (MSE) for regression.

The GPS features are extracted based on the algorithm in [39], and the accelerometer features include average, median, variance, standard deviation, minimum, maximum, range, first quartile, third quartile, inter-quartile, inter-quartile range, RMSE, MAD, skewness, kurtosis, mean jerkiness, and lag one auto-correlation of accelerations, correlations between pairs of axes, zero crossing rate on each axis. For daily negative affect, the features are extracted using the last 24 hours of data from the response timestamp; for social anxiety, we calculate only the proportion of time the users are being active (i.e., mean acceleration above 0.2 using 5 minute windows). We build generalized models with all participants' data using random forest and logistic regression for classification, and using random forest and LASSO linear regression for regression.

Sensitivity Analysis on Thresholds and Sensing Windows

The acceleration and displacement thresholds are application dependent, and related to the computation of reward signal in the adaptive sensing strategies. The state feature window size affect both the energy efficiency and our performance metrics. We conduct sensitivity analyses to understand these impacts using the following values for accelerometer and GPS: 1) the acceleration thresholds $th_{acc} = \{0.05, 0.1, 0.2, 0.5, 1\}(g)$ and the state window sizes $acc_{sw} = \{20, 30, 60, 120\}$ (seconds); 2) the GPS displacement thresholds

$th_{GPS} = \{0.05, 0.1, 0.2, 0.5, 1\}$ (km) and the state window sizes $gps_{sw} = \{600, 1200, 1800\}$ (seconds).

The accelerometer sensitivity analyses are conducted by fixing the GPS displacement threshold and state window size at 0.1 (km) and 1200 (seconds); while the GPS sensitivity analyses are conducted by fixing the acceleration threshold and state window size at 0.2 (g) and 20 (seconds).

Personalization of the Learning Policies

Every RL adaptive sensing strategy is encoded by the learned policy, which is represented as a weight vector corresponding to the state features. We analyze and visualize these policies collectively and individually, to understand how the action decisions are made under different state values.

4.6 Results

4.6.1 Comparisons of Performance in Adaptive Sensing Strategies

Figure 4.4 shows the performance metrics on the various values each RL parameter uses in the simulations. For accelerometer, the learning rate α at 0.01 consistently provides the best results on all four metrics; the discount rate γ is less sensitive to all four metrics, and is chosen at 0.05 based on energy saving; the eligibility trace-decay parameter λ is also quite insensitive to all four metrics, and is chosen at 0.1 based on energy saving; the initial exploration rate ϵ_0 is chosen at 0.1. We choose the same set of values from accelerometer for all four RL parameters in GPS.

Figure 4.5 shows the performance of the four sensing strategies on the four performance metrics defined in Section 4.4.3 using all available accelerometer and GPS data from each study participant in the simulations. Thus the performance metrics from each participant are computed based on different T values.

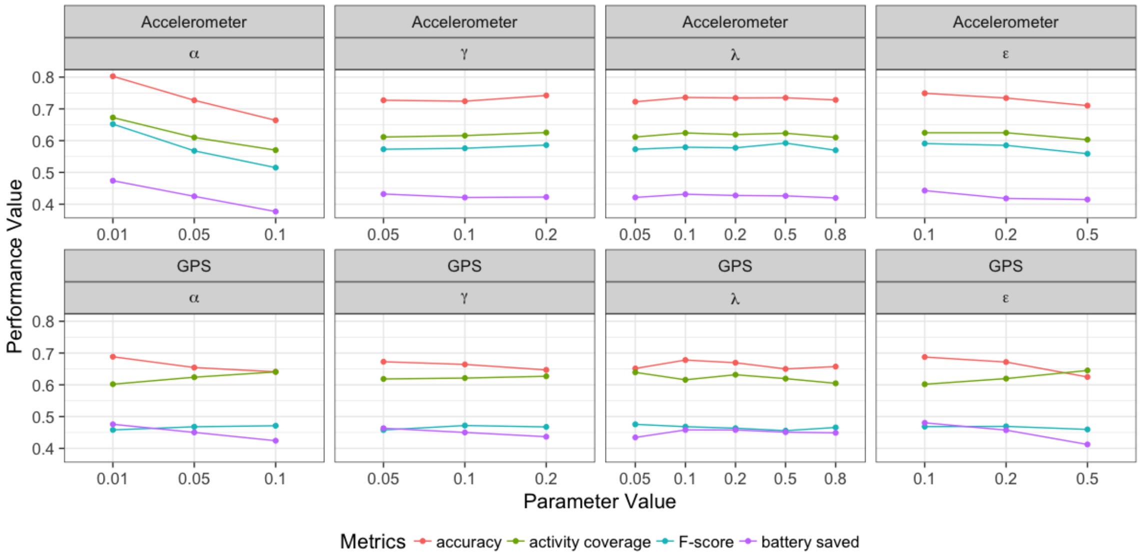

FIGURE 4.4: Parameter Tuning

Single vs. Multiple Modality. For accelerometer, single modality method outperforms the multiple modality method on all four metrics. For example, the accuracy, activity coverage, F-score, and battery saved of the single modality method are $0.904, 0.826, 0.841, 0.262$, respectively, as compared to $0.826, 0.762, 0.841, 0.242$ for the multiple modality method. For GPS, the single modality method attains better performance in accuracy and battery saved at 0.687 and 0.308, as compared to 0.683 and 0.277 in the multiple modality method; and worse performance in activity coverage and F-score at 0.66 and 0.612, as compared to 0.727 and 0.624 in the multiple modality method. Overall, multiple modality method seems to perform worse than single modality, even though it leverages more data to construct the state features and guide action decision. With more information from all available sensors, the potential benefits of the multiple modality method is obvious. However, if the state features are not properly chosen, more information may not necessarily be translated into better performance. One possible explanation could be that state features generated from different sensor data provides conflicting information and therefore lower the accuracy of sensing decisions. More specifically, movements detected by accelerometer (e.g., high value in average acceleration) could either represent movements within a confined space, or between different locations. Similarly, displacements detected by GPS (e.g., high value in displacement)

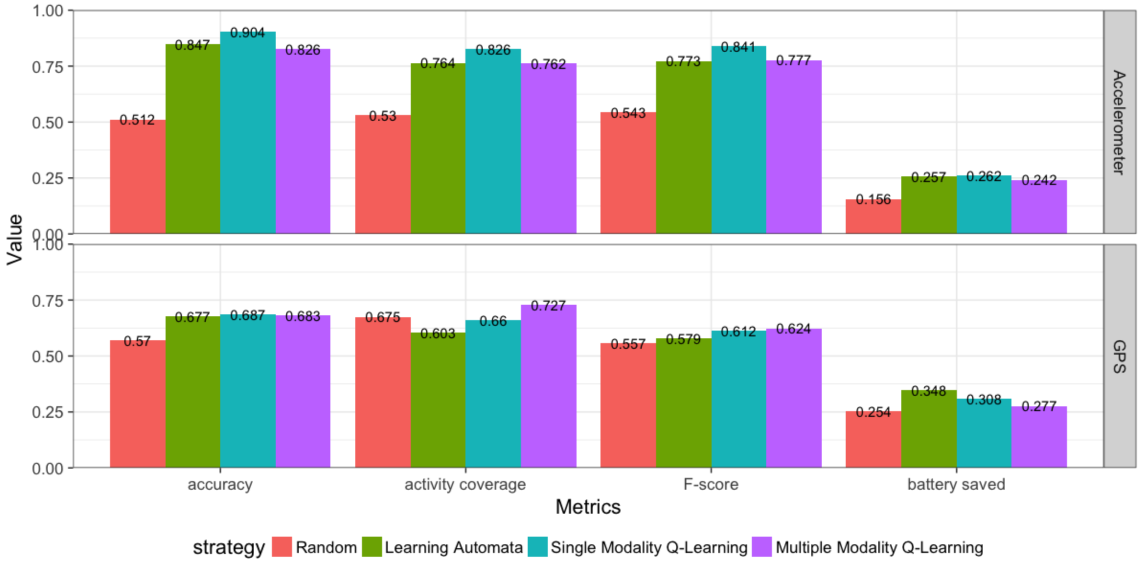

FIGURE 4.5: Performance on Sensing Strategies.

could either represent displacements occur in a vehicle, or while running.

RL Strategies vs. Baseline Methods. The accelerometer single modality QLADE consistently outperform the learning automata method, while multiple modality QLADE is slightly worse than but fairly closed to it. The GPS learning automata baseline is able to save more energy, but has lower activity coverage rate when compared to the QLADE algorithms. Currently the changing rate α in the learning automata method is set to 0.5. Higher α value results in faster changes in the sensing probability p_t. When a user frequently changes his/her state, the learning automata method is expected to have worse performance due to the swinging of the sensing probability. Overall, the QLADE algorithm has better performance than the baseline methods. And since it leverages contexts extracted from the sensor data to inform sensing decision, we believe that with properly designed state features, it can achieve significantly better results.

Figure 4.6 and 4.7 show the average performance over time for all four sensing strategies in accelerometer and GPS. The horizontal axis represent the time steps (5 minutes for accelerometer and 1 hour for GPS), and the dashed vertical lines mark the first quartile in number of learning samples or time steps in all participants. In Figure 4.6, the single modality QLADE method consistently outperforms the other methods over time for accelerometer. We also observe that the multiple modality QLADE method has similar

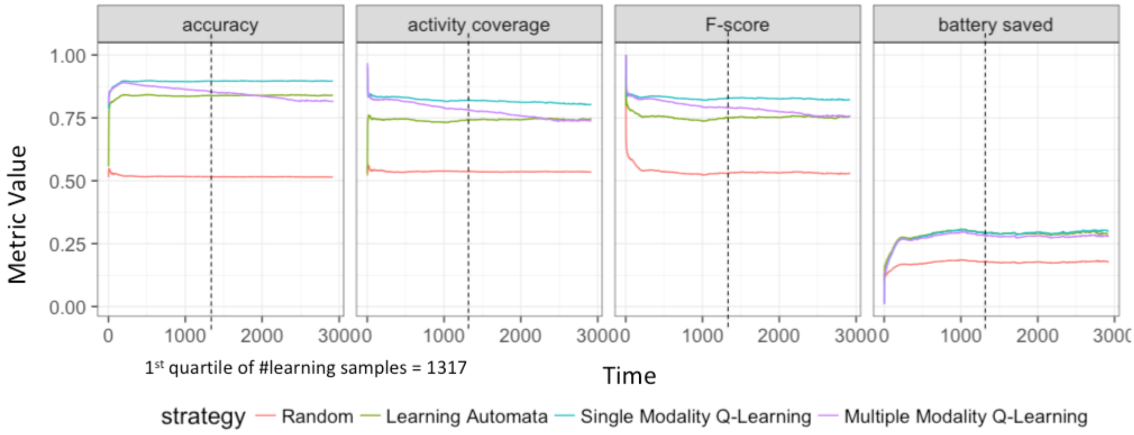

FIGURE 4.6: Average performance over time across all participants on accelerometer sensor.

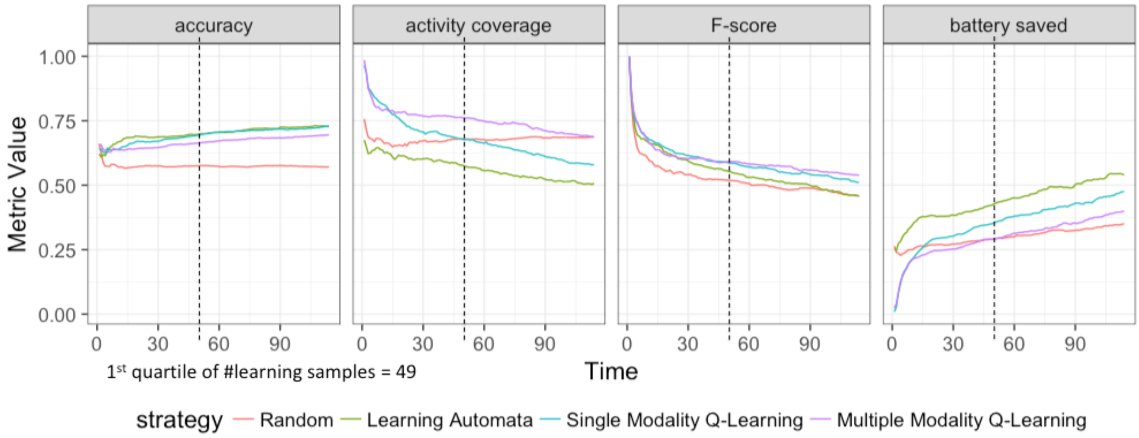

FIGURE 4.7: Average performance over time across all participants on GPS sensor.

initial performance to the single modality QLADE method. However, its performance drops around time step 250, and gradually goes below the learning automata method. In Figure 4.7, we observe less consistent average performance over time between the QLADE methods and the learning automata baseline. Specifically, there is no convergence in performance over time, and the random baseline method performs better than the other methods up to certain time points in activity coverage and battery saved. These inconsistencies reveal that it is much more challenging to balance coverage of place movements and energy saving. One possible explanation could be that the GPS data has lower data density due to the sensing rate and higher missingness when compared to the accelerometer data.

TABLE 4.3: Comparison of data utility among three different sensing strategies. The social anxiety score has a range of 0-80, and the daily negative affect has a range of 0-100.

		Classification (accuracy)		Regression (mse)	
	sensing strategy	Random Forest	Logistic Regression	Random Forest Regression	LASSO Linear Regression
SIAS	adaptive	0.611	0.637	138.1	148.6
	duty cycle	0.556	0.582	140.8	134.3
	continuous	0.600	0.630	134.0	145.8
Daily Negative Affect	adaptive	0.774	0.774	543.4	534.8
	duty cycle	0.779	0.778	525.5	533.1
	continuous	0.780	0.778	528.0	529.4

4.6.2 Data Utility Using Various Sensing Strategies

Table 5.1 shows the performances on both the classification and regression tasks using random forest, logistic regression, and linear regression on social anxiety and daily negative affect. The adaptive sensing strategy provides better performance than the duty cycle and continuous strategies in the classification task, while slightly worse results in the regression task for social anxiety. The differences among the three sensing scenarios are even smaller for daily negative affect, regardless of the type of tasks or algorithms. Overall, using adaptive sensing does not compromise data utility in predicting social anxiety, and daily negative affect.

4.6.3 Sensitivity Analysis on Thresholds and State Windows

Figure 4.8 shows the sensitivity analyses on accelerometer. All four performance metrics in accelerometer are not sensitive over different state windows. In contrast, different acceleration thresholds greatly influence battery saved, activity coverage, and F-score, and to a lesser extent accuracy in accelerometer. Meanwhile, all four performance metrics in GPS are not sensitive over different accelerometer state windows and acceleration thresholds. The impacts from the changes of the acceleration threshold is expected as higher acceleration threshold corresponds to less activation of accelerometer, leading to

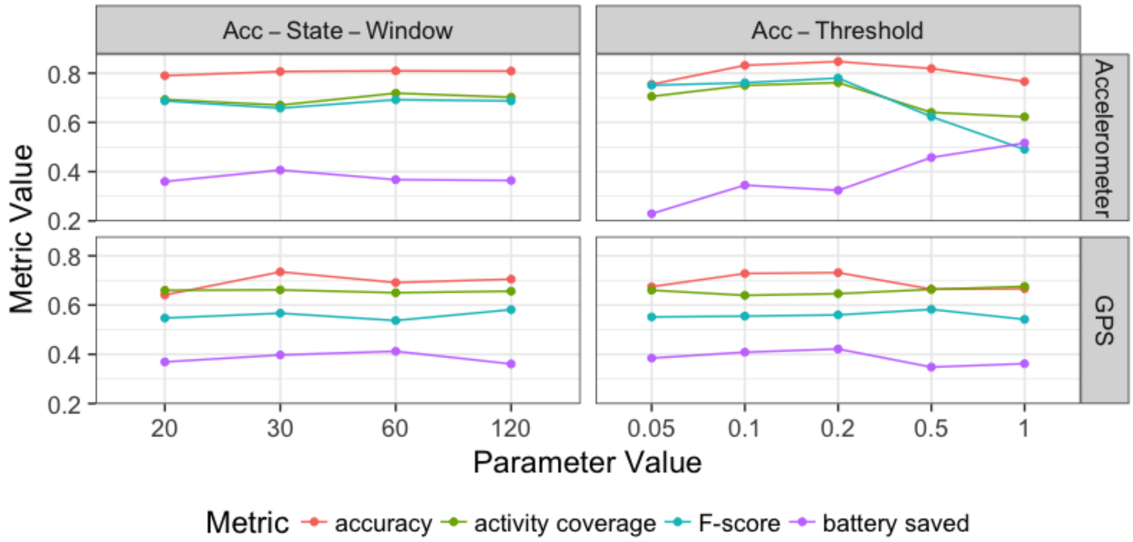

FIGURE 4.8: Sensitivity analysis on accelerometer state Window and acceleration threshold when GPS state window and displacement threshold are fixed at 20 minutes and 0.1 kilometers.

reduced energy consumption. However, higher energy saved will trade-off lower activity coverage.

Figure 4.9 shows the sensitivity analyses on GPS. Varying the GPS state window and displacement threshold does lead to significant variance in all four performance metrics in accelerometer. For GPS, the same conclusion from accelerometer sensitivity analyses can be made, that is varying the GPS displacement threshold leads to increasing battery saved and decreasing activity coverage, while all four performance metrics are not sensitive to changes in the state window.

4.6.4 Personalization of the Learned Policies

Figure 4.10 visualizes the distribution of counts of favored actions by each state feature as the values of the state features increase. Specifically, the red bar in each state feature corresponds to the 'On' action, while the green bar corresponds to the 'Off' action. The counts are the number of study participants that have the state feature favoring the corresponding action when its value increase. For example, 86 accelerometer agents favor the 'On' action when the time is in early morning, and 130 accelerometer agents favor the 'On' action when average acceleration increases. Likewise, 137 accelerometr agents favor

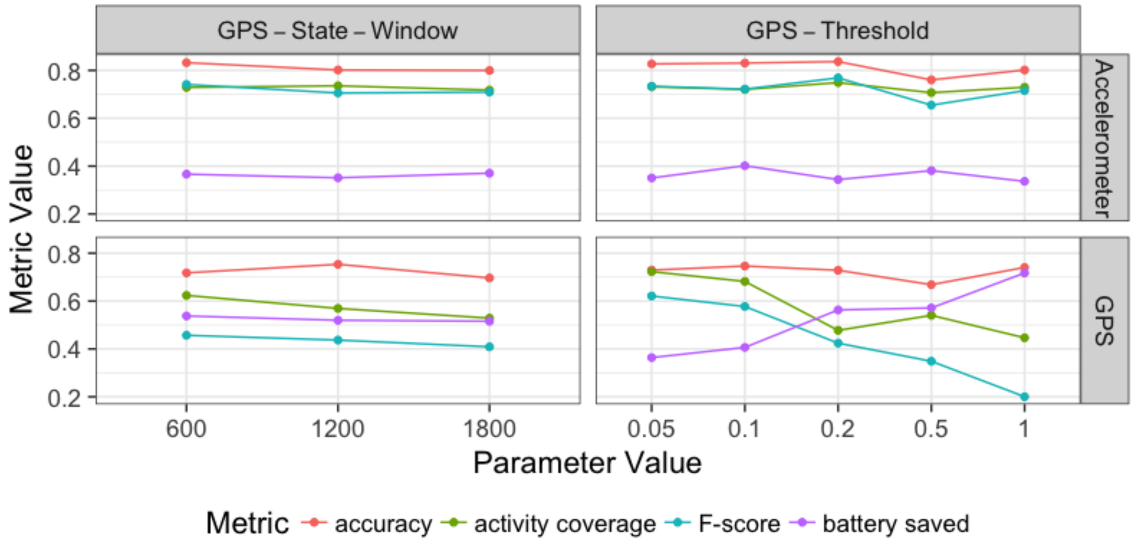

FIGURE 4.9: Sensitivity analysis on GPS state window and displacement threshold When accelerometer state window and acceleration threshold are fixed at 20 seconds and 0.2 gravity unit.

the 'On' action when total distance increases. Average acceleration and total distance are the two most dominant positive features that favor the 'On' action, followed by average acceleration in $t-1$, and displacement. For GPS, early night time and standard deviation of acceleration in t favor the 'Off' action, while all four GPS state features favor the 'On' action, as their values increase.

Figure 4.11 visualizes the individual policies for participants with the number of learning samples in the top three quartiles for both accelerometer and GPS. Red cells in these two heatmaps represent favoring the 'On' action as the corresponding state feature value increases, while dark blue cells represent favoring the 'Off' action as the corresponding state feature value increases. Sensing agents located at the bottom of the heatmaps tend to be less active for most of the time throughout the day. This can be seen from the blue cells corresponding to the temporal state features in both heatmaps. In some GPS policies, high values in average acceleration in t favor the 'Off' action, which is possible only when the corresponding participants move sporadically within short time window (e.g., checking phones frequently). We also find that higher standard deviations of acceleration for a significant portion of accelerometer and GPS policies favor the 'off' action. One possible explanation is that when average acceleration in t favors the 'Off' action,

Figure 4.10: Number of participants favoring each of the two actions with respect to each state features using the multiple modality Q-learning strategy. Only data from participants that have learning samples in the top three quartiles are counted.

the participants may be moving sporadically within short time window; when it favors the 'On' action, the participants may be engaging in activities with consistently high acceleration.

From the collective and individual views of sensing policies in both accelerometer and GPS, we show the individual differences among participants' behavior patterns, and therefore the need for adaptive sensing strategies. The RL framework is an ideal solution to personalize adaptive sensing.

4.7 Discussion

Energy efficiency is an important topic in mobile sensing applications, which create significant energy demands while compete with other services personal smartphones provide for limited battery. In this work, we systematically identify opportunities in different stages within the mobile sensing application pipeline to address the energy challenge, and

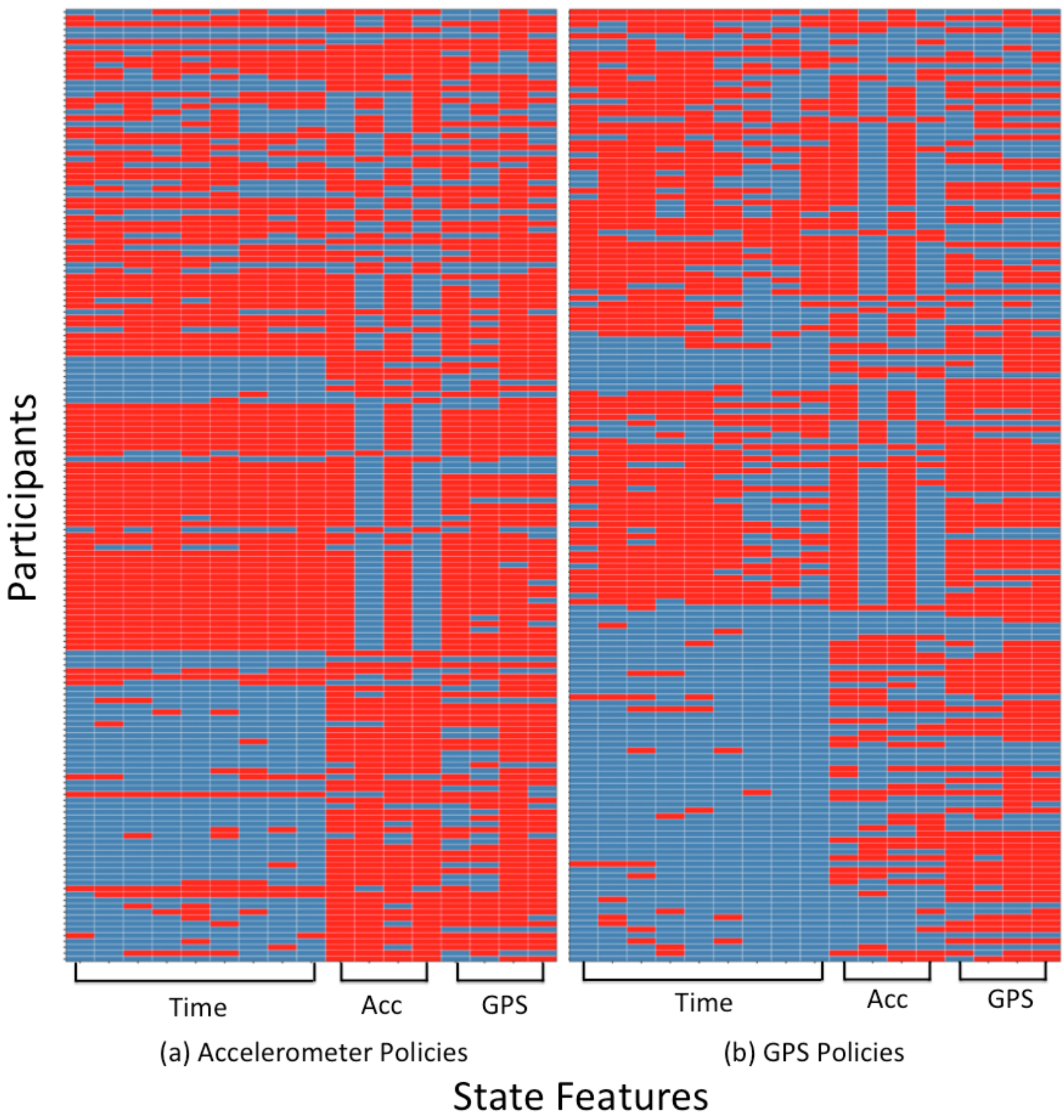

FIGURE 4.11: Individual state feature profiles on the two actions using the multiple modality QLADE strategy in accelerometer and GPS. Red color represents the 'On' action, and blue color represents the 'Off' action.

propose a reinforcement learning framework for sensing management that can lead to higher energy efficiency in mobile sensing.

Our proposed adaptive sensing strategy is based on a basic principal, that is sensors should be deployed only when it matters. User contexts can be leveraged to help sensing decision to reduce sensor deployment and thereby enhance energy efficiency. The RL framework enables contextual understanding using carefully designed state features, and optimizes sensing by learning and maintaining a sensing policy. Using a model-free approach, our adaptive sensing strategy requires no prior training, environment modeling, or building context recognition models. In addition, the learned policies are personalized and adaptive, which are very important features for energy efficient mobile sensing.

The per timestep computation complexity for updating an existing policy is linear with $|A| \times |\Phi|$, hence the energy overhead is quite trivial when compared to energy consumed by hardware sensing [144]. Energy overhead also comes from extracting state features using computationally intensive methods. However, this is not a concern in this work. We are currently working on deploying our proposed strategy in a human study to evaluate its effectiveness. We discuss several limitations in our proposed methods and some ideas for future works below.

4.7.1 Limitations

First, the current formulation posits our adaptive sensing strategy as the middle ground between continuous sensing and duty cycling. This formulation limits energy efficiency to exceed duty cycle method, while trade-off some energy for activity coverage. However, it is possible to reformulate the adaptive sensing problem, and redefine the sensing actions to achieve better energy efficiency than the duty cycle method.

Second, more sophisticated state features can be designed to better characterize the contexts that are relevant for sensor deployment. The performance of the multi-modality QLADE strategy is not improved by adding additional state features from other sensors. Thus, a better understanding of the correlations among data from different sensors is needed to aid the design of state features. One approach is to design a unique set of state features for each sensor in the multi-modality scheme, rather than having a common set of state features for all sensors.

Third, under the current formulation, the reward signal relies on thresholds that require manual specification. In this work, we choose $0.2g$ in acceleration for accelerometer, and $0.1km$ in displacement for GPS. However, it is not always straightforward for other sensors such as light sensor and microphone to set these thresholds. To mitigate this challenge, we could redesign the reward signal to get rid of its reliance on the threshold parameter, as long as it can provide desired feedback to the sensing agent in updating the policy. Removal of the threshold parameter can also increase the robustness of the reward signal. For example, activities that last in short time span may be treated as no movement due to taking the average value.

Last and most importantly, we did not evaluate our proposed adaptive sensing strategy in a real study using randomized controlled experiment, and thus the current evaluation in energy efficiency is only exploratory. Although, we argue that the comparison among the several strategies using the current performance metrics is fair and valid, and the current work paves ways for our future human study.

4.7.2 Future Work

To amend the above limitations and extend our current work, we propose the following future directions. First, we want to reformulate the adaptive sensing problem to allow more sensing actions, which can further improve energy efficiency. Unlike the action space in this work, a new action space can include control options that determine the duration of sensor deactivation. This new design in action space also requires adjustments in the reward signal to properly estimate the reward values in time steps that the sensors are deactivated. One approach is to discount the reward proportionally to the length of the deactivation window.

Second, in this work, we apply universal state features for all sensors in the multi-modality scheme, which fails to deliver better performance. More sophisticated and unique state features need to be considered for each sensor. For example, instead of using mean and standard deviation of acceleration as the accelerometer state features, we can apply distribution based features that can better characterize the underlying activities. For GPS, many existing works leverage accelerometer signal to activate GPS. However, movement within a confined space (e.g., being indoors) will lead to unnecessary triggers of GPS. We need to consider this issue when design state features using accelerometer data for GPS.

Third, we will redefine the reward signal to remove the threshold parameter. In this work, the reward signal takes binary values. We will experiment with a continuous reward that is proportional to discounted cumulative 'inertia' level (e.g., movement for accelerometer, and displacement for GPS). Lastly, We will explore other reinforcement learning algorithms for further improvements in energy saving and activity coverage.

4.8 Conclusion

This work proposes an adaptive passive sensing framework using reinforcement learning to control low-level sensor sensing cycles. Our proposed approach can be combined with other methods targeting energy challenges arising from different sources in the mobile sensing application pipeline (Figure 4.1).

From our experiments, we find that the single modality QLADE algorithm attains better performance than the learning automata baseline approach. Specifically, it outperforms the learning automata method in all four metrics in accelerometer. In GPS, both single and multi-modality QLADE methods are better than the learning automata baseline approach in accuracy, activity coverage, and F-score, but worse in battery saved. When compared to typical duty cycle strategy, we trade-off around 75% of energy for around 70% of activity coverage, using the current selected acceleration and displacement thresholds. When including only data from study participants ranked in the top three quartiles in the amount of available learning samples, we see consistently better average performances in all four metrics over time in accelerometer. Our proposed method does not compromise data utility in three sensing scenarios including adaptive sensing, duty cycle, and continuous sensing. Specifically, we are able to obtain similar prediction performances in both classification and regression tasks in social anxiety and daily negative affect. We show that our proposed method is robust in the state feature window but sensitive to the movement threshold.

Chapter 5

Adaptive Passive Sensing Using Action-augmented QLADE with Various State Designs and a Continuous Reward Signal

In this chapter, we extend the proposed work in Chapter 4 by reformulating adaptive passive sensing with an expanded action space, new sets of state features, and a more generalizable reward signal. Our proposed RL strategies consistently outperform the baseline methods including the dynamic function method, the learning automata method, the duty cycling method, and a random strategy in energy saving. To verify the impacts of these different strategies on data utility, we predict social anxiety and daily negative affect using real data collected in a mobile sensing study on mental health. Using our proposed RL strategy does not result in lower prediction performance when compared to the baseline strategies.

5.1 Introduction

Embedded motion sensors, including acceleromter, gyroscope, compass, and altitude sensors etc., are deployed in many mobile sensing applications [180, 67] to continuously collect users' motion data, which leads to significant energy challenge on the smartphone. In addition to hardware sensing, processing and transmitting large amount of motion data will also consume substantial energy. [144] Applying adaptive passive sensing strategy can greatly mitigate these energy challenges by only collecting motion data when significant state changes are detected. In Chapter 4, we propose a general design framework for adaptive passive sensing using reinforcement learning. There remains several limitations in the problem formulation and design in the RL components.

Primarily, when we formulate adaptive passive sensing as a middle ground between continuous sensing and duty cycling, the energy efficiency we obtain using adaptive sensing is upper-bounded by that of duty cycling. The sensing window within every time step guarantees regular assessments of user's contexts. However, making each time step equal length limits energy saving in moments when no signal changes for longer period of time happen. One easy extension to address this issue is to augment our previous formulation with actions that can control the length of time when sensors are kept from deployment. Specifically, we choose a basic time unit (step) and define the action to be the number of time units the RL agents can skip through the next sensing cycle. Each sensing cycle consists of sensing for one time unit and skipping for A_t number of time units. We aim to improve energy efficiency by allowing the sensors to be turned off in longer time periods that have no state changes detected in the sensing time unit.

State representation is another limitation in our proposed method. Many non-adaptive methods directly apply thresholds to define controlling conditions for sensor deployment. In the previous chapter, we apply state features that characterize 'movements' in accelerometer and GPS, considering the goal of capturing varying acceleration and displacement signal in these sensors. However, there is a lack of works on applying RL in adaptive sensing, and thus a lack of guidance in state representation learning for adaptive sensing. To fill in this gap, we propose three state design methods and evaluate their effectiveness in this chapter.

Lastly, in our proposed method, the reward signal depends on a manually specified threshold value in 'movements', which may not generalize across different sensors, and can fail to correctly guide the policy updating. The role of reward signal is to reinforce or weaken the bonds between actions and their associated states. This mapping is eventually translated into a policy that is leveraged to control sensor deployments. Given the goal is to capture signal changes in the sensor data, designing the reward signal should follow this basic principle: if the action is desired with respect to the goal, the reward signal should be reinforcing selection of the action in the given state; on the contrary, the reward signal should be penalizing selection of the action in the given state. We follow this design principle in our previous method and define a binary reward signal that takes value -1 and 1 based on the chosen action and a threshold value. In order to set free from this dependency, we design the reward using a continuous scale that estimate signal changes proportional to the time length in this chapter.

In this work, we propose to control the deployment of a set of motion sensors by adaptive sensing using accelerometer. For example, if the accelerometer detects state changes in a sensing time unit, the chosen set of motion sensors, including the accelerometer, will be deployed to collect data. Hence, we explore adaptive sensing using accelerometer with an action-augmented formulation that enable more energy saving, while address challenges in state and reward design. In the remainder of this chapter, we summarize related works in Section 5.2, explain the new problem formulation in Section 5.3, propose different RL methods in Section 5.4, lay out our experiment plan in Section 5.5, and present their simulation results in Section 5.6. We provide a brief discussion on limitations and future works in Section 5.7 and make our conclusion marks in Section 5.8.

5.2 Related Works

State representation design is an important topic in RL across various application domains such as resource management [139], traffic control [8], robotics [107], and more recently, chemistry [248], news recommendation [245], real-time bidding [96], and adversarial games [199]. Although its importance has been noted, it has not been systematically covered in Sutton and Barto's classic introductory work in RL [203]. State design plays an important role in most of these applications but often requires domain knowledge to manually extract features from data generated in the different environments.

In [139], the state is designed as color coded images to represent jobs being processed in clusters and jobs waiting in job slots in the resource management environment. In [8], the state is represented by an eight-dimensional feature vector with each element representing the relative traffic flow at one of the lanes. The relative traffic flow is defined as the total delay of vehicles in a lane divided by the average delay at all lanes in the intersection. In [248], the state is represented by a combination of different aspects in an experimental condition in chemical reactions. In [245], four sets of state features are extracted from user logs in online news services such as Google News. These features include 417 news features that describe properties appear in the piece of news, 2065 features regarding user clicks in different time windows, 25 user and news features regarding interactions between user and certain piece of news, and 32 context features describing the contexts when news requests are made (e.g., time and age of the news). In [96], the cumulative cost and revenue between merchants and consumers in each merchant cluster, and a set of slowly-changing consumer features (e.g., total cost and revenue) in each learning episode (e.g., a period of time) are applied to characterize the state in advertisement bidding.

The advances in deep learning research have led to efforts in autonomous state representation learning, in which real-world sensor data are applied to automatically develop a state representation for RL agents in an end-to-end approach [23]. However, this may not always be feasible in practical problems that are expensive to acquire large amount of learning data, especially those in mobile health and smartphone sensing. In our current work, we focus on manually extracting state features that can characterize the underpinning human activities in accelerometer data.

Many different accelerometer features have been applied in human activity recognition (HAR), and these include discrete Wavelet Transform (DWT) features, fast Fourier Transform (FFT) features, and different time-domain features [172]. Some recent work applied end-to-end method using deep learning approach for feature extraction in HAR.

Specifically, the authors in [45] constructed CNN models using tri-axial acceleration signals to predict eight typical daily activities. Among the above methods, time-domain features are most popular as they require no computational intensive process like DWT and FFT for data transform, and are less demanding on data quality caused by missing data and sampling rate. [172] This is very important in the context of adaptive sensing using RL, because applying computationally intensive process to construct the various RL components can create significant energy overhead, compromising energy efficiency in our proposed RL approach. Among the time-domain features, entropy features extracted from accelerometer trajectories can measure uncertainties in the data. Plotz et al. propose distribution-based features from accelerometer data for activity recognition tasks [171]. They found that distribution-based features can greatly improve HAR performance by accurately represent the acceleration signal's underlying activity. We apply both entropy and distribution-based state features in our current work.

Several existing works proposed RL-based adaptive sensing methods towards improving energy efficiency in mobile sensing. The Jigsaw engine proposed by Lu et al. [134] adjusts the sensing rate by formulating adaptive sensing in GPS as a Markov Decision Process (MDP) with sensing duration, hardware status (e.g., remaining battery budget), and mobility as state features. Krause et al. created a MDP for activity transition using real accelerometer data, and translated the adaptive sensing problem into an optimization problem that chooses the best sensing timings with respect to the activity recognition accuracy based on the activity MDP subject to certain energy constraints. Wang et al. proposed to use hidden Markov model (HMM) to adapt duty cycles in different user states given by pre-trained HAR models [228]. The action in each time step consists of an sensor activation decision, and a decision on deactivation duration when sensor is deactivated. The user state is estimated with uncertainty levels when no sensor data is available. Two obvious limitations with the above methods are as following: 1) model-based approach requires building environment model using real data, and can only be applied in discrete state spaces. 2) these approaches rely on HAR models to understand the environment state. These issues limit their generalizability across different sensors to enable adaptive

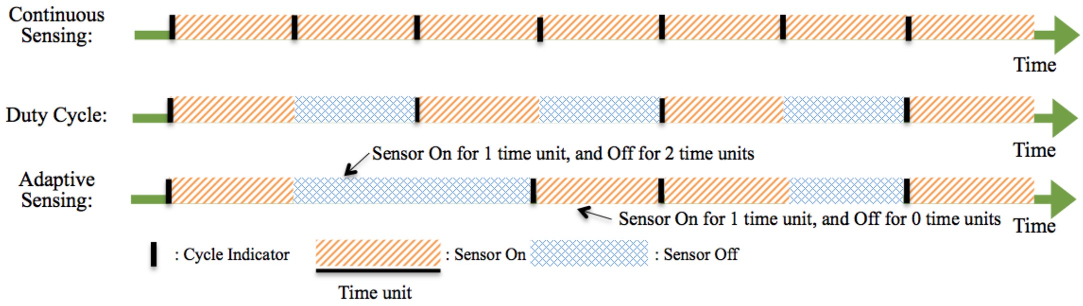

FIGURE 5.1: New problem formulation in adaptive passive sensing.

sensing.

Rachuri et al. proposed two different adaptive sensing approaches, one using a set of advance and back-off functions to control the sensing rate [178], while the other applying learning automata to adapt a sensing probability that controls sensing cycles [179]. We keep the learning automata method as our baseline comparison as we do in the previous chapter, and add the dynamic function approach to our baseline methods by generalizing it from controlling sensing rate to adapting sensing cycles for accelerometer. More details about them will be provided in Section 5.5.

5.3 Problem Formulation

Adaptive passive sensing could be formulated as a discrete time sequential controlling problem for sensor deployments. Time is divided up into steps with chosen step sizes for different sensors. At the first time step, the targeted sensor will be deployed, and the collected sensing data will be processed to inform sensor deployment decision in the next few time steps. If the sensor is turned on, it will sense in the next time step; otherwise, it will be turned off for a certain amount of time steps. The number of time steps to sleep the sensors is a decision by the controlling algorithm. A sensing cycle is therefore consists of a sensing window for one time step, and an adaptive sensing window with variable time steps. Figure 5.1 shows a few examples of the sensing cycles. Sensing cycle 1 consists of 3 time steps with the sensor being turned off for two time steps; while sensing cycle 2 consists of only 1 time step, and the sensor is not turned off.

This problem formulation is different from our general formulation in Chapter 4, in which adaptive sensing is posited as a middle ground between continuous sensing and duty cycling. In that case, each sensing cycle has the same length, and can be broken down into a sensing window and an adaptive sensing window. In contrast, our current formulation enables us to emulate energy efficiency performance in duty cycling, while potentially maintain superior activity coverage in our proposed reinforcement learning methods. It is achieved through generalizing from a fixed adaptive sensing window to a flexible adaptive sensing window (i.e., the sensor can be turned off for a longer period of time in certain contexts) using a more sophisticated action space design. More details will be provided on this in Section 5.4.

5.4 Methods

5.4.1 Reinforcement Learning Methods

Each reinforcement learning method requires specification of the three essential components: State Space S, Action Space A, and Reward R. We propose four different RL methods for adaptive passive sensing in this chapter based on their state feature design. All methods will include temporal features based on the sensing time. We apply the same algorithm proposed in the previous chapter, namely Q-learning with linear approximation and decaying exploration (QLADE). Below we describe the details on each of the proposed methods.

State Space Design

RL method 1: Entropy-based Adaptation (EA). Let $\mathbf{X}$ denote the sensing stream generated by the target sensor from the sensing window in a sensing cycle. A histogram is fitted based on $\mathbf{X}$, resulting in N bins, which contains n_i number of data points in the ith bin with $i = 1, 2, \ldots, N$. We define P_i to be $\frac{n_i}{N}$ and calculate the entropy of the sensing stream $\mathbf{X}$ using the following formula:

$$H(\mathbf{X}) = E[-logP(X_i)] = -\sum_{i=1}^{N} P_i logP_i.$$

Once we obtain $H(\mathbf{X})$, it is fitted into the predefined M buckets, which is dependent on N. In doing so, we discretize $H(\mathbf{X})$ into one of the M levels as a discrete state feature. These features combined with the temporal features will serve as the state features for the entropy-based adaptation algorithm.

RL method 2: Distribution-based Adaptation (DistA). Similarly, for sensing stream $\mathbf{X}$, we first derive the empirical cumulative distribution function (ECDF) F using the standard Kaplan-Meier estimation [74]. Specifically, the probability of getting a data point smaller or equal to x is:

$$\widehat{P}(x) = 1 - \prod_{i:x_i \leq x} (1 - \frac{d_i}{n_i}),$$

where d_i is the number of values in sensing stream $\mathbf{X}$ that is equal to x, and n_i is the number of values in sensing stream $\mathbf{X}$ that is greater than or equal to x. $\widehat{P}(x)$ takes monotonically increasing values from $[0, 1]$. We then choose a fixed set of N points $\mathbf{p} = \{p_1, p_2, \ldots, p_N\}$ in $[0, 1]$, and estimate the inverse values of $\widehat{P}(x)$ using cubic interpolation $C^{\mathbf{p}}$. We use a family of cubic interpolating splines called Catmull-Rom splines [213]. Specifically, we have

$$x(p) = c_3 p^3 + c_2 p^2 + c_1 p + c_0. \tag{5.1}$$

Assuming p falls between $\widehat{P}(x_i)$ and $\widehat{P}(x_{i+1})$. For simplicity, let us denote $\widehat{P}(x_i)$ and $\widehat{P}(x_{i+1})$ as p_i and p_{i+1}, respectively. Without loss of generality, let us assume all x_i are sorted within $\mathbf{X}$, and the nearest points smaller than p_i and greater than p_{i+1} are p_{i-1} and p_{i+2}, respectively. We have

$$x(p_i) = c_3 p_i^3 + c_2 p_i^2 + c_1 p_i + c_0 = x_i, \tag{5.2}$$

$$x(p_{i+1}) = c_3 p_{i+1}^3 + c_2 p_{i+1}^2 + c_1 p_{i+1} + c_0 = x_{i+1}, \tag{5.3}$$

$$x'(p_i) = 3c_3 p_i^2 + 2c_2 p_i + c_1 = \frac{1}{2}(x_{i+1} - x_{i-1}), \tag{5.4}$$

$$x'(p_{i+1}) = 3c_3 p_{i+1}^2 + 2c_2 p_{i+1} + c_1 = \frac{1}{2}(x_{i+2} - x_i). \tag{5.5}$$

Solve the above four equations for the four parameters c_3, c_2, c_1, c_0, we have

$$c_3 = \frac{4c^2}{12p_i^2c^2 + 12p_ibkc - 4ac + 6b^2}\left(\frac{x_{i+1} - x_{i-1}}{2} - \frac{4c(x_{i+1} - x_i) - bk}{4c^2}\right),$$

$$c_2 = \frac{k - 6bc_3}{4c},$$

$$c_1 = \frac{(x_{i+1} - x_i) - ac_3 - bc_2}{c},$$

$$c_0 = x_i - c_3 p_i^3 - c_2 p_i^2 - c_1 p_i,$$

where $a = p_{i+1}^3 - p_i^3$, $b = p_{i+1}^2 - p_i^2$, $c = p_{i+1} - p_i$, and $k = x_{i+2} - x_{i+1} - x_i + x_{i-1}$. In the case of endpoints, the right hand side of equation 5.4 or 5.5 will become $\frac{1}{2}(x_{i+1} - x_i)$, depending on which end we are at. The resulted inversed values using $\mathbf{p}$ in equation 5.1 will be used as the state features. The idea is to derive a representation of the input sensing stream that can preserve the structural information, thus capture the underpinning activities the user is performing [171].

RL method 3: Location Augmented Distribution-based Adaptation (LocDistA). In LocDistA, we add location state features on top of method 2. To extract the location state, we implement an online place learning algorithm on the incoming GPS stream given in Section 3.4. Each adaptive sensing agent will maintain a place database for the user. At each sensing cycle, the most recent GPS coordinate will be compared with all the places in the place database. If it is falling within a radius of 30 meters in one of the learned places, then the corresponding place will be treated as the current location. If none of the places satisfy this condition, the location state will be set as 'unknown'. The place database will be updated on an hourly basis using GPS data collected within the past hour. Each newly formed cluster will be checked against existing ones for potential combinations. If a newly learned place's center is within 30 meter of an existing place's center, they will be combined using the weighted geometry center as the new place center. Using linear approximation enables us to flexibly add new place label as state features. It is also possible to drop places that have not been visited for a long time to make the size of the state space stable. This strategy will account for the dynamic human activity environment that is changing over time.

RL method 4: Combining all state features from the above methods with linear approximation (ComA). Lastly, we combine all state features from the previous three proposed methods and called this approach ComA.

Action Space Design

We generalize the action space from 2 actions (e.g., 'On' and 'Off') in Chapter 4 to K actions. We can encode the K actions to represent the number of time steps to skip in each sensing cycle. Take an action space of $\{0, 1, 2, 3, 4, 5\}$ as example, when the action is 0, sensors are deployed for one time step and no time step will be skipped. Similarly, when the action is 3, sensors are deployed for one time step, followed by skipping sensing for three time steps. By designing the action space this way, we can reduce the amount of sensor operation time to achieve higher energy efficiency, when compared to traditional duty cycling strategy, in which sensors are always turned on and off for chosen fixed times. Within each sensing cycle, the collected sensor data will be leveraged to decide the length of time steps for sensor deactivation.

Reward Signal Design

In Chapter 4, the reward signal is designed using a threshold-based approach. Specifically, we used acceleration threshold for accelerometer, and displacement threshold for GPS, and the reward took either a value of 0 or 1 based on the collected sensing data. There are two challenges with this threshold-based approach. First, it requires specification of a reasonable threshold level, which may not always be well-defined for a particular sensor; second, taking only 0 or 1 as the reward signal may fail to differentiate the magnitude of the 'inertia' levels for those that are above or below the chosen threshold, leading to less efficient learning. In this work, we amend these deficits by proposing a continuous reward signal that can proportionally reflect the changes in the sensing data.

The reward signal is computed by leveraging the sensing data collected in the sensing window of the next sensing cycle. In general, the definition of the reward signal is sensor dependent. Because we want the sensor to be turned off when no changes or lack of 'activity' is detected in the user's context, or to be deployed when otherwise, it needs to consistently reflect this motivation. Let us use R_t' to denote the metric that measures the magnitude of 'inertia' level for any sensor using the sensing data collected at the sensing window of the next sensing cycle. The larger R_t' is, the more intensive the activity is. Thus the reward signal R_t at sensing cycle t is given by the following:

$$R_t = \begin{cases} R_t' - R^r, & \text{when } a_t = 0 \\ -\sum_{k=0}^{a_t} \beta^k R_t' + R^r, & \text{when } a_t = 1,\ldots,\text{K-1} \end{cases} \tag{5.6}$$

Here β is a discounting factor between 0 and 1 that backward estimates the activity level in the time step(s) that the sensor is turned off using the most recent available activity level. When β is small, we have lower confidence in the estimations. When $a_t = 0$, we keep the sensor on in the next time step. If the activity level is high, that reinforces the action, and vice versa. When $a_t \neq 0$, the sensor is left off for a chosen number of time steps. $\sum_{k=0}^{a_t} \beta^k R_t'$ is proportionate to the activity level during this period, and if it is large, we penalize the action by adding a negative sign to it, and vice versa. In both cases, we add a reference level R^r to prevent a constantly positive or negative reward signal, which can lead to explosion of the linear coefficients in the learned policy. R^r can be any arbitrary value within the range of R_t'.

5.4.2 Performance Evaluation

The lack of activity ground truth makes it extremely challenging to evaluate the performance of our proposed adaptive sensing algorithms against the continuous sensing data.

In Chapter 4, we defined accuracy, F-score, percentage of battery saved, and percentage of activity coverage using a threshold-based approach. In order to remove the need of threshold specification, we apply a continuous reward signal, which can not be used to define accuracy and F-score. Thus, we focus only on energy saving and activity coverage based on the following definitions:

- **Percentage of energy saving** (PES) is defined as the number of time steps that the sensors are not turned on when compared to the continuous data stream. Let us denote the total number of time steps in the continuous data stream as I, and the sensor being turned off at time step i as $a_i \neq 0$, we have

$$PES = \frac{\sum_{i \in \{1,\ldots,I\}} \mathbf{1}(a_i \neq 0)}{I},$$

  where $\mathbf{1}(a_i \neq 0)$ is an indicator function that takes 1 when $a_i \neq 0$.

- **Percentage of activity coverage** (PAC) is defined as the proportion of sensed activity level over total activity level in the continuous data stream. We have

$$PAC = \frac{\sum_{i \in \{i: a_i = 0\}} R'_i}{\sum_i R'_i},$$

  where $a_i = 0$ indicates the sensor being turned on at time step i.

5.4.3 Adaptive Sensing in Motion Sensors

In Section 5.4.1 and 5.4.2, we propose our RL adaptive sensing methods, the baseline methods, and performance evaluation metrics in general terms that are applicable to any sensor. However, in this work, we want to narrow our focus down on motion sensors (e.g., accelerometer, gyroscope). In particular, our experiments will be conducted using data from accelerometer in a continuous mobile sensing study [28]. In this section, we will describe all the necessary configurations in various components according to the proposed methods above for accelerometer.

Smartphone embedded accelerometer typically generate sensing stream in the X,Y,Z axes, each referenced to the center of the gravity field on earth, and are perpendicular to each other. Due to the uncontrolled orientation of personal smartphones, we combine the sensing stream from all three axes into the acceleration stream using $\sqrt{X^2+Y^2+Z^2}$. All subsequent acceleration stream at sensing cycle t will be denoted as $\mathbf{X_t}$.

To design adaptive sensing for accelerometer, we choose 1 minute as the length of one time step, the actions to be $\{0,1,2,3,4,5\}$, and the activity level R'_t to be $\sum_{j=0}^{n-1}(X_{j+1} - X_j)$, where n is the number of data points within $\mathbf{X_t}$. $\sum_{i=0}^{n-1}(X_{j+1} - X_j)$ reflects the cumulative changes within $\mathbf{X_t}$. When point to point differences are substantial, the sum of these differences will be large; otherwise, it can take zero if all points are identical, which is highly unlikely even if the user is making repetitive motions. For method 1, we choose N to be 10 in buckets with the following break points: $\{0.01, 0.2, 0.5, 1, 2, 3, 5, 8, 10\}$; M to be 5 with break points:$\{0.1, 0.2, 0.5, 0.9\}$. For method 2, we choose $\mathbf{p} = \{0, 0.1, 0.25, 0.5, 0.75, 0.9, 1\}$. For the reward signal reference R^r, we set it to be the first quartile of $\sum_{i=0}^{n-1}(X_{i+1} - X_i)$ from a randomly drawn small sample of participants.

5.5 Experiments

In this section, we describe our baseline approaches and experiments to address several research questions. We adopt the same dataset as described in Section 4.5.1 for simulations.

5.5.1 Baseline Methods

We use the following baseline methods as comparisons to measure the performances of our proposed RL adaptive sensing methods.

Baseline 1: Random Strategy (RS). The random strategy does not leverage any context information extracted from the collected sensor data to make sensor deployment decision. At each time step, the sensor will be deployed based on a coin flip. Half the times the sensor will be deployed, and the other half the sensor will be turned off.

Baseline 2: Duty Cycling (DC). In the duty cycling strategy, the sensor will be turned on and off alternatively, with our chosen sensing and sleeping windows. In our experiments, these windows are chosen to be both 1 minute for accelerometer.

Baseline 3: Learning Automata (LA). The learning automata strategy makes sensing decision based on a sensing probability, which is continuously being updated based on feedbacks from sensor data collected during the deployed time steps. When the sensor is turned off, no updates to the sensing probability will be made. More details about LA can be found in Section 4.5.2 or [179].

Baseline 4: Adaptive Sampling Using Dynamic Functions (DF). In DF, the authors proposed to apply a set of functions for controlling sensor deployments based on two detected contexts – 'missable' and 'unmissable' events. [178] The idea is that when a 'missable' event is detected, the sleeping interval is increased using a back-off function; when an 'unmissable' event is detected, the sleeping interval is shrunk using an advance function. The back-off functions include linear ($k \times x$), quadratic (x^2), and exponential (e^x); and the advance functions include linear ($\frac{x}{k}$), quadratic ($\sqrt{x}$), and exponential ($log_e x$). We use the dynamic adaptation algorithm from this work as our fourth baseline method. In both baseline method 3 and 4, we use the following simple rule to define 'missable' and 'unmissable' events: if $R_t' \geq R^r$, then we have an unmissable event; otherwise, we have a missable event.

5.5.2 Experimental Settings and Research Questions

There are four parameters in the QLADE algorithm including the initial exploration rate ϵ_0, the step-size (or learning rate) α, the discount rate γ, and the eligibility trace-decay parameter λ. All four parameters fall within a range of 0 and 1. In addition, the defined accelerometer reward signal has a discounting parameter β. Instead of tuning these parameters, for every participant in each method, we randomly choose values for them from the following values: 1) $\alpha = \{0.01, 0.05, 0.1\}$; 2) $\gamma = \{0.05, 0.1, 0.2\}$; 3) $\lambda = \{0.05, 0.1, 0.2, 0.5, 0.8\}$; 4) $\epsilon_0 = \{0.1, 0.2, 0.5\}$; and 5) $\beta = \{0.3, 0.5, 0.7, 0.9\}$. The exploration decaying rate is fixed to be $d = 0.999$.

With the above settings, we want to find answers to the following research questions: 1) How does the design of the state features impact the performance on PES and PAC? 2) How is the performance of the proposed RL methods compared to the baseline methods? 3) How do the various sensing strategies impact the utility of the collected data on predicting social anxiety and daily affect scores? We follow the same prediction features, outcomes, and algorithms in Section 4.5.3 but comparing the prediction performance across five different settings including the ComA, learning automata, dynamic function, duty cycle, and continuous sensing strategies.

In addition, we will vary the action space and compare the impact of different action spaces on the performance of the proposed RL methods on PES and PAC. Specifically, we will investigate different values of K using $K = \{4, 6, 8, 10\}$ in the ComA method to understand how we should design the action space. And lastly, we will also analyze how the different state features impact the action decision in both individual and collective views using visualizations.

5.6 Results

Parameter Tuning. Figure 6.4 shows the average PES and PAC aggregated across different parameter values in the four RL strategies. Since no consistent best parameter values across the four strategies and two metrics, we will choose the set of parameter value based on the results from ComA and PES, and the corresponding best combination of parameter values are: $\lambda = 0.2, \alpha = 0.1, \beta = 0.5, \gamma = 0.1$, and $\epsilon_0 = 0.5$.

5.6.1 Performance Comparisons in Different Strategies

Comparisons within RL Strategies. Figure 5.3 shows the average PES and PAC across the four RL strategies with different state features. All RL strategies obtain similar performance in PES with the entropy-based adaptation method trading-off 0.05 in PES for 0.1 in PAC. Because ComA also contains all the entropy features, and has similar

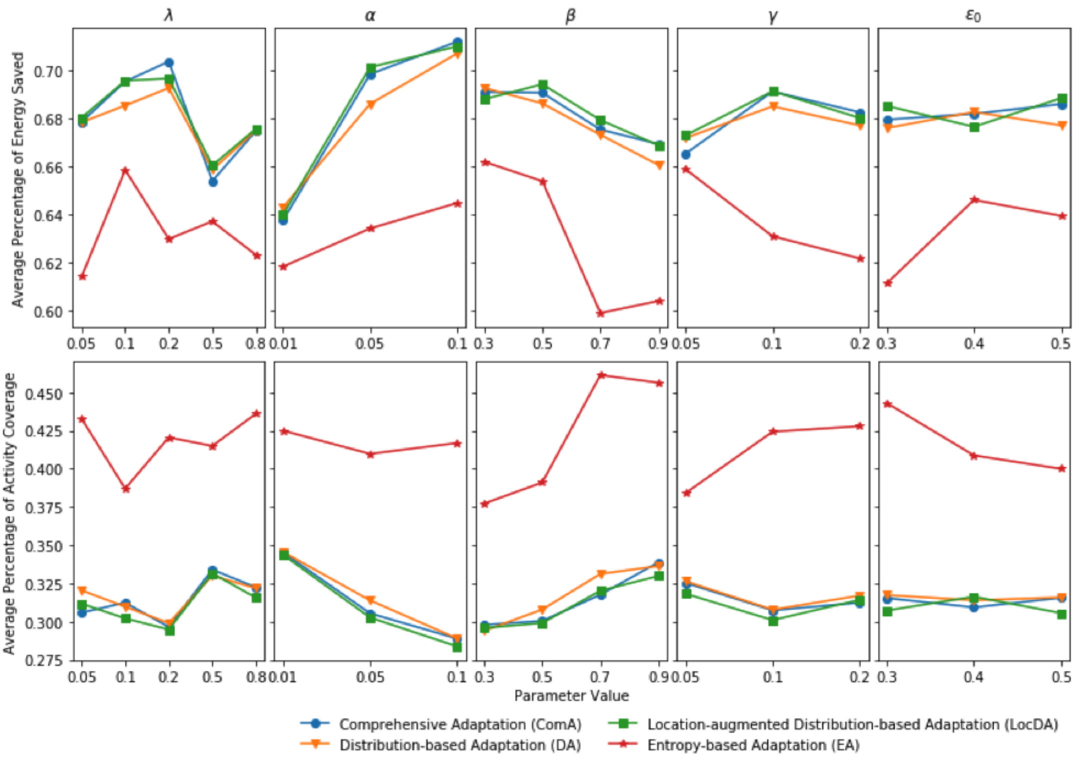

FIGURE 5.2: Parameter Tuning. (a) PES; (b) PAC.

performance to the two distribution-based methods, the entropy features may be dominated by the other features. Augmenting the distribution-based features with location feature that is extracted from GPS data do not lead to performance increase either.

Comparisons between RL strategies and Baseline Methods. From Figure 5.3, we can see that all RL strategies outperform the the baseline methods in PES, but trade-off activity coverage for energy saving. Specifically, the best RL strategy achieves a PES of 0.68, compared to 0.6 from the dynamic function method. The entropy-based strategy is more similar to the dynamic method, with a PES of 0.63 vs. 0.60, and a PAC of 0.42 vs. 0.46, respectively. When compared to the duty cycling method, with the new formulation, the RL strategies significantly save more energy than the duty cycle method. We conclude that no one single strategy attains best performances in both metrics. Thus choosing which strategy becomes consideration in trading off between energy saving and activity coverage. Although the final decision can be dependent on the prediction analysis in Section 5.6.2.

Performance Comparisons across Time. Figure 5.4 shows the average PES and

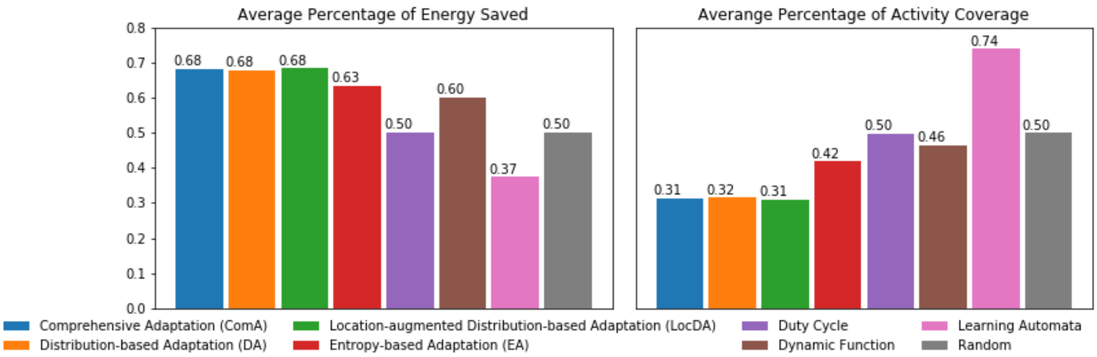

FIGURE 5.3: Performance by strategies. (a) PES; (b) PAC.

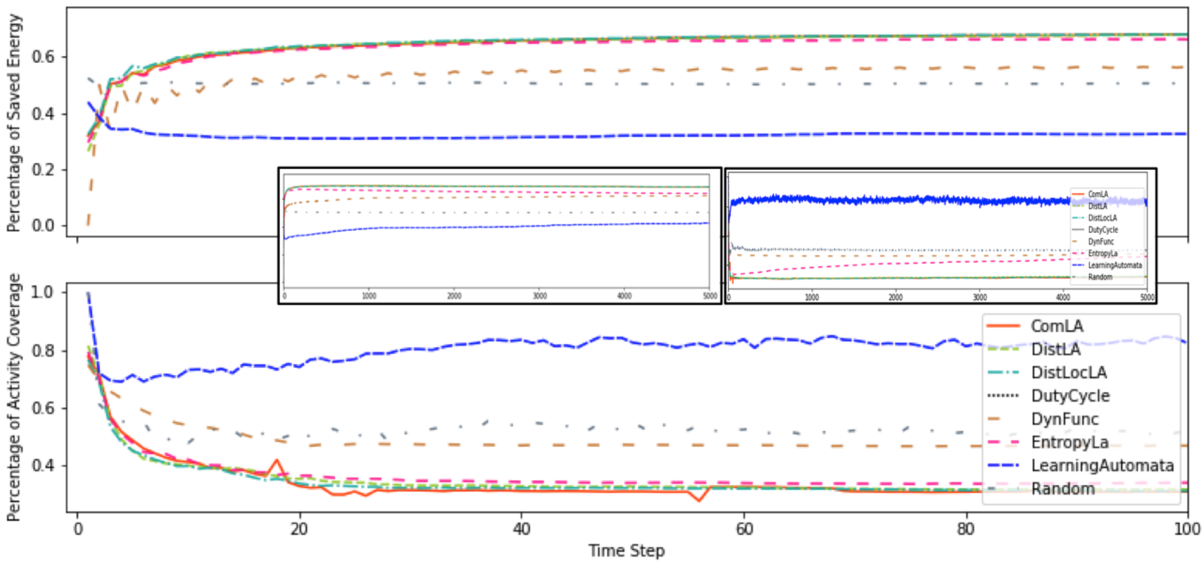

FIGURE 5.4: Strategy Performance over time.

PAC across time in all strategies. All RL strategies maintain a PES level around 0.7 and a PAC level around 0.4. The dynamic function method stabilizes at a PES level around 0.55 and a PAC level around 0.5, while the learning automata method is at a PES level around 0.35 and a PAC level around 0.8. We observe that the average performance stabilizes pretty quickly, which implies that the policies overall converge pretty quickly. Although each policy may be converging at a different speed, or even not converging at all. When we zoom out to a larger time scale, we confirm that the average performance time series are pretty stable. In particular, the learning automata strategy time series fluctuate most over time. This is because when user's activity changes frequently, the sensing probability is also updated frequently, leading to unstable performance.

TABLE 5.1: Comparison of data utility among five different sensing strategies. The social anxiety score has a range of 0-80, and the daily negative affect has a range of 0-100.

		Classification (accuracy)		Regression (mse)	
	sensing strategy	Random Forest	Logistic Regression	Random Forest Regression	LASSO Linear Regression
SIAS	ComA	0.635	0.650	135.5	139.8
	Learning automata	0.627	0.651	136.1	139.8
	Dynamic function	0.583	0.648	138.0	141.1
	Duty cycle	0.625	0.650	136.0	139.8
	continuous	0.601	0.610	135.4	137.4
Daily Negative Affect	ComA	0.786	0.783	516.1	526.3
	Learning automata	0.786	0.782	537.3	525.9
	Dynamic function	0.785	0.783	516.8	529.5
	Duty cycle	0.786	0.786	512.6	524.6
	continuous	0.786	0.783	517.3	526.5

5.6.2 Data Utility Using Various Sensing Strategies

Table 5.1 shows the performance of the various prediction tasks using five different sensing strategies we investigated in this chapter. We can see that all sensing strategies have achieved comparable performance in both classification and regression tasks on predicting both social anxiety and daily negative affect. Specifically, in predicting social anxiety, the ComA RL strategy achieves the second best accuracy at 65%, when compared to the learning automata strategy at 65.1% using logistic regression in classification tasks. In regression tasks, the ComA RL strategy achieves the second best MSE at 135.5, when compared to continuous sensing method at 135.4 using random forest regression. In predicting daily negative affect, the ComA RL strategy has the highest accuracy at 78.6% using random forest in classification tasks; and the second best MSE at 516.1, when compared to the duty cycle strategy at 512.6 using random forest regression.

5.6.3 Personalized Passive Policies

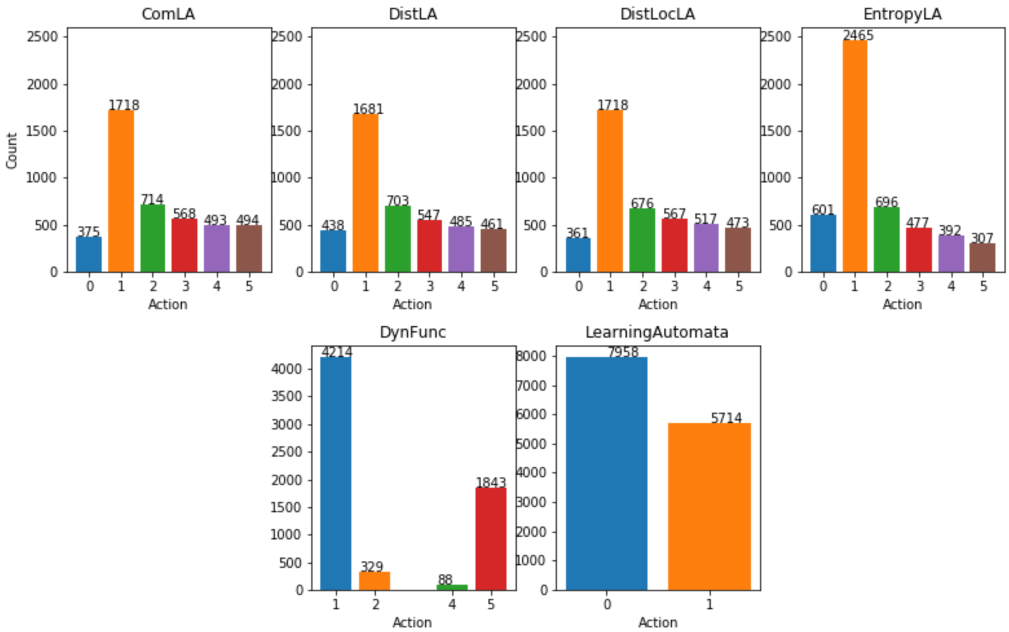

FIGURE 5.5: Distribution of Action Selection by Strategies.

Distribution of Actions by Strategies. Figure 5.5 shows the distribution of actions being made in each sensing cycle aggregated across all participants' data. All strategies except learning automata favor turning off the sensor for 1 time step in the majority of sensing cycles. The distribution among the RL strategies are similar, with Entropy adaptation method more heavily favoring $a_t = 1$. This explains why Entropy adaptation method trades off most energy for activity coverage among the RL strategies. The dynamic function strategy is more divergent in action decision due to the properties of the selected functions, thus we do not see many middle value actions being taken. In the learning automata method, $a_t = 0$ is chosen more of the time, resulting in much higher PAC and lower PES.

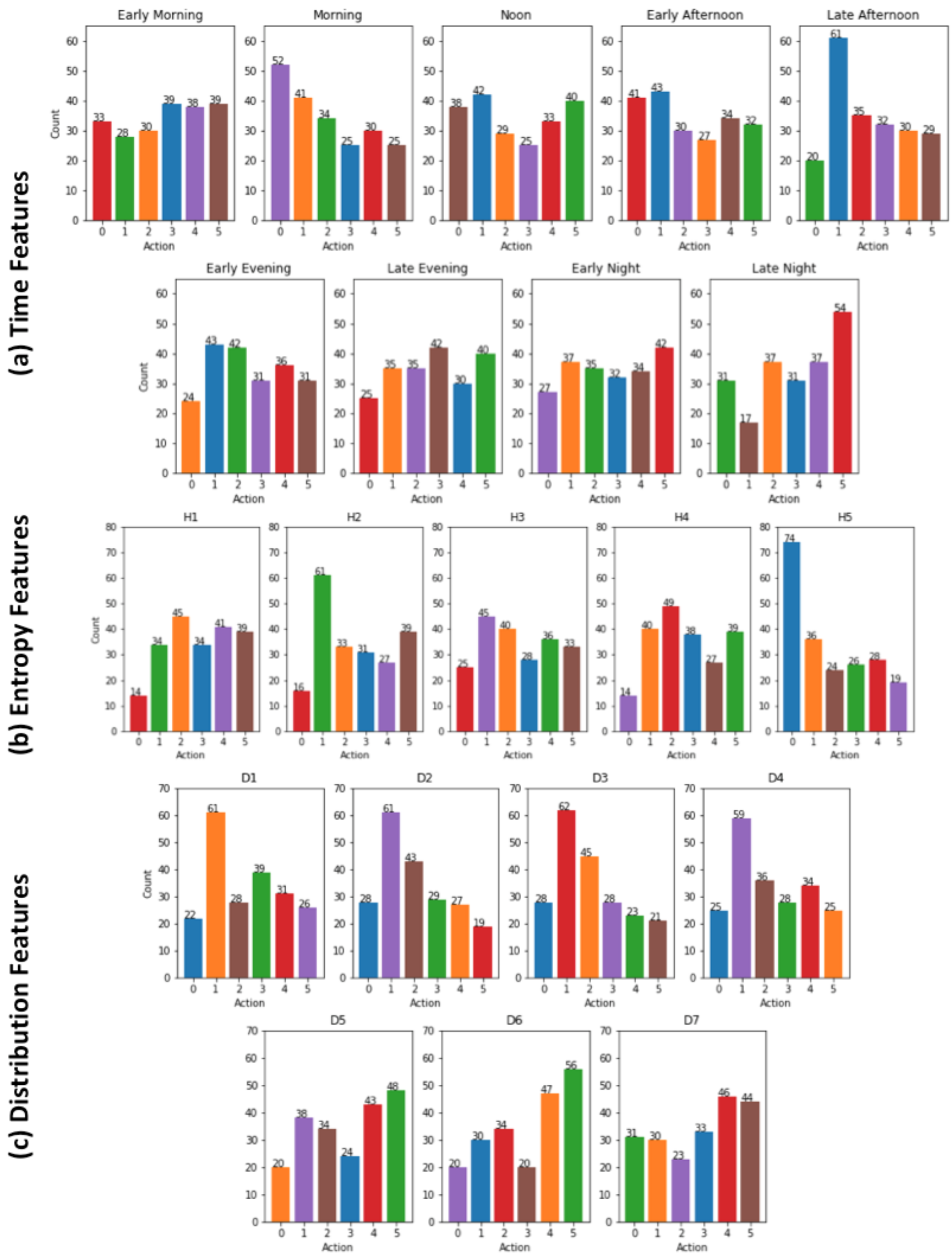

FIGURE 5.6: Overall Policy Profiles.

Distribution of Actions by State Feature Values in ComA. Figure 5.6(a) shows the distribution of actions by time feature values. When time is early morning, noon, early afternoon, early evening, late evening, and early night, the distribution of policies favoring each action is fairly even; when time is morning, 51 policies favor $a_t = 0$; when time is late afternoon, 61 policies favor $a_t = 1$; and when time is late night, 54 policies favor $a_t = 5$. One explanation for morning and late afternoon could be that during these period of time, participants may be most active due to commuting to school and going home. It is self-explanatory that late night favors $a_t = 5$ as during this period of time, people are usually less active, although some people could be heavily using their phones for social network and gaming before going to bed.

Figure 5.6(b) shows the distribution of actions by entropy features. The entropy features represent the information contained in the sequence of accelerometer data being captured in the sensing time window. The higher level the entropy bucket is, the more changes the data contain. In general, if these features are effective, we will expect to see more policies in the higher level buckets favoring lower value actions, and vice versa. In $H5$, 74 policies favor $a_t = 0$, and only 19 of them favor $a_t = 5$. This result meets our expectation. In $H1$, $H2$, and $H3$, relatively few policies favor $a_t = 0$, although most policies in $H2$ favor $a_t = 1$. In $H4$, the most favored action is $a_t = 1$ and $a_t = 2$. Overall, we think these features work as we expect them to contribute to the sensing decision.

Figure 5.6(c) shows the distribution of actions by the distribution features. We observe that in $D1$, $D2$, $D3$, and $D4$, the actions have similar distributions, with the majority of policies favoring $a_t = 1$, while in $D5$, $D6$, and $D7$, more policies leaning towards $a_t = 5$. It is more involved to interpret the underpinning of these features in sensing action decision as this set of features are designed to be a structure representation on the underlying activity. And since these features simultaneously present in each state with different set of values, the frequency of them in these figures represents the number of policies has the biggest values in the coefficients across different actions. Thus larger values in the features favor the higher value actions in $D5$, $D6$, and $D7$, while favor the lower value actions in $D1$, $D2$, $D3$, and $D4$.

Individual Policy Profiles. Figure 5.7 shows the propensity of each state feature value towards the six actions from 0 to 5 as they increase. Each row in the heatmap is a participant's learned adaptive passive sensing policy, while each column is a value in a state feature. For example, for the first participant from the top row, if it is in the morning, then it is more likely the sensing agent will choose $a_t = 2$, if the entropy feature takes a value $H1$, the sensing agent will choose $a_t = 2$ as well. This visualization informs us that each participant's different behavior patterns can lead to different sensing policies being generated by the RL passive sensing agent. It emphasizes the importance of developing personalized passive sensing policies for different users.

5.6.4 Sensitivity Analysis on Action Space

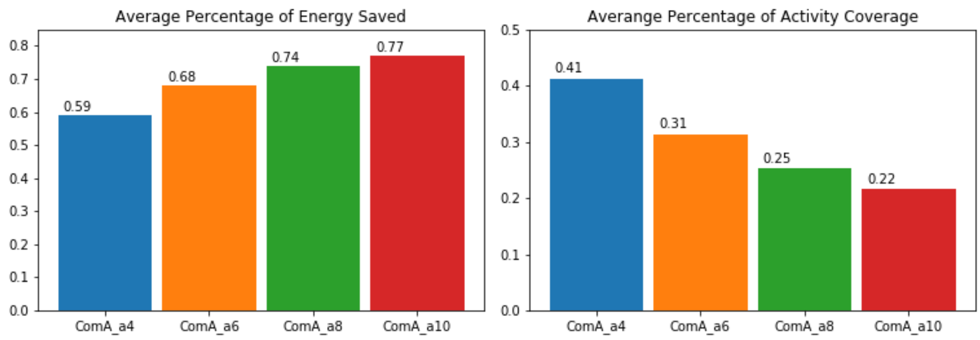

FIGURE 5.8: Sensitivity Analysis in Action Space Designs. (a) PES; (b) PAC.

The above analyses and visualizations adopt an action space $A_t = \{0, 1, 2, 3, 4, 5\}$ with six different actions. However, we do not yet know whether this design is appropriate, and how to choose the optimal action space. Figure 5.8 shows us the performance comparisons in four different action spaces using the ComA algorithm. The observation is quite clear: as the action space is being expanded, it trade offs activity coverage with energy saving. When we increase the action space from $A_t = \{0, 1, 2, 3\}$ with four actions (denoted by ComA_a4), to $A_t = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ with ten actions (denoted by ComA_a10), the PES gradually increases, while the PAC gradually decreases.

5.7 Discussion and Limitations

This chapter is an extension to Chapter 4 to address several limitations. First, we reformulate the adaptive passive sensing problem to enable a larger action space that include actions for choosing the length of sensor deactivation window. This design improves energy efficiency that is previously impossible when we formulate the problem as a middle ground between continuous sensing and duty cycling. Second, we propose three different sets of state features that may be predictive of signal changes in the accelerometer data, while taking into consideration the potential energy overhead posed by extracting them. Lastly, we design a more generalizable reward signal that is free from specification of a 'movement' threshold parameter. With the expansion in the action space, the reward signal also needs to accommodate the 'not trigger' actions (e.g., when $a_t \neq 0$) by estimating the 'movement' change level that is proportional to the length of the 'not trigger' window.

Our newly proposed RL strategies consistently outperform the baseline methods in energy saving, however, by trading off some activity coverage. This prompts us to rethink how to define 'activity' in the 'activity coverage' concept. Currently, we define it as the sum of consecutive acceleration changes for accelerometer data. At this point, we could not come up with an alternative definition. So we have to rely on examining the performance in data utility. With the current lack of global best strategies in both metrics, nonetheless, we can still choose one of these different strategies based on our energy requirement, as retention of participants is always the top priority in any mobile sensing applications, and battery drainage has been shown to be the most serious challenge in past studies that involved over 3000 study participants.

Our current work only addresses the most important limitations from Chapter 4. However, there are more we can do in the future. First, state representation learning is a very important topic in RL with domain areas such as robotics that involve sensory data. [23] In particular, end-to-end deep learning approaches as function approximator can avoid state feature design and potentially achieve higher performance in energy saving, while balance activity coverage and maintain data utility. Though there are downsides to this more complicated approach: 1) it generates more energy overhead during learning; 2) it demands more learning data, and potentially a longer offline training period of time. Overall, it worth exploring this option in further simulations. Second, the proposed methods are examined in isolation to a real mobile sensing system, in which the collected sensing data may be processed or modeled simultaneously. Thus when implementing them, more thoughts are required to design the data pipeline, especially how we coordinate the sensing state (i.e., missing data due to deactivation of sensors or other problems) with the data consumption in preprocessing, feature extraction, and modeling. Most importantly, the proposed strategies need to be evaluated in future deployments in real mobile sensing studies.

5.8 Conclusion

In this chapter, we extend the proposed work in Chapter 4, improving the problem formulation with an expanded action space, proposing new sets of state features, and generalizing the reward signal. Our proposed RL strategies consistently outperform the baseline methods including the dynamic function method, the learning automata method, the duty cycling method, and a random strategy in energy saving. To verify the impacts of these different strategies on data utility, we predict social anxiety and daily negative affect using data collected by the ComA strategy and all baseline strategies except the random strategy, plus the continuous sensing strategy (i.e., using the actual data). The results show no significant differences in the prediction performances among these different strategies.

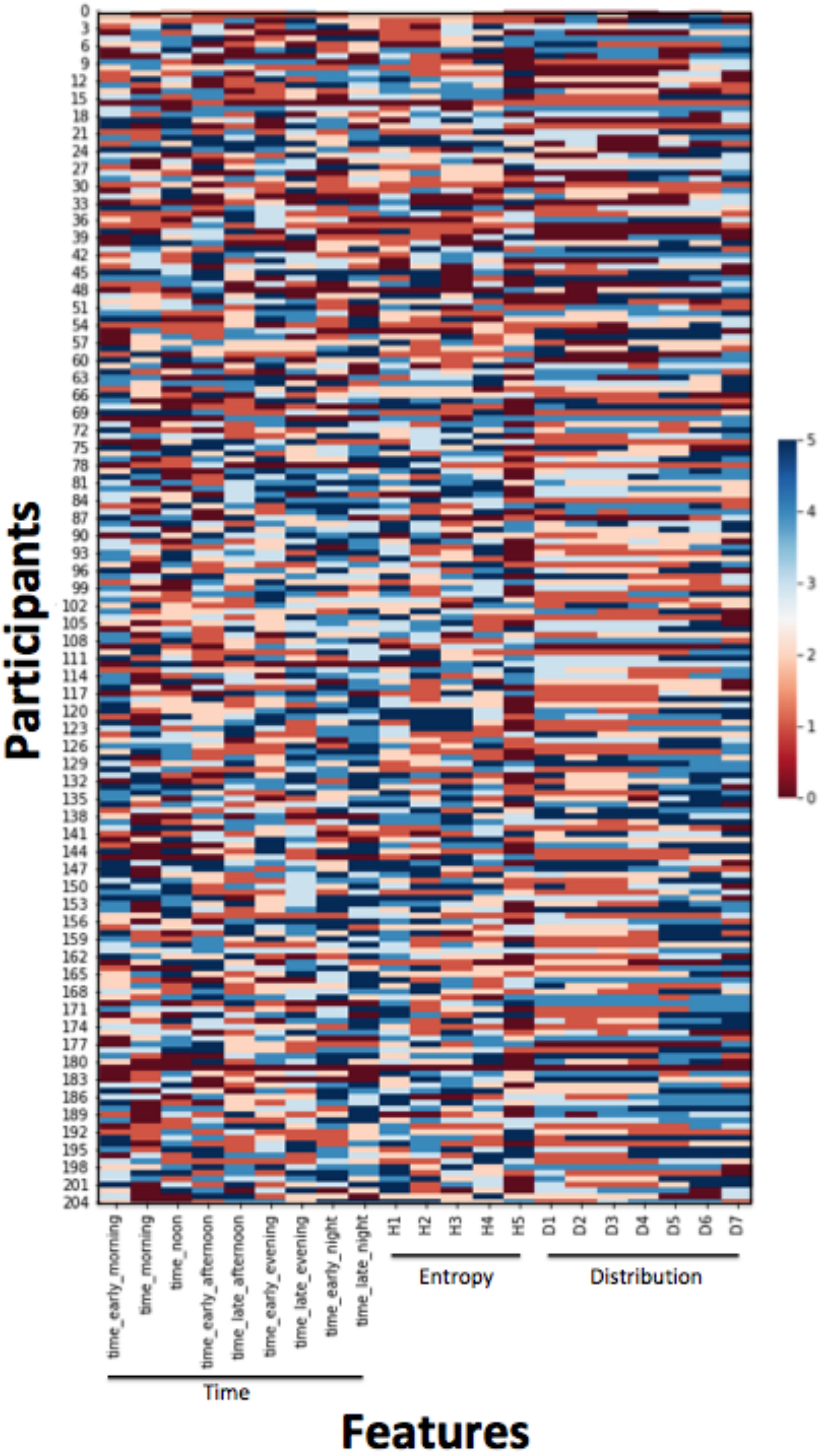

FIGURE 5.7: Individual Policy Profiles.

Chapter 6

Adaptive Active Mobile Sensing In Mobile Ecological Momentary Assessment

Passive data collected by smartphone embedded sensors may not provide critical and accurate state determinants to mobile sensing applications. To amend this limit, active sensing prompts users to directly self-report these metrics. However, active sensing tasks such as mobile Ecological Momentary Assessments (EMAs) require substantial user efforts to complete, leading to low compliance. One major source of low compliance is triggering active sensing tasks at inopportune moments. In this chapter, we propose adaptive active sensing strategies using the reinforcement learning (RL) framework to address the timing and context challenge, aiming to improve long term cumulative compliance. We show that our proposed RL strategies consistently outperform the baseline methods including a random strategy and a supervised strategy in mobile EMA compliance. We also investigate several different techniques including applying a more compact representation in routine state, incorporating motivation as a state feature, and adopting the Dyna-Q framework for better sample efficiency. Although the results are not encouraging, they provide ideas to explore new strategies for further compliance improvement using adaptive active sensing.

6.1 Introduction

In mobile sensing, active sensing refers to the sensing mechanism that requires active involvement in completing sensing tasks. In contrast to passive sensing, users expend time and efforts to provide measurements of key variables that cannot be directly collected or easily derived from passive sensing modalities (e.g., emotion and perceptions). Active sensing modalities include text and voice messages, phone calls, notifications via mobile apps, digital surveys, and dialogue systems etc. Amid these different modalities, mobile Ecological Momentary Assessment (EMA) is a digital surveying method implemented using smartphone. Unlike traditional retrospective survey methods (e.g., telephone/paper/web surveys), EMA frequently collects self-reports to capture the dynamics of human behaviors, while reduce recall bias and enhance ecological validity [211, 99, 232, 210, 78, 183, 17, 16, 215, 186]. It has been dubbed Experience Sampling Methods (ESM), real-time data capture, diary methods (paper diary or semi-paper diary), ambulatory assessment, continuous psycho-physiological, biological, and behavior monitoring, and everyday experience methods. [210, 184]

In the early days when mobile phones were neither "smart" nor pervasive, EMA surveys were carried out in the format of paper diary [160], self-recording using pagers [16], telephone [64], and personal digital assistant [210]. Web-based EMAs were also used [140], but surveys were usually delivered through non-portable terminals such as desktops. Mobile EMA has become the typical choice owing to the increasing ownership of smartphones and accessibility of wireless network in the past decade. [33] Many EMA studies also captured passive sensing data while collecting EMAs, thereby enabling context-aware mobile EMA. [93, 129, 43, 25] Although becoming more convenient, active participation in mobile EMAs still demands substantial efforts from users, and poses significant compliance challenge over time.

Low response compliance in active sensing can be attributed to declining user motivation over time. Existing research has applied human behavior theories to engage and motivate users in active sensing applications (e.g., substance use logging [177] and weight management [218]). While motivation has been an important challenge to address in active sensing, low compliance can result from another significant challenge, inopportune timings and contexts, which could be caused by 1) unavailability at the moment of sensing requests, and 2) interruptions that distract user's attention from his/her current more prioritized task(s). Underlying these causes are the different contextual and cognitive states (e.g., activity, location, time, and stress level) the user is situated. At each active sensing prompting decision point, a user could be at certain location and engaging in certain activities. Given these different contexts, the user may not be available and interruptible, failing to attend and respond to the sensing request. Our goal is to identify opportune moments to trigger active sensing to the users, while not interrupt them in unsuitable moments, thereby achieving higher compliance in the long term.

Adaptive active sensing leverages passive sensing to understand user's context, and based on this understanding, adapts the trigger timings to those moments that are more likely free of interruption and convenient for the user to respond. In addition to being context-aware, adaptive active sensing also need to avoid bias in the collected data that is coming from being selective in trigger timings. [114] In this chapter, we design adaptive active sensing strategies using the reinforcement learning framework under a formulation that reduces bias in data. We propose to model the state of the user using a combination of low-level features including the trigger time, user's location, transportation mode, momentary and hourly activeness, and a high-level routine feature called k-routine that places the user within the context of their frequent daily living patterns. When an active sensing task is not triggered, we design a reward signal that works with our proposed QLADE algorithm to overcome the delayed reward challenge. To understand the impact of motivation, we approximate a user's temporal motivation using a moving fixed window compliance rate as a state feature. A bootstrapping method called Dyna-Q is incorporated into the QLADE algorithm to mitigate the challenge from limited learning samples. All these RL strategies are then compared with a supervised and random baseline strategy.

We evaluate the above proposed methods in simulations using real mobile EMA dataset from 220 college students over two weeks in a mental health study. Reis has categorized mobile EMAs based on the types of triggers, including interval-contingent, signal-contingent, and event-contingent EMAs. [184] Other researchers simply group them as event-based and time-based, while within time-based EMAs, the triggers can be either fixed-time or random-time. [211, 105] We focus only on adaptive active sensing using random-time mobile EMAs. However, our proposed RL strategies can be generalized towards other active sensing modalities.

The structure of the remaining chapter is as following: Section 6.2 reviews related works on active sensing, Section 6.3 provides the problem formulation of active sensing, Section 6.4 describes the user model with all specific state features and the proposed k-routine mining algorithm, Section 6.5 designs the RL active sensing strategies, Section 6.6 and 6.7 present the experiments and their results. Lastly, the chapter is closed with discussions in Section 6.8 and some conclusion marks in Section 6.9.

6.2 Related Works

6.2.1 Mobile Ecological Momentary Assessment

Ecological Momentary Assessment (EMA) is an intensive data collection approach that allows subjects to repeatedly report their experiences in real-time and in situ. EMAs have been preferred over one-off retrospective self-reports to collect longitudinal data in a wide range of research areas such as clinical assessment [78, 16], psychology/cognitive process and their mechanisms [229, 208, 13], and mobile health [99, 116]. It provides a better understanding of dynamics in human behavior/experience over time and across situations [196]. Mobile EMA leverages smartphone's portability and closed proximity with their users to effectively collect active data, which are usually augmented by passive data to provide more contextual understandings about the user. [25]

Response compliance problem in mobile EMA has been noticed and explored by various researchers in recent years [145, 166]. Most of the existing works focus on understanding different sets of factors that may influence mobile EMA response compliance. Serre et al. [194], Sokolovsky et al. [201], and Broda et al. [34] studied impacts of demographic and self-reported contextual factors on EMA response compliance. Vhaduri et al. systematically investigated the impacts of various design factors on response compliance and quality of the collected data in a mobile EMA setting. [217] Comparing to our proposed work, these studies did not leverage passive sensing capabilities to understand contextual states of users but relied on self-reports and pre-specified triggering schedules from EMAs. In addition, they also did not intervene with any strategies to improve user response compliance. A third group of researches by Vhaduri [216], Markopoulos [141], and Hofmann [87] investigated the impacts of delivery timing and reminders on EMA response compliance. Their strategies using user chosen delivery times and regularly dispersed reminders are not adaptive to users' changing contexts. None of the above works proposed adaptive systems that can enable continuous learning and intervention to improve active sensing (e.g., mobile EMAs) response compliance and user experience.

6.2.2 Interruption Management

The ubiquitous computing community has conducted numerous research on how to deliver emails [88], text messages [168], phone calls [22]. The goal of these works is to identify opportune moments of users' routine lives to avoid interruptions that may disrupt their ongoing tasks. Another thread of research in mobile notification interruption management focuses on application of context-awareness to identify opportune moments for notification delivery [164, 146, 158, 142]. They found that contents, social relationship, and physical activity level [146], location and time [164], current task [165],current activity [86, 157, 68], psychological traits [147] obtained from both passive and active sensing can be leveraged to predict opportune moments for interruptions.

Similar to our proposed work, this research leveraged both passive (e.g., smartphone embedded sensors) and active sensing (e.g., mobile EMAs) to learn the users' contextual states and predict whether a moment is interruptible. However, their approach using activity recognition and rule-based reasoning did not take into account the dynamical interactions between the users and their environment. Most significantly, the response of notifications is unreliable due to the fact that most notifications may not require active user response. Further, using removal of notifications from the notification center as an indicator of a response could bias the prediction outcome. Even when a user response is desired (e.g., answering phone calls), active sensing response may be more burdensome and less motivating (i.e., picking up a phone call from family member is more motivating than taking a mobile survey).

6.2.3 Just-in-time Adaptive Intervention

Our proposed work is closely related to behavior intervention because by adapting timings and contexts in active sensing, we aim to modulate participants' active sensing response behaviors. From this perspective, a special type of behavior interventions, namely Just-in-time Adpative Intervention (JITAI), is of interest to us. JITAI provides supports when the users are most in need through adaptation in the delivery timing and intervention contents. Frameworks guiding the Design of JITAIs with mobile technologies have been proposed by Nahum et al. [154]. However, implementation of adaptive and interactive systems that deliver JITAI is still very challenging, and most existing works either involved only conceptual developments and/or proof-of-concept, or were not truly adaptive (e.g., recipient chosen timings, or after help/support is requested through responded EMAs [84]).

Timms et al. proposed to use control engineering framework to model mobile interventions [111, 207], while Kelly et al. proposed to use RL to facilitate the delivery of intelligent real time treatment (iRTT) with continuous self-reports through mobile EMAs [100]. Both of these proposed methods are dynamic and leveraging active sensing to obtain real-time understandings about the users, which will serve as controlled variable(s) in the control engineering framework or reward signal in the RL framework to determine intervention decision and timing of delivery. We propose to use RL framework in adaptive active sensing to address the same delivery timing challenge in JITAI. After all, if users are not responding to mobile EMAs in a timely fashion, decisions of intervention contents that are relying on self-reports will not be possible. However, we will leverage passive sensing capabilities in smartphones to obtain contextual states from the participants instead of using self-reports.

6.3 Adaptive Active Sensing

Adaptive active sensing leverages passive sensing data streams to understand the users' context, and interacts with the users for subjective data collection. The main goal of adaptive active sensing is to improve user's active sensing compliance while reduce unnecessary interruptions to the users in data collection. To achieve this goal, the adaptive active sensing problem can be formulated as selection of timings for active data collection within given interruption budget to obtain maximum user compliance. The interruption budget refers to the allowable active sensing tasks that we can trigger on a given time frame. For example, in a study that mobile EMAs are collected three times daily, the interruption budget is three.

Imposing interruption budget is important to avoid over burdening users and maintain user compliance. [122, 120] We will also need to spread the active sensing tasks as evenly as possible across a given time window to avoid 'contextual dissonance', which biases the collected data due to context selection. [114] We follow a classical approach to split each day into some number of blocks, and within each block randomly select a time for active sensing decision. [122] For example, in our social anxiety study [28], a day was split into 6 2-hour windows from 9am - 9pm, and within each window, a time was randomly drawn to trigger a momentary state affect survey. As a result, this design achieved a budget constraint of 6 evenly distributed active sensing tasks daily. If we have a lower active sensing budget, we can keep the same 6 2-hour windows, and choose 3 from them to trigger active sensing tasks. Our adaptive strategies will then determine 3 out of the 6 randomly selected times from those 2-hour windows for active sensing tasks. This design meets the daily active sensing constraint while minimizes the bias in the collected active data. In order to guarantee triggering exactly 3 active sensing tasks daily, we take into consideration the opportunity costs and let the RL agent incorporate this knowledge through learning from each episode. For example, if the RL agent decides not to trigger active sensing tasks in the first three opportunities, it has no choice but to trigger them in the remaining three opportunities in order to meet the budget. When the RL agent triggers three tasks before the end of the daily cycle, later assessment moments will not be considered any more.

Adaptive active sensing can be formulated as a discrete time episodic sequential controlling problem. An episode is often chosen to be a targeted time frame (e.g., from 9am to 9pm) within a day. Within each episode, we follow the above design, and apply the RL framework to develop sensing policies that assess value of each assessment moment for active sensing trigger decision. The user contexts will be extracted using passive data stream, and trigger decision will be made based on the learned contexts. We repeat this process until the interruption budget runs out, at which point no more active sensing will be triggered and the episode ends.

The main challenge of adaptive active sensing lies in context recognition for the assessment of user's state. There are two aspects to this challenge. On one hand, determining what contextual factors are important to users' response compliance and interruptibility is not trivial. Fortunately, many existing works in notification and interruption managements can provide us insights on state feature design. On the other hand, the modeling process is online and interactive between the application and the users. This implies that data accumulate over time, and users' behaviors and environments constantly change, making it very challenging to model the user environments. In this chapter, we propose and test new adaptive active sensing algorithms under the reinforcement learning

TABLE 6.1: Momentary state features on the selected active sensing times.

Features	Description
Time	early morning (8-10am), morning (10am-12pm), noon (12pm-2pm), early afternoon (2-4pm), late afternoon (4-6pm), early evening (6-8pm).
Location	unique place labels that are learnt by a tempo-spatial clustering algorithm [97].
Speed	being still, walking, running, being in vehicle using average speed cutoffs $(0.1, 1, 5)m/s$. Speed is calculated based on average distance between consecutive GPS coordinates within the 10 minutes time window divided by their corresponding time spans.
Hourly Activeness	proportion of time average acceleration in 5 minute windows within the past hour is beyond 0.2.
Momentary Activeness	proportion of time average acceleration in 1 minute windows within the past 10 minutes is beyond 0.2.

framework. The details of our proposed methods are provided in the next two sections.

6.4 Modeling User Contexts

A high level knowledge about users' routine contexts (e.g., resting at home after playing basketball for 2 hours with friends in the gym) can be very useful in understanding their behaviors (e.g., active sensing compliance). Given that momentary contexts are also critical in the active sensing response decision process, combining both the high level routine knowledge with the low level momentary context may lead to better prediction of users' active sensing compliance. To achieve this goal, we propose a two-level context model with the low-level being the momentary state, and the high-level being the routine state.

In the momentary state, we capture the time, location, speed, hourly activeness level, and momentary activeness level. Table 6.1 defines these momentary state features. They will also be applied to map the momentary state with the current routine state.

In the routine state, we learn the high-level routines of the users on a daily scale using our proposed multi-level frequent life-block generation algorithm. The concept of routine is similar to itemset, a concept used in classic association rule mining algorithm [4]. We first define a basic information unit that describes the whereabouts and activities of

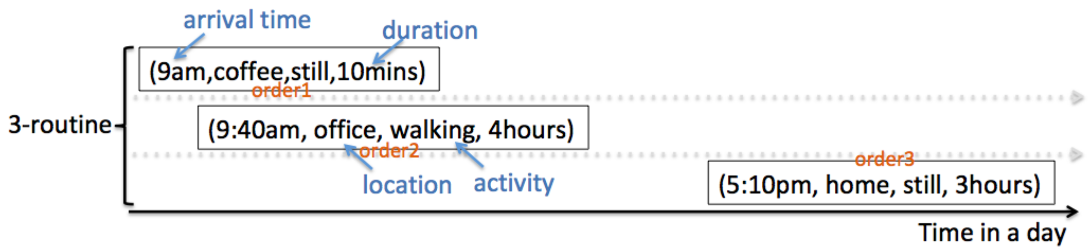

FIGURE 6.1: Illustration of an example 3-routine.

a user at a given time, and we call such an information unit 'life-block'. A life-block generally consists of time, location, physical activity based on speed, duration, and any other available contexts that can be extracted through passive sensing data and other mobile phone usage logs. We call a routine consists of k life-blocks k-routine in analogy to k-itemset in association rule mining. Without loss of generality, we denote a life-block as (t, loc, act, d) using time (t), location (loc), physical activity based on speed (act), and duration (d) in our examples below. An example of a 1-routine could be: **($9:30$am, office, walking, 4hours)**, and a 2-routine could be: **(9am, gym,still, 30 mins), ($9:30$am, office, walking, 4hours)**. Life-blocks within k-routines do not have to be consecutive in time, but have to be sorted by time. For example, Figure 6.1 shows a 3-routine with a significant gap between life-block 2 and 3. Note that higher order k-routines are formed by combinations of lower order k-routines. For each learned unique routine, we assign it a unique code for reference. After being mapped with the momentary state, the routine state will be represented using this assigned code, making this routine state feature categorical. In the next few sections, we provide the details on how we generate these daily k-routines, and map them to the momentary state.

6.4.1 Mining K-routines

The process of constructing life-blocks is similar to that of extracting the momentary state. In order to capture more fine-grained temporal patterns, we divide all input data into ten minute segments, and extract the location, speed from each segments. In cases where the user has been in more than one location or one speed category within one segment, we adopt the place or speed category with most data points. If the user is in transition from one place to another, the place label would be denoted as 'in-transition'. For consecutive segments that the users have same place and speed value combinations, they will be concatenated into a life-block with the time being the arrival time at the place, and duration being the number of segments multiply by 10 minutes. From this procedure, an entire day of mobile sensing data will be converted into a trajectory of life-blocks.

Focusing on daily level, we treat each life-block as an item, and all life-blocks within a day as an ordered transaction, in analogy to the concepts in classic association rule mining algorithm. However, we can not directly apply frequent itemset generation algorithm in existing association rule mining methods to mine k-routines due to two key differences. The first issue relates to the time order of life-blocks within a day. Life-blocks are sorted by time to form a k-routine. The second issue relates to the availability of data being an incremental online process. Data are made available throughout each day, and the algorithm will process the data at 10-minutes increments to generate daily life-block sequences. At the same time, whenever a new life-block is constructed, the k-routine database will be updated to reflect the changes. We propose the k-routine mining algorithm in Algorithm 5.

In Algorithm 5, ***K-routines*** and ***Places*** are the accumulated learned k-routines and visited unique places up to time t. ***LBs*** are the life-blocks of the same day up to time t, and *plb* is the pending life-block that is being generated and maintained at time t. ***GPSs*** are newly available GPS points in a ten minute segment starting at time t. Algorithm 5 is an online algorithm that will be repeatedly called every 10 minutes.

The number of life-blocks on each day is dependent upon the number of context features that are used to define them, and the number of unique values in each context feature. However, due to the variation in arrival times, uncertainty in visited places (i.e., new places being visited over time), and duration staying at each place, we cannot reliably estimate its per day computation complexity. Assuming a day has K life-blocks, without limiting the order of k-routines, this will result in 2^K k-routines with $k = 1, 2, \ldots, K$. If we limit k to be $\hat{k}$, then the total unique k-routines on the day will be $\sum_{i=1}^{\hat{k}} C_K^i$, where

Algorithm 5 K-routine Mining Algorithm.

Input: $\boldsymbol{K\text{-}routines}$, $\boldsymbol{Places}$, $\boldsymbol{LBs}$, plb, $\boldsymbol{GPSs}$, t.
Output: $\boldsymbol{K\text{-}routines}$, $\boldsymbol{Places}$, $\boldsymbol{LBs}$, plb, t.

$act = extractAct(\boldsymbol{GPSs})$
$\boldsymbol{Places}.update(\boldsymbol{GPSs})$
$loc = extractLoc(\boldsymbol{GPSs})$
if $plb.loc == loc$ and $plb.act == act$ **then**
  $plb.update()$
else
  $\boldsymbol{LBs}.append(plb)$
  $\boldsymbol{K\text{-}routines}.update(\boldsymbol{LBs})$
  Clear plb.
  $plb = (t, loc, act, 10mins)$
end if
if $t + 10$mins remains in the same day **then**
  $t = t + 10$mins
else
  $t = t + 10$mins
  Clear $\boldsymbol{LBs}$ and plb.
end if
return $\boldsymbol{K\text{-}routines}$, $\boldsymbol{Places}$, $\boldsymbol{LBs}$, plb, t.

C_K^i is the combination of choosing i life-blocks from K life-blocks. For example, if we limit k to be 3, then we will have $C_K^1 + C_K^2 + C_K^3$ unique k-routines.

6.4.2 Merging K-routines

After obtaining these unique k-routines on a new day, we need to merge them with those learned in the past days if they are similar to each other. We define similarity using the following rules:

1. k_1-routine and k_2-routine are similar only if $k_1 = k_2$.
2. If condition 1) is met, k_1-routine and k_2-routine are similar only if each pair of life-blocks with the same order is similar.
3. Two life-blocks are similar if their place and speed (or activity) are the same, and their arrival time and visiting duration are similar.
4. Let (t, d) denotes the values of arrival time and visiting duration. (t_1, d_1) and (t_2, d_2) are similar if the Euclidean distance between them is smaller than a chosen threshold.

6.4.3 Mapping K-routines

The k-routines we learn will be applied to augment our state representation in adaptive active sensing strategies. To achieve this goal, we need to map them to the momentary states. Below we describe how to map the learned k-routines to the momentary state.

Consider at moment t, we want to assess whether it is an opportunistic time to collect active data from the user. We take the following steps to map moment t to the learned k-routines up to time t:

1. Let $t.arrival$ and d denote the arrival time and duration of a life-block. Existing k-routines will be filtered out if t does not fall in $[t.arrival, t.arrival + d]$ with $t.arrival$ and d referrng to the arrival time and stay duration in the last life-block in a k-routine.

2. The remaining k-routines satisfying the above condition will be filtered out if the momentary location and activity are not the same with those associated with the last life-block in each of them.

3. For k-routines with $k > 1$, we apply the same procedure as in merging newly mined k-routines with existing ones, on all life-blocks other than the last life-block against the life-blocks on the day prior to t. We choose the longest k-routine that survives the above filtering conditions as the routine state associated with moment t.

4. When no k-routines survive the above tests, we assign 'new routine' as the routine state.

6.4.4 Visualizations of K-routines Learned from Real Data

Figure 6.2 shows two different views of a selected participant's 1-routines. Note that higher order k-routines are formed by all unique 1-routines. The top view shows the location, the duration, and time of day of each 1-routine for this selected participant. Each bar represents a 1-routine, and they are ordered by frequency. The right hand side barplot shows the frequency of them. The bottom view shows the 1-routines by locations. Each

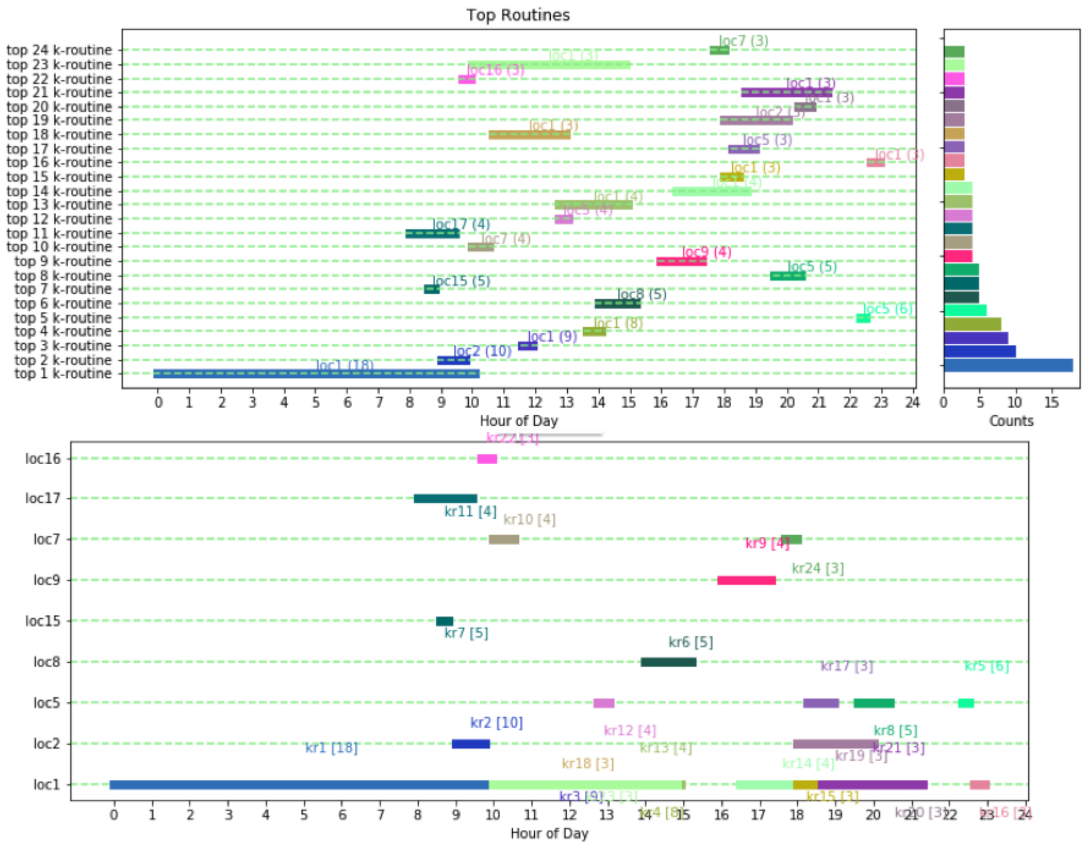

FIGURE 6.2: 1-routines of a selected study participant. The number in bracket is the frequency of the 1-routines.

location can form different 1-routines based on time of day and duration. For example, location 1 contains several different 1-routines with different frequencies, some of which are overlapping. We can see that this participant spent most of his time between midnight to 10am at location 1. We can infer that this location is his/her apartment or dorm.

In the next section, we incorporate momentary state and routine state in our adaptive active sensing methods using the reinforcement learning framework.

6.5 Adaptive Active Sensing Methods

In order to implement adaptive active sensing, we apply the RL framework, which specifies the user state, action that will be taken at a given state, and a reward signal that shapes the policy that selects optimal action at a given state. We propose several strategies with different state designs, while keeping the same actions and reward signal. Below we provide the details about these proposed methods, performance evaluation, and adaptive active sensing using mobile EMA.

6.5.1 Reinforcement Learning Methods

We apply a modified version of QLADE, which is developed in previous chapters on adaptive passive sensing (Chapter 4 and 5). It is shown in Algorithm 6. Compared to passive sensing, active sensing requires active involvements to complete the sensing tasks, and thus has to consider interruption budget on each sensing episode. Also, We have no immediate reward signal when the action is 'not trigger' the active sensing task, because no response information will be available. Below we provide the design details in the RL components to address these challenges.

State Design

We propose several different state feature sets including momentary state features as described in Table 6.1, first order routine feature, second order routine feature in two different encoding schemes, a motivation feature using the fixed-size moving window compliance. To compare the marginal effectiveness of each feature set, we combine them incrementally to create five different RL strategies including: 1) RL with momentary state features; 2) RL with momentary and first order routine state features; 3) RL with momentary and second order routine state features; 4) RL with momentary and a more compactly encoded second order routine state features; 5) RL with momentary, motivation, and a more compactly encoded second order routine state features. The difference between the compact and non-compact second order routine representation lies in how k-routines are encoded. In the non-compact encoding, a k-routine is represented by its routine ID; while in the compact representation, a k-routine is represented by all the routine IDs of the first order routines that form the k-routine. In the situation when two different k-routines are applied as the routine state feature, some same-order life-blocks within them could be the same, and will become the same routine state feature. Thus the corresponding coefficient can be updated more often. This is the reason why the compact encoding is more efficient in learning the impact of the proposed k-routine state features.

Action Design

The action space in adaptive active sensing can include only two actions, 'trigger' and 'not trigger' the active sensing task; or more than two actions that expand the 'trigger' action into 'trigger' with different modalities using sound, vibration, flash lights etc. In this study, we consider only two actions – 'trigger' and 'not trigger'.

Reward Design

We design the reward signal in the following way: it takes a binary value when we trigger an active sensing task with the following conditions: 1) if the task is completed, it receives a positive value 1; if the task is not completed, it receives a negative value -1. When we do not trigger an active sensing task, the reward signal is more involved because we will not directly receive any feedback as if we would have triggered active sensing task. To address this challenge, we need to estimate whether the 'not trigger' decision is beneficial at the end of each day based on how many completed active sensing tasks we have received for the day. If all triggered active sensing tasks are completed, we want to reinforce these decisions in their associated states. In contrary, if we end up having fewer completed active sensing tasks than the number of triggered ones, we want to weaken these decisions in their associated states. Let $s_i^{nt}, i = 1, \ldots, m$ denote the states associated with the 'not trigger' actions on a given day, and $\boldsymbol{w}_i^{nt}, i = 1, \ldots, m$ denote the associated coefficients. We simply reinforce or weaken the coefficients associated with each state feature in s_i^{nt} by $\beta|\boldsymbol{w}_i^t|$, a proportion of the weight coefficients corresponding to the 'trigger' action. The overall reward function is given below:

$$r_t = \begin{cases} 1 & a_t = \text{trigger } \& \; task = \text{completed} \\ -1 & a_t = \text{trigger } \& \; task = \text{not completed} \\ \beta|\boldsymbol{w}_i^t| & a_t = \text{not trigger } \& \text{ all tasks are completed} \\ -\beta|\boldsymbol{w}_i^t| & a_t = \text{not trigger } \& \text{ not all tasks are completed} \end{cases}$$

In Algorithm 6, Line 15 to 20 keep track of all required components for updating the 'not trigger' action value function at the end of the episode, and Line 32 to 36 update the 'not trigger' action value function after the episode ends.

Experience Replay Using Dyna-Q

Due to the limited study time in our current dataset, we may not have sufficient data to train RL policies that can effectively guide active sensing deployment. To address this challenge, we apply a RL framework called Dyna-Q, which integrates planning with learning. Section 2.4 provides some brief information about the concept of planning and learning, and the Dyna-Q framework. We simply adopt a bootstrapping sampler, in which all past episodes including the current one are randomly drawn and replayed to update the policy. In our implementation, we replay ten times at the end of each day to boost the sample size for policy updates. And we combine all available state features with Dyna-Q to be a sixth RL strategy in our simulation evaluations.

6.5.2 Performance Evaluation

An efficient adaptive active sensing strategy will improve the compliance in active sensing tasks. We will measure the active sensing compliance using the following compliance metrics:

- Daily compliance (DC). DC is calculated based on number of all the responded triggered active sensing tasks (e.g., mobile EMAs) divided by number of all triggered tasks on each day.

Algorithm 6 Adaptive Active Sensing Using Q-learning with Linear Approximation and Decaying Exploration.

Input: $\boldsymbol{S}, \boldsymbol{A}, \gamma, \lambda, \alpha, \epsilon_0, d$.
Output: $\boldsymbol{w}^a$, $a \in \boldsymbol{A}$.

Initialize $\boldsymbol{w}^a$ and $\boldsymbol{e}^a$ for each $a \in \boldsymbol{A}$
Set $\boldsymbol{S}^{nt}$, $\boldsymbol{E}^{nt}$, $\boldsymbol{W}^{nt}$, $\boldsymbol{W}^{i+1}$, $\boldsymbol{S_{i+1}}$, $\boldsymbol{A_{i+1}}$ to be Φ.
for all $t = 1, 2, \ldots$ until termination within an episode **do**
  Observe s_t.
  if s_t is not terminal state **then**
    Take $a_t \sim \epsilon$-greedy with $\arg\max_{a \in \boldsymbol{A}} \boldsymbol{\Phi}(s_t, a)^T \boldsymbol{w}^a$.
    Transition to s_{t+1}, and take $a_{t+1} \sim \epsilon$-greedy with $\arg\max_{a \in \boldsymbol{A}} \boldsymbol{\Phi}(s_{t+1}, a)^T \ \boldsymbol{w}^a$.
    $\boldsymbol{e}^{a_t} = \boldsymbol{e}^{a_t} + \boldsymbol{\Phi}(s_t, a_t)$
    if $a_t \neq$ Not Trigger **then**
      $\delta_t = r_t + \gamma \boldsymbol{\Phi}(s_{t+1}, a_{t+1})^T \boldsymbol{w}^{a_{t+1}} - \boldsymbol{\Phi}(s_t, a_t)^T \boldsymbol{w}^{a_t}$
      for all $a \in \boldsymbol{A}$ **do**
        $\boldsymbol{w}^a \longleftarrow \boldsymbol{w}^a + \alpha \delta_t \boldsymbol{e}^a$
        $\boldsymbol{e}^a \longleftarrow \gamma \lambda \boldsymbol{e}^a$
      end for
    else
      Append s_t to $\boldsymbol{S}^i$, $\boldsymbol{e}^{nt}$ to $\boldsymbol{E}^i$, $\boldsymbol{w}^{nt}$ to $\boldsymbol{W}^i$, $\boldsymbol{w}^{a_{t+1}}$ to $\boldsymbol{W}^{i+1}$, s_{t+1} to $\boldsymbol{S}^{i+1}$, and a_{t+1} to $\boldsymbol{A}^{i+1}$.
      for all $a \in \boldsymbol{A}$ **do**
        $\boldsymbol{e}^a \longleftarrow \gamma \lambda \boldsymbol{e}^a$
      end for
    end if
  else
    Take $a_t \sim \epsilon$-greedy with $\arg\max_{a \in \boldsymbol{A} \setminus \mathrm{nt}} \boldsymbol{\Phi}(s_t, a)^T \boldsymbol{w}^a$.
    Observe r_t, transition to s_{t+1}.
    Take $a_{t+1} \sim \epsilon$-greedy with $\arg\max_{a \in \boldsymbol{A}} \boldsymbol{\Phi}(s_{t+1}, a)^T \boldsymbol{w}^a$.
    $\boldsymbol{e}^{a_t} = \boldsymbol{e}^{a_t} + \boldsymbol{\Phi}(s_t, a_t)$
    $\delta_t = r_t + \gamma \boldsymbol{\Phi}(s_{t+1}, a_{t+1})^T \boldsymbol{w}^{a_{t+1}} - \boldsymbol{\Phi}(s_t, a_t)^T \boldsymbol{w}^{a_t}$
    for all $a \in \boldsymbol{A}$ **do**
      $\boldsymbol{w}^a \longleftarrow \boldsymbol{w}^a + \alpha \delta_t \boldsymbol{e}^a$
      $\boldsymbol{e}^a \longleftarrow \gamma \lambda \boldsymbol{e}^a$
    end for
  end if
  for $i \in \mathrm{range}(|\boldsymbol{S}_i|)$ **do**
    Set $s_i = \boldsymbol{S}_i[i], a_i = \mathrm{nt}, s_{i+1} = \boldsymbol{S}_{i+1}[i], a_{i+1} = \boldsymbol{A}_{i+1}[i], \boldsymbol{w}^{a_i} = \boldsymbol{W}_i[i], \boldsymbol{w}^{a_{i+1}} = \boldsymbol{W}_{i+1}[i], \boldsymbol{e}^{a_i} = \boldsymbol{E}^i[i]$.
    $\delta_i = r_i + \gamma \boldsymbol{\Phi}(s_{i+1}, a_{i+1})^T \boldsymbol{w}^{a_{i+1}} - \boldsymbol{\Phi}(s_i, a_i)^T \boldsymbol{w}^{a_i}$
    $\boldsymbol{w}^{a_i} \longleftarrow \boldsymbol{w}^{a_i} + \alpha \delta_i \boldsymbol{e}^{a_i}$
  end for
  if $d\epsilon < 0.1$ **then**
    $\epsilon \longleftarrow 0.1$
  else
    $\epsilon \longleftarrow d\epsilon$
  end if
end for
return $\boldsymbol{w}^a$, for each $a \in \boldsymbol{A}$

- Time constrained daily compliance (TCDC). TCDC is calculated based on number of all the active sensing tasks that are responded within a 10 minute window divided by number of all triggered tasks on each day.

- Cumulative compliance (CC). CC is calculated based on the number of cumulative responded triggered active sensing tasks divided by the number of cumulative triggered tasks up to the current day.

- Overall compliance (OC). OC is the final compliance calculated based on number of all responded triggered active sensing tasks divided by number of all triggered tasks during the simulation.

These metrics are not mutually exclusive with each other. Specifically, the overall compliance reflects the ultimate compliance rate, while ignoring the daily differences. However, it is also important to maintain acceptable daily compliance level as the data can be more representative across time during the data collection. In some application scenarios, when the active sensing tasks are time sensitive, the time-constrained daily compliance is also critical. The cumulative compliance reflects the overall compliance over time.

6.5.3 Adaptive Mobile EMA

We implement our proposed adaptive active sensing strategies in the context of mobile EMAs. A typical mobile EMA scenario aims to collect momentary data throughout each day. Without any constraints or concerns, we can schedule and trigger many more mobile EMAs, and hope that the users will respond to a sufficient number of them. However, this is not realistic, and we want to limit the number of triggers to m on a given day. One way to reduce sampling bias due to contexts related to compliance is to break down a day into windows, and we randomly select a time for EMA delivery within each window. With this setting, assume we break down the day into M windows with $M > m$, our goal is to choose m out of M windows based on the assessments on compliance propensity at the randomly chosen times within each window, and maximize the compliance performance in the long term. Once we trigger m EMAs before the day ends, no more EMAs will be triggered. When we choose not to trigger EMAs in the first $M - m$ windows, we are forced to trigger EMAs in the remaining m windows on the day, regardless of what the compliance propensity is at these randomly chosen moments in each remaining window. In both cases, we will trigger exactly m EMAs on each day. In our current study, we set $M = 6$ and $m = 3$.

Our proposed RL strategies will be fit into this scenario in attempt to improve response compliance. Algorithm 6 reflects this design in our implementation. Specifically, when we are in a terminal state, which includes being in the last time window of the day or when m EMAs are triggered, all the coefficients associated with 'not trigger' actions will be updated using our designed reward signal for it.

6.6 Experiments

6.6.1 Data

To evaluate our proposed RL adaptive active sensing strategies, we conduct simulation experiments using mobile EMA data as described in Section 4.5.1. We focus on the random time EMAs that were delivered 6 times a day in randomly selected moments within each two hour window from 9 am to 9 pm everyday for momentary affect scores.

6.6.2 Baseline Methods

We use two baseline methods as comparisons to measure the performances of our proposed RL adaptive active sensing methods. The first baseline method is employing a random strategy that randomly selects 3 out of 6 2-hour windows each day. The second baseline method creates a supervised model with all cumulative data available up to the prior day, and apply this model for action decision at each active sensing decision point. At the end of each day, this model will be retrained with all available data, and deployed for the next day. We will apply XGBoost, which is a boosting algorithm that can gracefully handle missing data. The setting for the second baseline method will be the same as to the proposed RL methods. We will use the same context features learned from our context modeling methods, including both the momentary and routine features.

6.6.3 Experimental Settings and Research Questions

There are four parameters in the QLADE algorithm including the initial exploration rate ϵ_0, the step-size (or learning rate) α, the discount rate γ, and the eligibility trace-decay parameter λ. All four parameters fall within a range of 0 and 1. In addition, the reward signal has a discounting parameter β associated with the 'not trigger' action. Instead of tuning these parameters, for every participant in each method, we randomly choose values for them from the following values: 1) $\alpha = \{0.01, 0.05, 0.1\}$; 2) $\gamma = \{0.05, 0.1, 0.2\}$; 3) $\lambda = \{0.05, 0.1, 0.2, 0.5, 0.8\}$; 4) $\epsilon_0 = \{0.1, 0.2, 0.5\}$; and 5) $\beta = \{0.05, 0.1, 0.15, 0.2\}$. The exploration decaying rate is fixed to be $d = 0.8$.

With the above settings, we want to find answers to the following research questions: 1) How does the design of the state features impact the performance on various compliance metrics? 2) How is the performance of the proposed RL methods compared to the baseline methods? We will also analyze how the different state features impact the action decision in both individual and collective views using visualizations.

6.7 Results

Figure 6.3 shows the distributions of number of days in study, number of triggered EMAs, and the actual overall compliance rate for the data we use in our simulations. Note that almost half of the participants have received less than 30 EMAs in total, leading to an average of daily EMAs that is under 3. This low value will potentially limit the effectiveness of the proposed RL strategies. Thus we will analyze the performance of the strategies including the baseline methods by segmentation of the above three statistics.

Parameter Tuning. Figure 6.4 shows the average overall compliance aggregated across different parameter values in the six RL strategies. Consistently, we find the best combination of parameter values to be: $\lambda = 0.05, \alpha = 0.1, \beta = 0.2, \gamma = 0.1,$ and $\epsilon_0 = 0.1$.

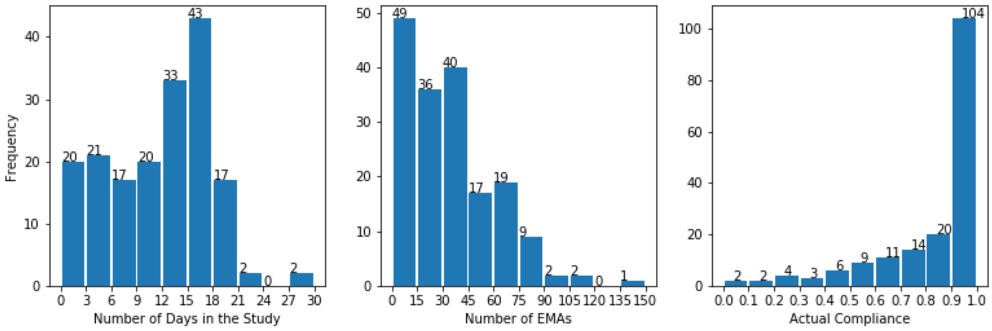

FIGURE 6.3: Study data and EMA statistics: (a) Distribution of number of days in the study for each participant. (b) Distribution of number of EMAs being delivered to each participant. (c) Distribution of actual overall compliance in the triggered EMAs of each participant.

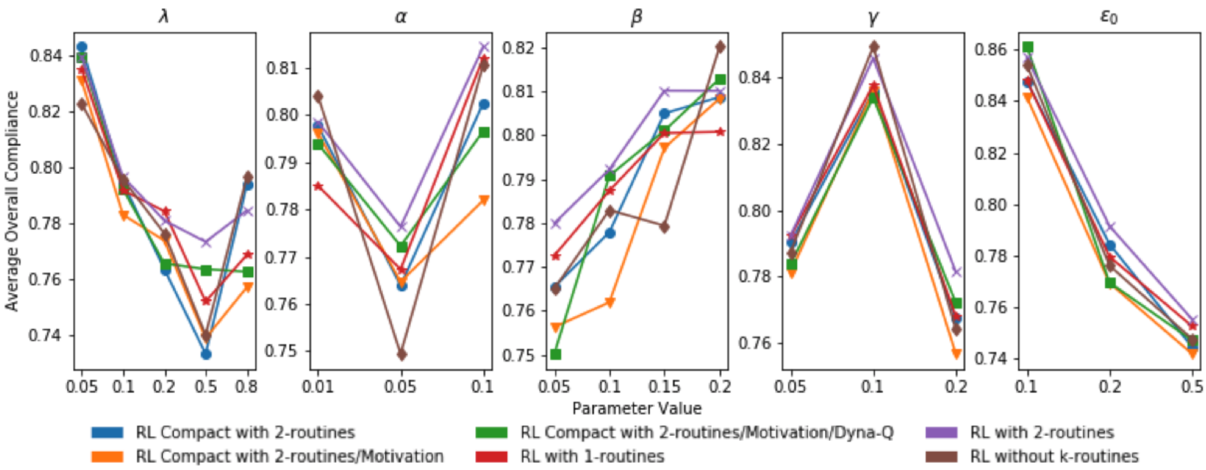

FIGURE 6.4: Parameter Tuning. Average overall compliance aggregated across different values in each RL parameter on the six RL strategies.

6.7.1 Performance Comparisons in Different Strategies

Comparisons within RL Strategies. Figure 6.5 shows the average overall compliance, daily compliance, and time-constraint daily compliance across the six RL strategies with different state features and the Dyna-Q method. Since all RL strategies use momentary state features, we will not mention it unless necessary. The RL strategy without any k-routine state feature has the same performance as the one with 1-routine state feature. But the strategy with 2-routines outperforms both of them. When using the compact representation, the performance has no improvements. Adding the motivation feature also does not lead to performance enhancements. Lastly, the Dyna-Q framework does not improve the overall performance either. Note that the order of performances among

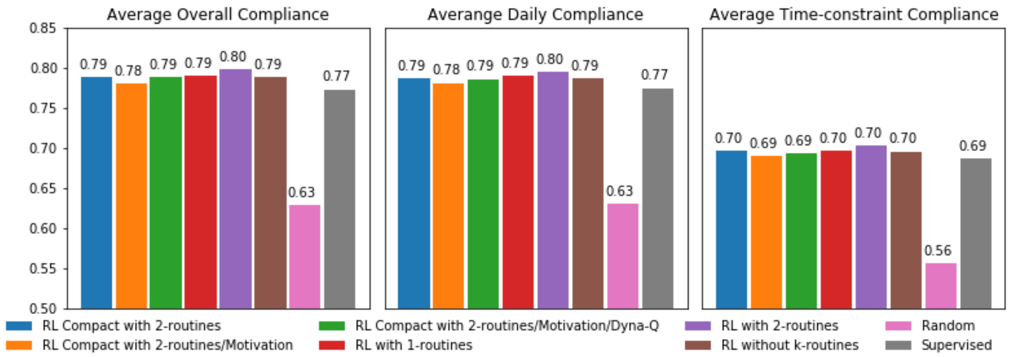

FIGURE 6.5: Performance by strategies. (a) Average overall compliance; (b) average daily compliance; (c) average time-constraint daily compliance across 6 RL strategies and 2 baseline strategies.

the different RL strategies in all three metrics are almost the same. The RL strategy with momentary and 2-routines state features slightly outperform all other strategies by a small margin.

Comparisons between RL strategies and Baseline Methods. From Figure 6.5, we can see that all RL strategies outperform the two baseline methods, including a random strategy, and a supervised strategy. In particular, the best RL strategy attains an average overall compliance 0.80, an average daily compliance 0.80, and an average time-constraint daily compliance 0.70, compared to 0.77, 0.77, and 0.69 in the corresponding metrics in the supervised method.

Comparisons by Segmentation in Number of Days in Study. Different participants remained in the study for different durations. We want to see whether these different lengths of time have impacts on the performance of the different strategies. Figure 6.6 shows the average performance segmented into three groups using the number of days in study. The cutoffs for the low, median, and high levels are 7 and 14 days. In the low level, the RL strategy with 2-routine state feature has the best performance in all three metrics. All RL strategies outperform the baseline methods. In the median level, the supervised strategy and the RL strategy with 2-routine state feature have comparable performance. The same conclusion is reached in the high level group.

Comparisons by Segmentation in number of Triggered EMAs in Study. Although the original schedule on EMAs are six per day in the study, most participants

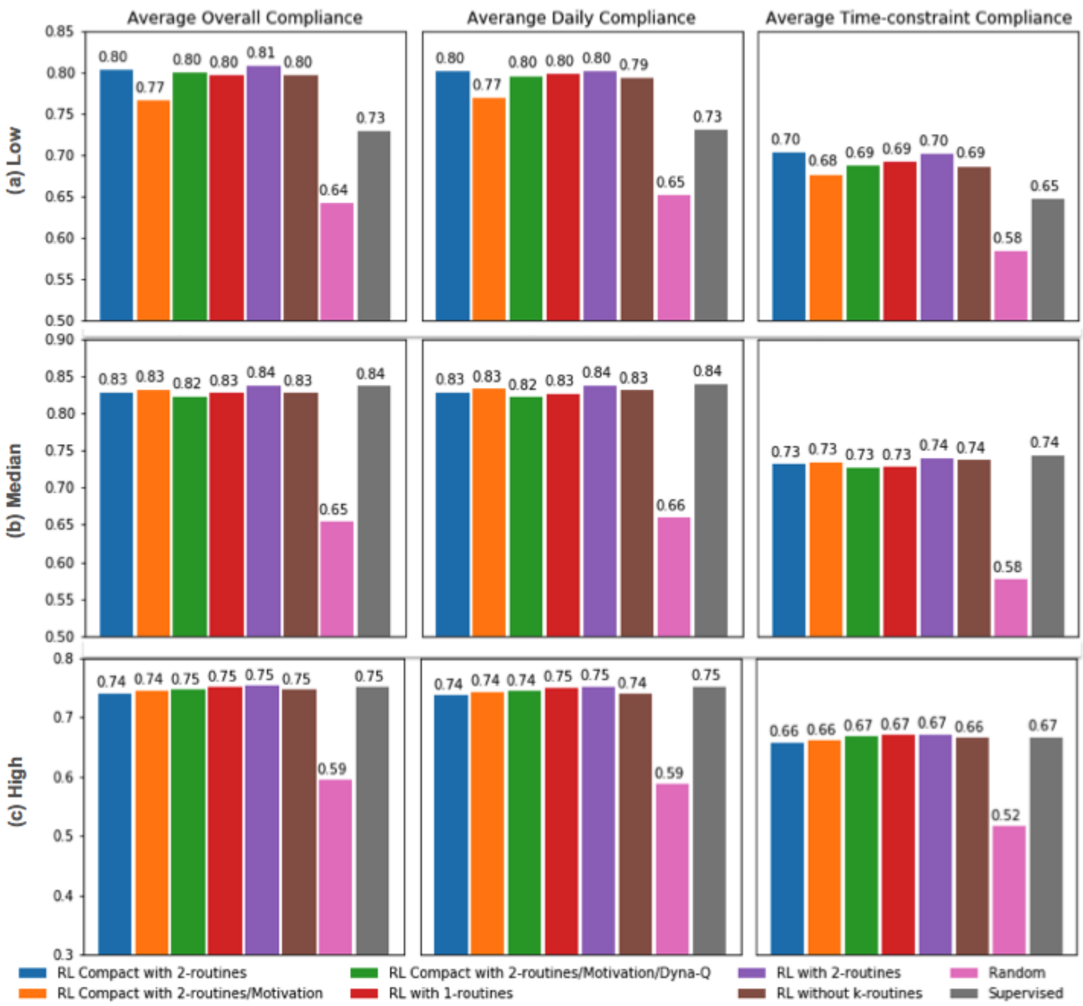

FIGURE 6.6: Performance by Strategies – segmented by number of days in study – low. The number of days in the study is (a) lower than 7 days; (b) between 7 and 14 days; (c) above 14 days.

did not receive six EMAs daily. This could be caused by app malfunction, drained battery, or switching off the Sensus app etc. For participants who had few data for learning, the performance in the various RL strategies may be compromised. Figure 6.7 shows the average performance segmented into three groups using the number of triggered EMAs during the study. The cutoffs for the low, median, and high levels are 30 and 60. In the low level, the RL strategies outperform the two baseline methods in all three metrics. In both the median and high level, comparable performances are observed between the RL strategies and the supervised baseline method. One possible explanation could be when the supervised method accumulated more training samples, its performance becomes better, leading to improvements and comparable results to the RL strategies.

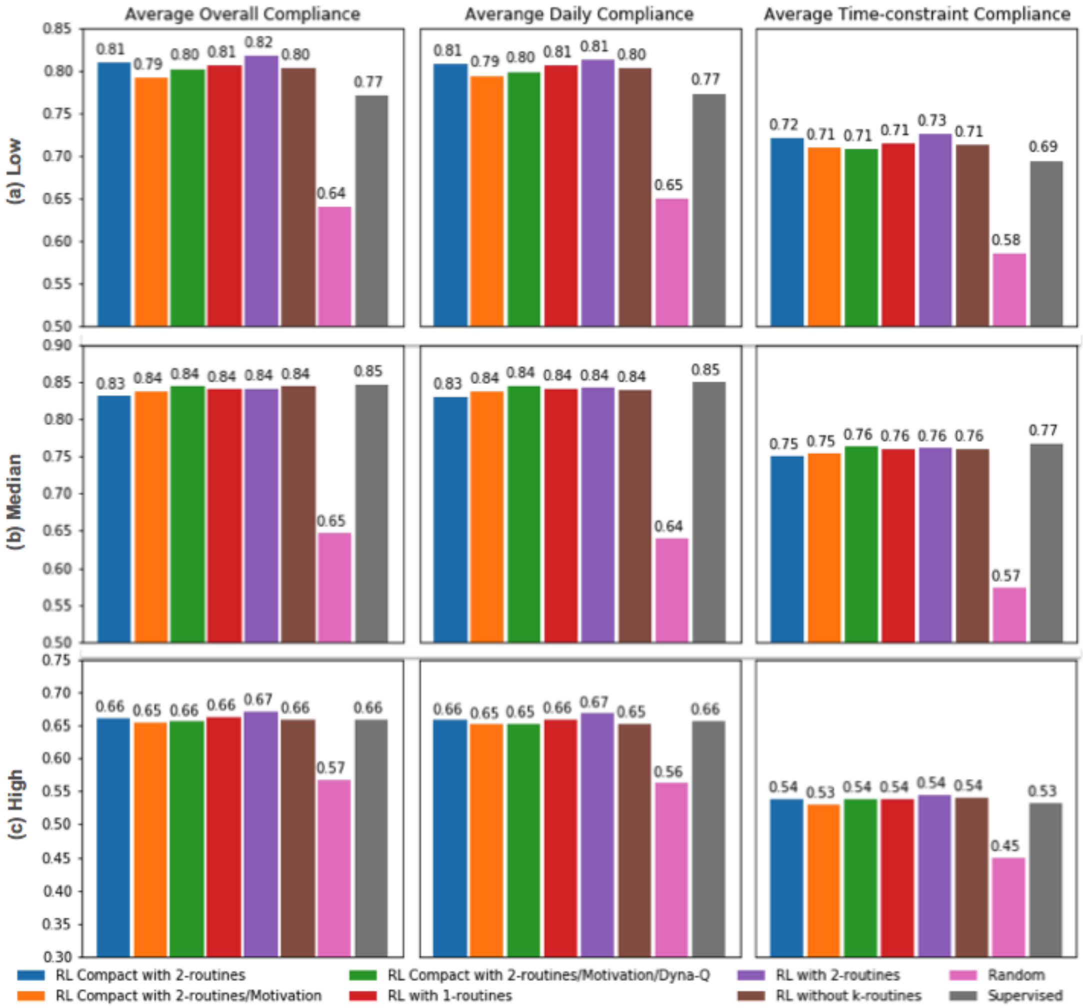

FIGURE 6.7: Performance by Strategies – segmented by number of EMAs in study – low. The number of triggered EMAs during the study is (a) lower than 30; (b) between 30 and 60; (c) above 60.

Comparisons by Segmentation in average number of daily EMAs in Study. Different from the total number of triggered EMAs during the study, the average number of daily EMAs takes into consideration the temporal impact, as our daily interruption budget is set to 3 in the simulations. When less than three EMAs were triggered in the actual data, the adaptive strategies may have no choice but to trigger all of them regardless of the value of the triggering moments measured by the action value function. Figure 6.8 shows the average performance segmented into three groups using the average number of daily triggered EMAs during the study. The cutoffs for the low, median, and high levels are 2 and 3. In the low level, we obtain similar conclusion, that is all RL strategies outperform the two baseline methods in all three performance metrics. One

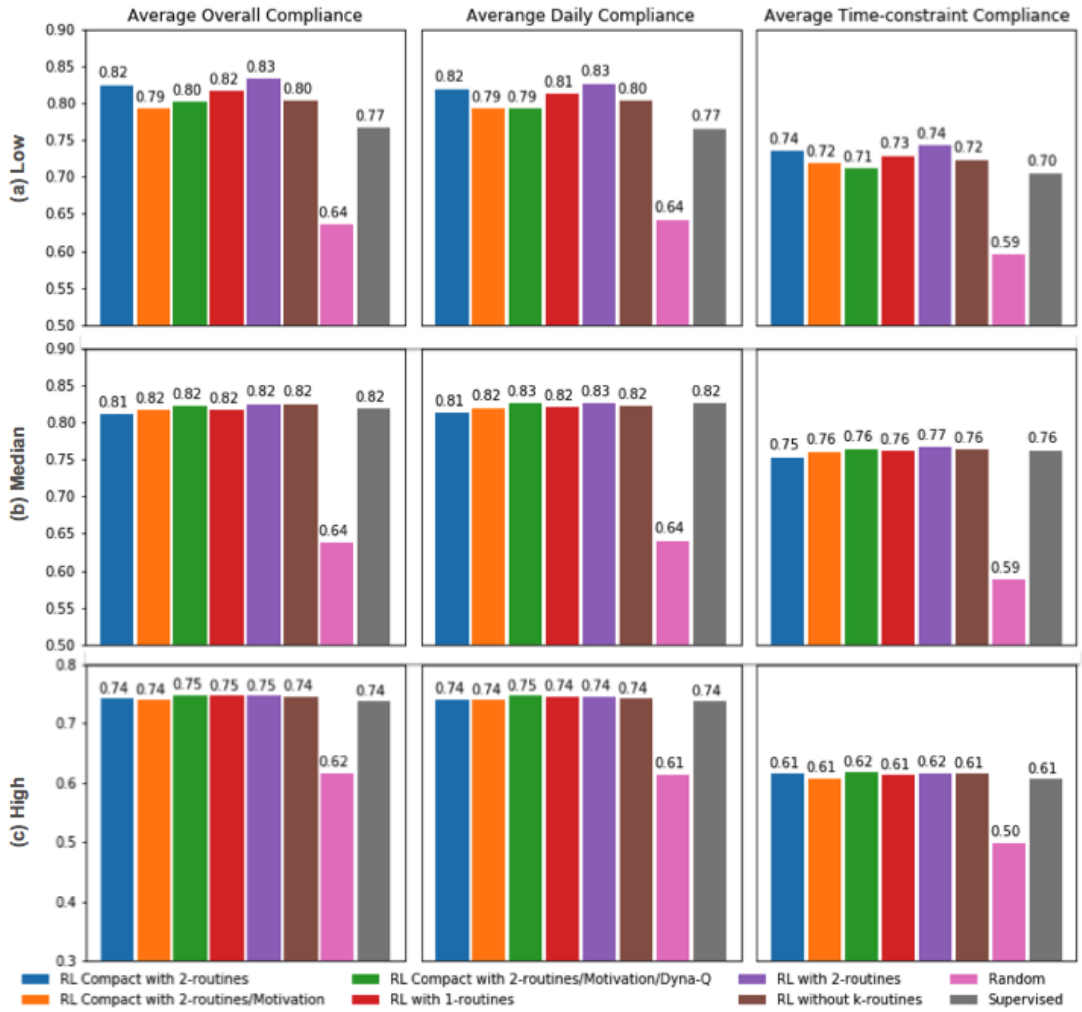

FIGURE 6.8: Performance by Strategies – segmented by average number of daily EMAs in study – low. The average number of daily triggered EMAs during the study is (a) lower than 2; (b) between 2 and 3; (c) above 3.

possible explanation could be that this group of participants is more sensitive to moments for active sensing given the fact that they receive less EMAs daily. In the median and high level, performances across different strategies, except the random method, are comparable.

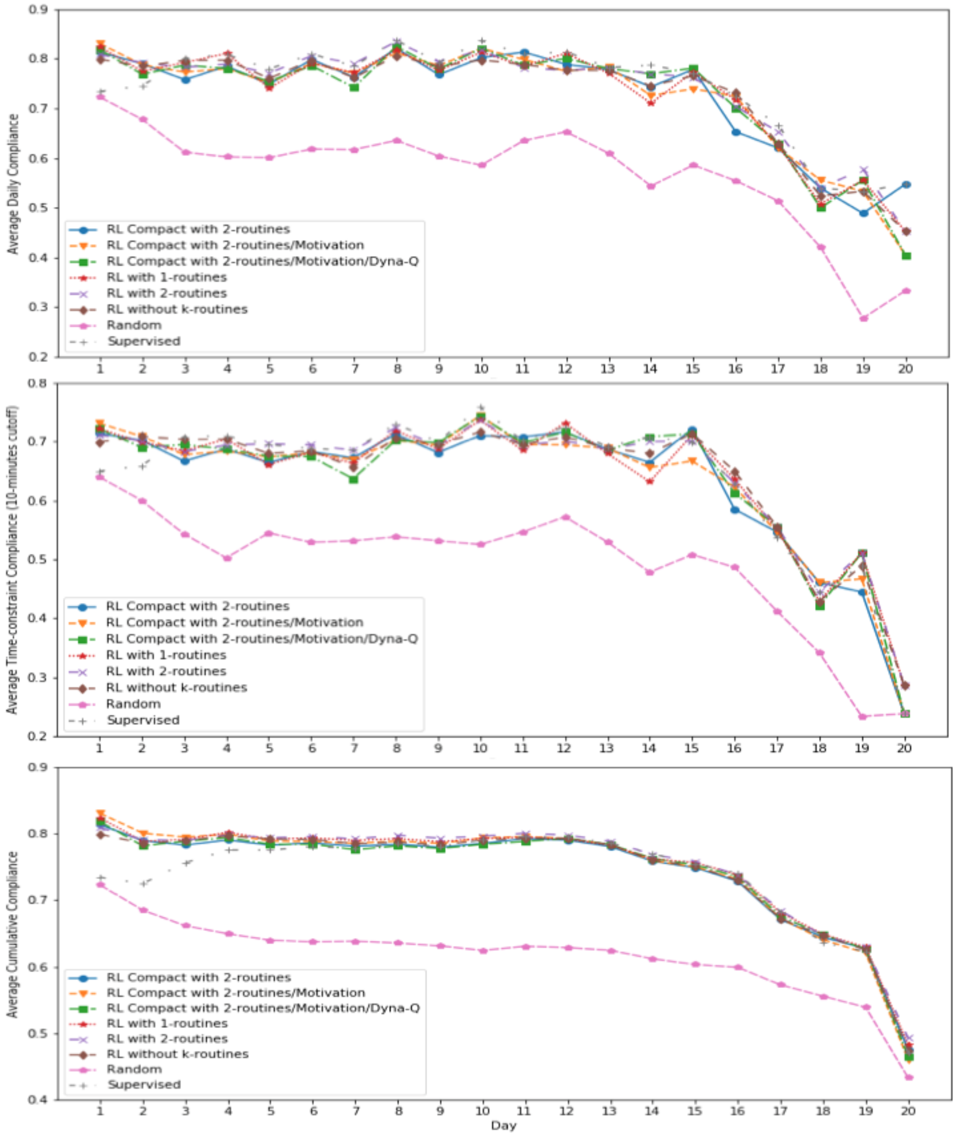

FIGURE 6.9: Performance of different adaptive active sensing strategies over time. (a) Average Daily Compliance; (b) average time-constraint daily compliance; (c) average cumulative compliance.

Performance Comparisons across Time. Figure 6.9 shows the average compliance over time across different strategies in the three metrics. The random strategy has the worst performance in all three metrics over time. We observe that the supervised method performs poorly at the first few days. This may be due to the cold start problem, in which the supervised agent does not have any data to train a classifier for action decision. Overtime, regardless of what strategies are being applied, the compliance in all three metrics is dropping. This attrition over time has been a significant challenge in almost all mobile sensing studies and applications. Overall, different strategies except the random strategy perform visually closed, as can be seen from the figures.

6.7.2 Personalized Active Sensing Policies

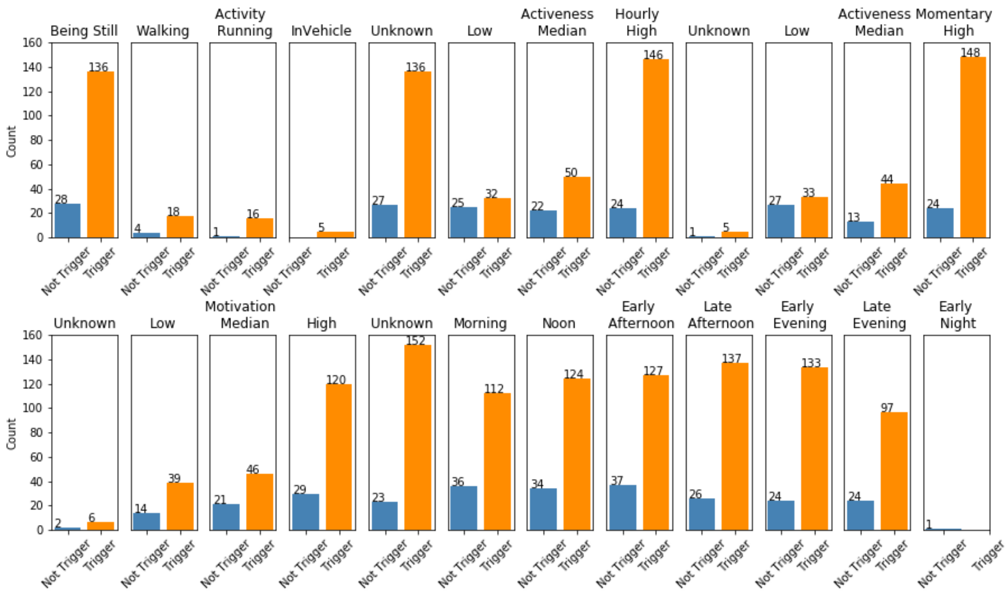

FIGURE 6.10: Counts of individual policies favoring either the 'trigger' or 'not trigger' action in each state feature value.

Figure 6.10 shows the counts of individual policies that favor either the 'trigger' or 'not trigger' actions in each state feature. The total number of policies learned with those that have any EMA data are 174. However, due to the scarcity of learning samples for each individual policy, some policies do not have the corresponding state values leading to much lower counts in some state feature values.

Figure 6.11 visualizes the propensity of each state feature value towards the two active sensing actions. Each row of the heatmap represents a participant's policy, and each column represents a state feature value. Steel blue represents favoring the 'trigger' action while dark red represents favoring the 'not trigger' action. White cells are caused by lack of learning samples containing the corresponding state feature value in the simulations.

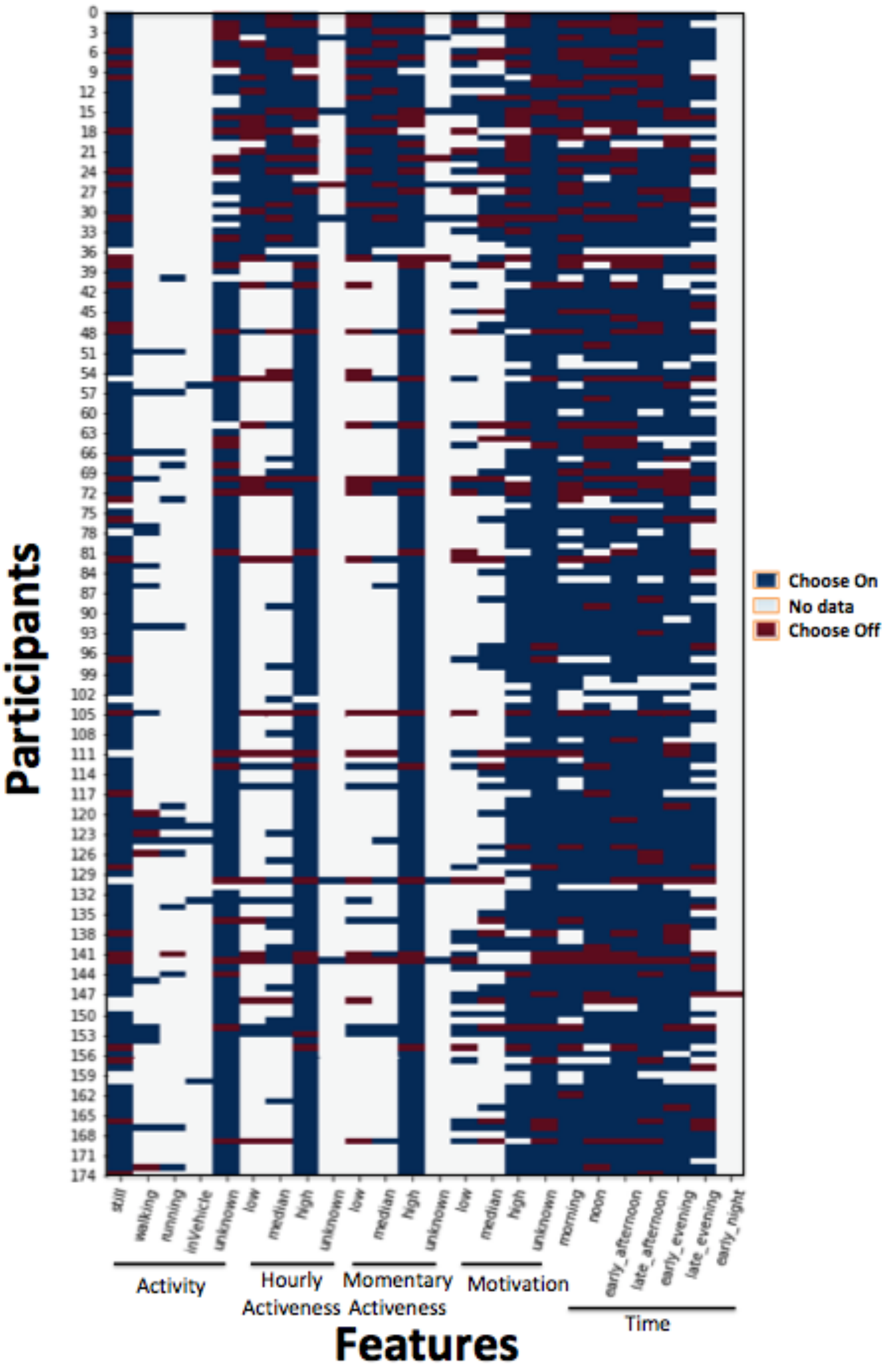

FIGURE 6.11: Individual Policy Profiles.

6.8 Discussion

The goal of adaptive active sensing is to improve users' active sensing compliance by leveraging their contextual states to better manage task deployments. RL is a natural fit to formulate active sensing as a personalized and adaptive sequential control problem. Designing a RL strategy requires effectively identifying critical state features that are relevant to compliance, and designing a reward signal that can shape the policy towards higher cumulative long term compliance. More research needs to be conducted in these topics in order to obtain more effective adaptive active sensing strategies under the RL framework.

In addition to employing adaptive active sensing, other solutions are desired to address the compliance challenge due to its complexity. For example, lower compliance over time is inevitable due to the law of attrition [174]. To maintain users' motivation, gamification strategies can help engage users to complete active sensing tasks by embedding them into games that are intrinsically more interesting and enjoyable to them. Works such as [177] and [218] design games using various psychological and behavioral theories to achieve this goal.

It is equally important to adopt good design practice in creating active sensing tasks. For example, the quantity, order, format, scale, clarity of survey items, study window, trigger mechanism and frequency, incentive, robustness of hardware and software platforms are all important considerations that may impact mobile EMA compliance. Minimal design in app interface and navigation can reduce the users' efforts and time. Reminders can also help mitigate forgetfulness in completing active sensing tasks.

In this work, we only focus on using adaptive active sensing to improve users' task compliance. However, there is another equally important aspect in active sensing, that is data quality. Adaptive active sensing algorithm should also consider data quality as a goal and factor it into the design of the RL strategies. Although this problem is out of the scope in this dissertation.

6.8.1 Limitations and Future Work

The data used for our simulations are relatively short in duration for modeling user behaviors. This may also compromise the performance of our RL strategies due to the limited learning samples. For example, some study participants were only active for less than a week, and within such short time frame, uncertainties in user states associated with the chosen actions are high, leading to suboptimal performance. To overcome this problem, we have conducted segmentation analysis based on total number of EMAs and number of days in study to compare the performance of the proposed strategies. We also apply the Dyna-Q framework to enhance sample efficiency by integrating planning with learning. However, we are not certain why this approach does not lead to improvement. In our future work, we will conduct more thorough investigations in different strategies that can enhance sample efficiency to improve compliance performance.

Another limitation in our current work lies in the technologies we used to collect the data. The Sensus mobile app we applied to collect the study data was not as mature and robust as it has become. Back then, it was less energy efficient and more buggy, which resulted in lower retention rate and average number of EMAs being collected daily. When less EMAs are available daily, our proposed RL strategies may be handicapped due to the way we formulate active sensing to be minimally biased and context selective. In future works, we will investigate these proposed RL strategies with higher quality data.

We have proposed a user model that combines both momentary and routine state features to represent the user environment. However, this is the first work that proposes to design adaptive active sensing using the RL framework. We still have no prior knowledge about what state features are more effective. Existing works in interruption and notification management have studied a rich set of human contexts that can predict interruptibility and receptivity in notifications. Our current passive data do not allow us to extract most of those features, especially those that require self-reporting. In our future work, we will include and test those state features in a real world implementation.

Lastly, the high-level routine feature we propose in this work has a significant computation challenge in higher orders and over longer period of time. Higher order k-routines are more expressive in people's life, especially those that occur frequently. When k is large, all possible combinations of life-blocks within and across days can blow up exponentially. In order to address this challenge, we need to maintain only the most frequent set of them after learning in a certain initial period of time, allowing new routines to enter the elite set, and outdated ones to be removed from it.

6.9 Conclusion

In this chapter, we propose several strategies for adaptive active sensing using the RL framework. We show that the RL strategy with momentary and 2-routine state features consistently outperform all other RL strategies and two baseline methods including a random strategy and a supervised strategy. We apply the Dyna-Q framework in attempt to enhance sample efficiency given limited learning samples in active sensing, when compared to most other RL applications. However, using a simple bootstrapping sample model, the Dyna-Q framework does not lead to improvement in performance. We overcome the delayed reward challenge by designing a reward signal that backward estimate the direction of updates in the policy. This is achieved by storing all necessary parts for updating the 'not trigger' action value function temporally for each episode. We estimate motivation using moving window cumulative compliance and incorporate it as a state feature to investigate possible influence of motivation level on the performance of the active sensing strategies. This intuition originates from the observation of individual compliance that is closed to perfect or near zero, meaning that no matter what contexts a person situates, their compliance may not be affected at all. However, the addition of the motivation feature does not lead to performance improvement. We will evaluate our proposed method using newer and better quality data while we work towards its real world implementation, which will be further explained in chapter 7.

Chapter 7

Sensus: Implementing an Adaptive Sensing Platform

Sensus is a multi-OS mobile sensing platform that provides supports for general purpose mobile sensing applications and data collection. [236] It has been continuously developed and maintained by our own developers at University of Virginia. We provide an introduction to the sensing capabilities in Sensus, and our plans to integrate adaptive sensing.

7.1 Supported Sensors

Sensing is the foundation of mobile sensing platforms. Sensus provides supports to collect data in a wide variety of smartphone embedded sensors. Table 7.1 shows a list of these sensors supported by Sensus. They are grouped based on the type of information they provide in motion, position, ambience, physiology, and phone usage. In addition, Sensus also supports integration of wearable devices such as Fitbit, Empatica, and Huawei

TABLE 7.1: Sensing probes available in Sensus.

	Motion	Position	Ambience	Physiology	Phone usage
Sensors	Accelerometer	GPS/visits	Altitude	Heart rate	Battery
	Activity	Wireless	Bluetooth	Pedometer	Power connection
	Speed	Bluetooth	Light/Proximity	Sleep	Calendar events
	Gyroscope	Estimote beacon	Microphone	HealthKit	Image metadata
	Compass				

FIGURE 7.1: Interfaces for sensing within Sensus. (a) Protocol Menu, (b) Probe Menu, (c) Linear Acceleartion Listening Probe, (d) Battery Level Polling Probe.

smartwatch through Bluetooth technologies. In the living link lab project, Sensus is coupled with Estimote beacon to collect indoors position information, which can be used to understand student and faculty's daily activities within the lab. All the above sensing capabilities are passive, while Sensus also supports collection of active data such as mobile surveys. Both mobile EMAs and passive sensing data have been collected using Sensus in multiple studies within our group [32, 6, 37, 36, 48, 28].

There are two sensing modes for different sensors, listening and polling. In listening mode, a sensor monitors any changes in signals periodically, and upon detection of changes, activates itself to collect data. In contrast, a sensor in polling mode senses at designated rates and rests at chosen intervals between successive polling operations. Figure 7.1 shows a few interfaces within Sensus for the supported sensors and their configurations.

7.2 Energy Management and Adaptive Passive Sensing

Sensus has several different ways to manage energy consumption when it is being deployed. First, sensing rate and interval can be configured within each sensor to save energy. Second, it can be configured to allow data transmission only when the phone is charged, WiFi is connected, and/or battery level is beyond a certain level. Third, a few simple sensing strategies are available. For example, accelerometer can be configured to collect data only when acceleration level is above certain designated threshold. Likewise, GPS will be activated only when speed is beyond a chosen threshold. However, there is no sophisticated sensing policies like the ones we developed in this dissertation.

To integrate adaptive passive sensing using RL, Sensus has implemented an architecture that allows adaptive sensing agents to be plugged into the mobile app. There are two types of adaptive sensing agents: 1) externally designed software sensing agent; 2) adaptive sensing policy language (ASPL) defined sensing agent. RL adaptive sensing agent can be created through the first approach. For Android, the RL agent is created as a class library and built into a dynamic link library (DLL), which can then be plugged into the Sensus mobile app through scanning a QR code. For iOS, since it does not allow loading codes at run time, RL adaptive sensing agent has to be hard coded into the Sensus mobile app, and the app has to be redeployed through the Apple app store before the sensing agent becomes available. For more details about these topics on adaptive passive sensing, please visit the Sensus development documentation. With this architecture design, it is convenient to create external adaptive sensing plug-in, or write codes directly inside the Sensus code base, and integrate them into the Sensus mobile app.

7.3 Adaptive Mobile EMA

Mobile EMA in Sensus is called 'scripted interaction'. Users can create as many surveys as they want in each sensing protocol by adding different scripts. Each script refers to one survey, and can be scheduled to trigger at a fixed time (e.g., fixed time mobile EMA) or a random time within a chosen time window (e.g., random time mobile EMA). Within each survey, different input elements are available to construct input groups, each of which corresponds to different survey questions.

To enable adaptive mobile EMA, Sensus is designed with a similar architecture to that of adaptive passive sensing through external software plug-in for Android or hard coding into the Sensus mobile app for iOS. More information on these developments are available on the development documentation as well.

7.4 Conclusion

Adaptive sensing, including both adaptive passive and active sensing, has important implications in real mobile sensing applications. It leverages contextual understanding feeding off by passive sensing probes to guide sensing deployments, both hardware sensors and surveys. The above architecture lay down the foundation for carrying out implementations on our proposed RL sensing strategies to evaluate their efficacy. Our future plan is to translate the proposed sensing strategies in this dissertation into both passive and active sensing agents that aim to improve long term sensing efficiency and user compliance.

Chapter 8

Plans and Future Challenges in Implementations

In this chapter, we discuss plans and various challenges in the design and implementation of mobile sensing and intervention system using Sensus as the foundation.

8.1 Plans: Integrating Monitoring, Modeling, and Intervening

Monitoring. The purpose of monitoring is twofold. First, data quality and compliance is of upmost importance to guarantee the utility of the collected data in most mobile sensing studies. Second, monitoring users in real time is important in some applications in order to assess users without substantial delay, and follow up with prompt actions and interventions. Sensus is coupled with an external web dashboard that enables monitoring of data collection in different studies and applications. This dashboard provides supports for study management and data monitoring. Figure 8.1 shows a few screenshots of its current design for the DAPAR WASH project.

Currently, Sensus lacks the capability to preprocess data, and use them to model users as they are being collected. This also prevents us from monitoring study participants more closely in future studies such as those concern participants' mental health. Being able to monitor user state closely will also empower researchers to better design behavior

FIGURE 8.1: Sensus Web Dashboard.

models and interventions, leading to more efficient study design and deployment, as well as potentially improved study outcomes. In the foreseeable future, we plan to add these capabilities to Sensus.

Modeling. Data collected by Sensus in various studies have been applied to study various human outcomes such as social anxiety [32], state and daily affect [6, 37], mental health in cancer patients [36], and compliances in mobile EMAs [25] using predictive models. However, These models are built offline and in retrospect. Currently, Sensus does not have any built-in modeling functionality to create model on the fly by incrementally consuming newly available data. As Sensus is being enhanced, these new capabilities will also be incorporated into it in the future.

Intervening. Transforming Sensus into an intervention platform is our ultimate goal. Mobile intervention is a study area that encompasses sensing, monitoring, and modeling,

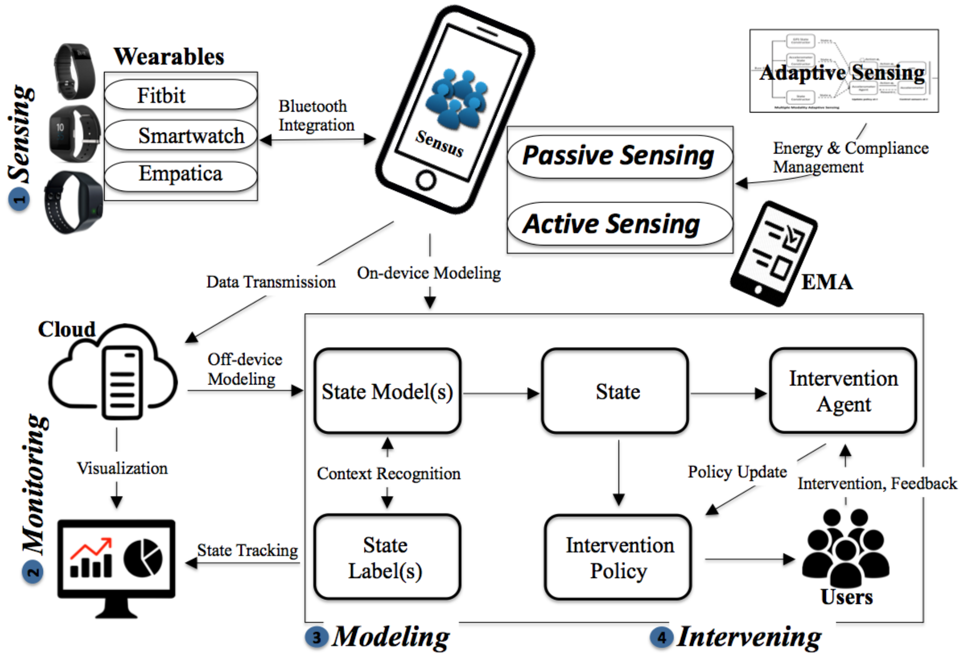

FIGURE 8.2: Sensus Design Architecture.

while based on analytical insights, proactively interact with users to deliver effective interventions. Specifically, there is a common foundation between adaptive intervention and adaptive sensing as both are built upon context-awareness using passive sensing data. Indeed, the RL framework has been proposed for implementing a type of mobile intervention called Just-in-time Adaptive Intervention (JITAI) [154]. After implementations of RL agents for adaptive sensing becomes mature, it is natural for us to transition from these foundations to design and enable mobile interventions within Sensus.

Figure 8.2 visualizes the different, both existing and future, capabilities within Senus. According to this architecture design, Sensus will provide options for on-device and cloud monitoring and modeling capabilities. Raw sensing data can be preprocessed locally, or uploaded to and preprocessed in the cloud, depending on the application configurations. Modeling pipelines can be embedded into the app or setup on the cloud to enable on-the-fly modeling with small latency. User states learned from context models will be visualized timely in dashboards. In addition, adaptive interventions will be implemented using RL framework, while the context models built from the modeling component will provide state information to the intervention agent. In the next few sections, we discuss some of the design and implementation challenges.

8.2 Designing Real Time Mobile Sensing System

Real time Mobile Sensing system empowers real time understanding and monitoring, opening up the possibility of real time intervention and decision making in many real world applications such as patient monitoring and service recommendation. [202] The vast majority of existing studies collected mobile sensing data with ground truth labels to build models for context recognition and behavior inference offline and in retrospect. However, building models in real time will require several substantial adjustments when compared to these offline models. Below we discuss several of these potential adjustments, namely extracting features in real time, multi-modal feature fusion, knowledge sharing among different users.

8.2.1 Extracting Features in Real Time

Extracting features for real time behavior inference has the following differences when compared to offline behavior inference. First, real time feature extraction encounters the 'cold start' problem, meaning that no data are available as the system starts up the behavior modeling process. As more and more data become available, many of the feature extraction algorithms (e.g., clustering algorithms that learns semantic place labels) needs to be adapted into their online learning version. In the case of obtaining semantic labels, it is no longer feasible to look at data from the entire study, and determine what semantic labels to use. Instead, because new and unexpected types of places can be visited over time, dynamically adjusting the semantic labels (i.e., adding new label classes and removing old label classes that are no longer visited) is preferred. Such dynamic process will also help maintain the system to be up to date and scalable in size.

Secondly, depending on outcome metrics, in order to enable real time inference, the algorithms also need to be designed to process the data incrementally so that the latency is minimized. For example, if we store the entire day of accelerometer data to compute the proportion of time a smartphone user is being active (i.e., moving around with his/her phone to approximate activeness), it will take up a large amount of storage space and much longer time to extract this feature. Instead, the proportion of time will be updated as batches of accelerometer data are received and processed, enabling the users to access and use it at a faster pace.

Third, continuous monitoring requires the same feature values being updated in a rolling window fashion. To achieve this and reduce resource consumption (e.g., memory and energy), the feature extraction algorithm also needs to be designed in a rolling fashion, reducing duplicated computations by reusing the parts of the results that remain the same at each time step. Last but not least, all the data preprocessing steps need to be performed on the fly as new data come in and missing data are detected. All these adjustments require careful engineering to enable an efficient real time behavior inference system.

8.2.2 Fusing Multimodal Features

Having access to data from multiple sensing modalities has several benefits, including more robust predictions (e.g., when models from individual modality all infer the same behavior), higher predictive power (e.g., different sensing modalities capture complementary information), and better resilience in prediction (e.g., data in one modality are missing). [12] However, existing mobile sensing works did not explicitly consider how to fuse data from multiple sensing modalities. Usually, features are extracted from each sensor and combined into a feature vector for context recognition and behavior inference. In multi-modal machine learning, this approach is called early fusion, in which all the features are applied in any uni-modal classifiers and regressors.

In [12], many other multi-modal fusion methods were reviewed based on their dependencies on the learning algorithms (e.g., model-agnostic vs. model-based methods). In model agnostic methods, in addition to early fusion method, we have late fusion method (i.e., performs integration after each of the modalities has made a decision) and hybrid methods that combine both early and late fusion methods. In model-based methods, many learning algorithms are designed for representing and inferring the target outcome through multi-modal sensing data. A few examples include kernel-based methods, graphical models, and neural networks. [12]

In [180], the authors designed a multi-modal deep learning model that independently learn hidden layers for each sensing modality before fusing those hidden layers as the input layer to the final neural network for activity recognition. When designing real time system, engineers need to consider how to most effectively represent and fuse data from heterogeneous sensing modalities, while address challenges arising from alignment and noise in them.

8.2.3 Sharing Knowledge Among Users

Sharing data across all users or within groups of similar users may lead to improvement in inference accuracy and learning rate. Up to this point, we assume adopting only individual data to create personal models for context recognition and behavior inference. However, it is also possible to speed up and improve learning by leveraging collective data from all users or similar users.

There are several advantages of sharing data across all users or within groups of similar users. First, it can mitigate the 'cold start' problem when no data from a new user is available for modeling; secondly, it can facilitate learning by sharing available models or parameter configurations as a new starting point for new users; lastly, it has the potential to greatly enhance inference performance by sharing global knowledge through various model designs (e.g., fusing personal features extracted through individual data and global features obtained from collective data).

In offline context recognition and behavior inference, it is much easier to take advantage of all existing data to achieve this knowledge sharing goal. For example, we can define certain similarity measures to group similar users together, and build group models instead of individual models. [6] This approach is especially useful for individuals with very limited data to create their own personal models. Unfortunately, when implementing real time systems, each individual does not have direct access to others' data, which requires careful system design to efficiently share the data without expending significant resources (e.g., costs incurred by data transmission). Another challenge lies in being real time. Transmitting and processing large amount of data for feature extraction and modeling can lead to significant latency. One solution is to update global components of the inference model daily and offline, and keep the same values for a day.

How to design mobile sensing systems that enable efficient data and knowledge sharing is still a new topic. Jiang et al. proposed the PLOS framework, a distributed mobile sensing learning algorithm that jointly model the commonness as well as the differences shared among the users. [95] In this work, the raw data of the users are processed locally, and only model parameters are sent to server for sharing among different users.

8.3 Personalized Feature Learning

The idea of Personalized Feature Learning (PFL) is in analogy to that of precision medicine. Its goal is to design individualized predictive features to achieve better inference performance. Precision medicine recommends unique treatment regime for each patient or group of similar patients. Likewise, PFL proposes to learn individualized features due to individual differences in behaviors, habits, personalities, cultures, and values. Universal features may not present the same predictive power in modeling different individuals, and could lead to suboptimal inference performance. One solution is to adopt personalized models (e.g., individual neural network models) that could uncover unique sets of features for each user. However, this requires sufficient data for each user. Another solution is to apply unsupervised techniques such as association rule mining to learn patterns or rules that are unique to each individual for predictive modeling. [190, 72, 191]

To date, most existing works apply supervised learning techniques for context recognition and behavior inference. This approach requires a universally defined feature set as input. PFL remains a future research topic in mobile sensing inferences.

8.4 Measuring Changes

In many applications, we need to detect changes instead of purely monitoring a given state over time. One such example is supporting smoking cessation [156], in which the authors define 'geofence' as a circular area surrounding self-reported vulnerable relapse locations, and use them to trigger support messages to the smokers when they enter the 'geofence'. In this example, the change of location is being continuously monitored and compared to the 'geofences'. Another example is detection of behavior changes for psychotic relapse [18], in which the authors highlighted the temporal changes of certain behaviors in five patients before hospitalization. In particular, they identified changes of different behavior markers (e.g., self-reported symptom changes, shifts in location patterns, increase in device use between 12pm and 6am, increase in speech frequency and duration, and declines in physical activity) before hospitalization using the same markers in other time windows as baseline. If these early warning signs can be captured, and interventions can be provided timely based on this information, then hospitalization may be prevented.

Unfortunately, formal methodologies are lacking in measuring changes. There are a few works attempting to measure or visualize changes. Tseng et al. applied a technique called Robust PCA on an hourly by days encoding matrix, and decomposed it into a routine pattern matrix for visualization of sutdy participants' behavior changes overtime. [212] Doryab et al. extracted features from noise level, movement (acceleration), light intensity, phone usage (e.g., Number of tasks and processes, frequency of change in tasks and processes, time between changes, and frequency and duration of screen on and off), location (e.g., time at/away from home, number of places visited, travel distance), and social communications (e.g., number of incoming and outgoing calls and text messages, number of contacts, duration of incoming and outgoing calls); calculated average values in these features for each participant as baselines; subtracted the baseline values from each feature and treated them as change measurements to study their correlations with depression. [55] In [223], the authors extracted low-level features (e.g., physical activity, sleep duration, sociability based on face-to-face conversational data) and high-level features (e.g., class attendance, studying, study duration, study focus, and social behaviors, such as, partying and partying duration), and combine them as predictors to build regression models using different time as breakpoints on various behaviors over the course of a semester. They considered the differences in slopes of the fitted linear models as measurement of behavior change to understand how they differ throughout the course of a semester. Harari et al. applied a piecewise linear regression model and a structural equation model (SEM) called Latent Growth Curve Model (LGC) on time series of weekly mean activity and sociability durations to measure the changes over time. [80]

Measuring changes in critical determinants over time and using them as trigger points for intervention is intuitive in mobile intervention. However, in context recognition and behavior inference, we usually extract features from segments of sensing stream, but ignore change-based features, which may be critical in inference performance. Thus, it is important to formalize change metrics in mobile sensing. We propose a three-step framework in measuring changes: 1) determine the reference point of changes; 2) define the metric(s) of changes; and 3) design the algorithm(s) to obtain the metric(s). In [55], Doryah et al. used the mean feature values as the reference for changes, and the difference between the mean and each sample feature value as the metric of change, subtraction between the two as the algorithm. In [223], mid-term and individualized breakpoints, slopes, and linear regression are the chosen reference point, metric, and algorithm for measuring changes, respectively.

8.5 Conclusion

Inducing positive change or supporting better decision making is the ultimate goal of mobile sensing inference. Inference serves as a step stone to understand users' contexts, monitor changes over time, and detect timely needs for triggering intervention. It is a critical component in the Just-in-time adaptive intervention framework, which aims to provide timely, personalized, localized, and on-demand interventions to users. [155] There remains many challenges between inference and intervention. These include efficiently collecting sensing data, accurately modeling behaviors, timely identifying the right moment for intervention, and correctly choosing the right intervention regime for delivery. The works in this dissertation position and prepare us to develop a mobile sensing platform that can bridge the gap between sensing and intervening, by incorporating the various components delineated in this dissertation using RL, which is considered a promising framework for realizing general purpose artificial intelligence.

Chapter 9

Conclusion

In this chapter, we conclude this dissertation with a summary of our work in the previous chapters, as well as ideas for future research.

9.1 Our Contributions

In this dissertation, we propose to unify both passive and active mobile sensing under the reinforcement learning framework for adaptive sensing. We summarize our contributions as below:

1. In Chapter 3, we propose a feature extraction framework to guide feature extraction in mobile sensing data. This FE framework is based on substantial reviews of existing works in mobile sensing applications. It helps mobile sensing researchers better understand what steps to take to preprocess raw mobile sensing data and extract features for modeling user contexts and behaviors.

2. In Chapter 4, we systematically categorize energy saving strategies in existing mobile sensing applications according to the proposed mobile sensing application pipeline (see Figure 4.1), which provides a clear view on the sources of energy challenge in continuous mobile sensing. To address this challenge, we formulate adaptive passive sensing as a sequential control problem, and propose a reinforcement learning algorithm called QLADE to optimally control low-level sensing in smartphone embedded sensors. Our simulations using real continuous mobile sensing data from

220 participants for more than 2 weeks show consistently better performances for the proposed QLADE algorithm when compared to the random and learning automata baselines for both accelerometer and GPS. We also show that our proposed adaptive sensing strategy does not compromise predictive performance in various machine learning models on social anxiety and daily negative affect when compared to continuous sensing and duty cycle methods.

3. In Chapter 5, we extend the proposed method in Chapter 4 by reformulating adaptive passive sensing with an expanded action space, new sets of state features, and a more generalizable reward signal. These new RL strategies are shown to consistently and significantly outperform four baseline methods on energy efficiency in simulations using the same dataset, while achieve similar predictive performance in modeling social anxiety and daily negative affect.

4. In Chapter 6, we switch focus to active sensing and propose adaptive active sensing strategies that aim to achieve higher long term cumulative user compliance. To effectively model user state, we combine low-level momentary context with high-level routine context using a proposed concept called k-routines. Using real random time Mobile EMA data in simulations, we show that our proposed RL strategies consistently outperform the baseline methods including a random strategy and a supervised strategy in mobile EMA compliance. We also investigate several different techniques including a more compact representation in routine state, incorporation of motivation as a state feature, and the Dyna-Q framework for better sample efficiency, to improve the compliance performance in mobile EMAs. Although the results are not encouraging, they provide ideas to explore new strategies for further compliance improvement using adaptive active sensing.

9.2 Future Directions

Many interesting topics in adaptive mobile sensing remain to be pursued post-graduation. We list them below:

1. In adaptive passive sensing, state representation learning using end-to-end deep learning approaches as function approximator in our proposed RL strategies has the potential to more effectively represent user's underlying activities, and achieve better energy efficiency. In addition, these end-to-end approaches can avoid manually extracting state features. We plan to investigate these deep learning approaches and their trade-off in energy overheads. Equally important is reward design. We plan to evaluate different options of R'_t in Equation 5.6, and compare their learning efficiency and performances in different adaptive sensing strategies.

2. In adaptive active sensing, successfully identifying critical state features that are key determinants of response compliance requires better user behavior models. In this work, we apply personalized features such as users' current location and high level routine context as state features, and show improvements in compliance performance. We plan to further explore other options on user modeling. Data quality in active responses is another important aspect of active sensing. We plan to develop adaptive strategies to achieve better data quality (e.g., response accuracy) in active sensing. We can design RL strategies that are multiple objective to simultaneously achieve these different goals. We also want to evaluate other avenues that can maintain users' motivation and thereby active sensing compliance. These include using gamifications and various incentive strategies. Lastly, due to limited learning samples, more sample efficient algorithms need to be investigated with data that are collected over longer period of time.

3. In this dissertation, our proposed methods are examined in isolation to a real mobile sensing system, in which the collected sensing data may be processed and modeled as they are made available. We need to carefully design a data pipeline that can

reduce redundant operations on processing the collected raw sensor data while integrate our proposed adaptive sensing methods seamlessly. This needs to take into consideration what state features are being extracted in our adaptive sensing strategies and whether they can be reused for other modeling purposes. We also plan to add all the desired capabilities according to Chapter 7 and 8 to further enhance the Sensus mobile sensing platform towards its ultimate goal of becoming an intelligent intervention system.

Bibliography

[1] Saeed Abdullah et al. “Cognitive rhythms: unobtrusive and continuous sensing of alertness using a mobile phone”. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016, pp. 178–189.

[2] Saeed Abdullah et al. “Towards circadian computing: early to bed and early to rise makes some of us unhealthy and sleep deprived”. In: *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. ACM. 2014, pp. 673–684.

[3] Phil Adams et al. “Towards personal stress informatics: Comparing minimally invasive techniques for measuring daily stress in the wild”. In: *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2014, pp. 72–79.

[4] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. “Mining association rules between sets of items in large databases”. In: *Acm sigmod record*. Vol. 22. 2. ACM. 1993, pp. 207–216.

[5] Suman Ambwani et al. “Negative affect, interpersonal perception, and binge eating behavior: An experience sampling study”. In: *International Journal of Eating Disorders* 48.6 (2015), pp. 715–726.

[6] Mawulolo K Ameko et al. “Cluster-based approach to improve affect recognition from passively sensed data”. In: *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE. 2018, pp. 434–437.

[7] David Lazer Anmol Madan Manuel Cebrian and Alex Pentland. “Social Sensing for Epidemiological Behavior Change”. In: *ACM UbiComp* (2010).

[8] Itamar Arel et al. "Reinforcement learning-based multi-agent system for network traffic signal control". In: *IET Intelligent Transport Systems* 4.2 (2010), pp. 128–135.

[9] Daniel Ashbrook and Thad Starner. "Using GPS to learn significant locations and predict movement across multiple users". In: *Personal and Ubiquitous computing* 7.5 (2003), pp. 275–286.

[10] Min Hane Aung, Mark Matthews, and Tanzeem Choudhury. "Sensing behavioral symptoms of mental health and delivering personalized interventions using mobile technologies". In: *Depression and anxiety* 34.7 (2017), pp. 603–609.

[11] Rajesh Krishna Balan et al. "The challenge of continuous mobile context sensing". In: *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*. IEEE. 2014, pp. 1–8.

[12] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal machine learning: A survey and taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2018), pp. 423–443.

[13] Marta Bassi and Antonella Delle Fave. "Flow in the Context of Daily Experience Fluctuation". In: *Flow Experience*. Springer, 2016, pp. 181–196.

[14] Daniel J Beal. "ESM 2.0: State of the art and future potential of experience sampling methods in organizational research". In: *Annu. Rev. Organ. Psychol. Organ. Behav.* 2.1 (2015), pp. 383–407.

[15] Fehmi Ben Abdesslem, Andrew Phillips, and Tristan Henderson. "Less is more: energy-efficient mobile sensing with senseless". In: *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*. ACM. 2009, pp. 61–62.

[16] Dror Ben-Zeev, Michael A Young, and Colin A Depp. "Real-time predictors of suicidal ideation: mobile assessment of hospitalized depressed patients". In: *Psychiatry research* 197.1 (2012), pp. 55–59.

[17] Dror Ben-Zeev et al. "Comparing retrospective reports to real-time/real-place mobile assessments in individuals with schizophrenia and a nonclinical comparison group". In: *Schizophrenia bulletin* 38.3 (2012), pp. 396–404.

[18] Dror Ben-Zeev et al. "CrossCheck: Integrating self-report, behavioral sensing, and smartphone use to identify digital indicators of psychotic relapse." In: *Psychiatric rehabilitation journal* 40.3 (2017), p. 266.

[19] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[20] Amit Birenboim et al. "High-resolution measurement and analysis of visitor experiences in time and space: The case of Aalborg zoo in Denmark". In: *The Professional Geographer* 67.4 (2015), pp. 620–629.

[21] Andrey Bogomolov et al. "Daily Stress Recognition from Mobile Phone Data, Weather Conditions and Individual Traits". In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2014, pp. 477–486. URL: http://arxiv.org/pdf/1410.5816.pdf.

[22] Matthias Böhmer et al. "Interrupted by a phone call: exploring designs for lowering the impact of call notifications for smartphone users". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2014, pp. 3045–3054. URL: http://matthiasboehmer.de/wp-content/uploads/2014/01/Boehmer-CHI2014-PhoneUI-CallHeads.pdf.

[23] Wendelin Böhmer et al. "Autonomous learning of state representations for control: An emerging field aims to autonomously learn state representations for reinforcement learning agents from their real-world sensor observations". In: *KI-Künstliche Intelligenz* 29.4 (2015), pp. 353–362.

[24] Mehdi Boukhechba et al. "ActiPPG: Using deep neural networks for activity recognition from wrist-worn photoplethysmography (PPG) sensors". In: *Smart Health* 14 (2019), p. 100082.

[25] Mehdi Boukhechba et al. "Contextual analysis to understand compliance with smartphone-based ecological momentary assessment". In: *Proceedings of the 12th EAI International Conference on Pervasive Computing Technologies for Healthcare*. 2018, pp. 232–238.

[26] Mehdi Boukhechba et al. *DemonicSalmon*. 2018. DOI: 10.17605/OSF.IO/WDUK6. URL: https://doi.org/10.17605/OSF.IO/WDUK6.

[27] Mehdi Boukhechba et al. "DemonicSalmon: Monitoring mental health and social interactions of college students using smartphones". In: *Smart Health* 9-10 (2018), pp. 192–203. DOI: 10.1016/j.smhl.2018.07.005. URL: https://doi.org/10.1016%2Fj.smhl.2018.07.005.

[28] Mehdi Boukhechba et al. "DemonicSalmon: Monitoring mental health and social interactions of college students using smartphones". In: *Smart Health* 9-10 (2018), pp. 192–203. DOI: 10.1016/j.smhl.2018.07.005. URL: https://doi.org/10.1016%2Fj.smhl.2018.07.005.

[29] Mehdi Boukhechba et al. "Energy Optimization for Outdoor Activity Recognition". In: *Journal of Sensors* 2016 (2016), pp. 1–15. DOI: 10.1155/2016/6156914. URL: https://doi.org/10.1155%2F2016%2F6156914.

[30] Mehdi Boukhechba et al. "Hybrid Battery-friendly Mobile Solution for Extracting Users' Visited Places". In: *Procedia Computer Science* 94 (2016), pp. 25–32. DOI: 10.1016/j.procs.2016.08.008. URL: https://doi.org/10.1016%2Fj.procs.2016.08.008.

[31] Mehdi Boukhechba et al. "Monitoring social anxiety from mobility and communication patterns". In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers on - UbiComp 17*. ACM Press, 2017. DOI: 10.1145/3123024.3125607. URL: https://doi.org/10.1145%2F3123024.3125607.

[32] Mehdi Boukhechba et al. "Predicting social anxiety from global positioning system traces of college students: feasibility study". In: *JMIR mental health* 5.3 (2018), e10101.

[33] Morten Brandrup et al. "Effects-Driven Participatory Design: Learning from Sampling Interruptions". In: *Participatory Design and Health Information Technology*. IOS Press, 2017, pp. 113–127.

[34] Michael Broda. "Using Multilevel Models to Explore Predictors of High School Students' Nonresponse in Experience Sampling Method (ESM) Studies". In: *Social Science Computer Review* (2016), p. 0894439316667049.

[35] Lihua Cai et al. "Adaptive Passive Mobile Sensing Using Reinforcement Learning". In: *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2019, pp. 1–6.

[36] Lihua Cai et al. "An integrated framework for using mobile sensing to understand response to mobile interventions among breast cancer patients". In: *Smart Health* 15 (2020), p. 100086.

[37] Lihua Cai et al. "State Affect Recognition Using Smartphone Sensing Data". In: *Proceedings of the 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies*. CHASE '18. Washington, DC: Association for Computing Machinery, 2018, 120–125. ISBN: 9781450359580. DOI: 10.1145/3278576.3284386. URL: https://doi.org/10.1145/3278576.3284386.

[38] Francesco Calabrese et al. "Real-time urban monitoring using cell phones: A case study in Rome". In: *IEEE Transactions on Intelligent Transportation Systems* 12.1 (2010), pp. 141–151.

[39] Luca Canzian and Mirco Musolesi. "Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis". In: *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. ACM. 2015, pp. 1293–1304.

[40] Xin Cao, Gao Cong, and Christian S Jensen. "Mining significant semantic locations from GPS data". In: *Proceedings of the VLDB Endowment* 3.1-2 (2010), pp. 1009–1020.

[41] Andrea Capponi et al. "A Survey on Mobile Crowdsensing Systems: Challenges, Solutions, and Opportunities". In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2419–2465.

[42] Giuseppe Cardone et al. "MSF: An efficient mobile phone sensing framework". In: *International Journal of Distributed Sensor Networks* 9.3 (2013), p. 538937.

[43] Yung-Ju Chang, Gaurav Paruthi, and Mark W Newman. "A field study comparing approaches to collecting annotated activity data in real-world settings". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 671–682.

[44] Fanglin Chen et al. "My smartphone knows i am hungry". In: *Proceedings of the 2014 workshop on physical analytics*. ACM. 2014, pp. 9–14.

[45] Yuqing Chen and Yang Xue. "A deep learning approach to human activity recognition based on single accelerometer". In: *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE. 2015, pp. 1488–1492.

[46] Zhenyu Chen et al. "ContextSense: unobtrusive discovery of incremental social context using dynamic bluetooth data". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM. 2014, pp. 23–26.

[47] Zhenyu Chen et al. "Unobtrusive sleep monitoring using smartphones". In: *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and ... 2013, pp. 145–152.

[48] Philip I Chow et al. “Using Mobile Sensing to Test Clinical Models of Depression, Social Anxiety, State Affect, and Social Isolation Among College Students”. In: *Journal of Medical Internet Research* 19.3 (2017).

[49] Ionut Constandache et al. “Enloc: Energy-efficient localization for mobile phones”. In: *IEEE INFOCOM 2009*. IEEE. 2009, pp. 2716–2720.

[50] Victor P Cornet and Richard J Holden. “Systematic review of smartphone-based passive sensing for health and wellbeing”. In: *Journal of biomedical informatics* 77 (2018), pp. 120–132.

[51] Jean Costa et al. “EmotionCheck: A Wearable Device to Regulate Anxiety through False Heart Rate Feedback”. In: *GetMobile: Mobile Computing and Communications* 21.2 (2017), pp. 22–25.

[52] Katharine E Daniel et al. “Do I really feel better? Effectiveness of emotion regulation strategies depends on the measure and social anxiety”. In: *Depression and anxiety* (2019).

[53] Alexander R Daros et al. “Relationships between trait emotion dysregulation and emotional experiences in daily life: an experience sampling study”. In: *Cognition and Emotion* (2019), pp. 1–13.

[54] Soumya Kanti Datta et al. “oneM2M architecture based IoT framework for mobile crowd sensing in smart cities”. In: *2016 European conference on networks and communications (EuCNC)*. IEEE. 2016, pp. 168–173.

[55] Afsaneh Doryab et al. “Detection of behavior change in people with depression”. In: *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*. 2014.

[56] Afsaneh Doryab et al. “Extraction of Behavioral Features from Smartphone and Wearable Data”. In: *arXiv preprint arXiv:1812.10394* (2018).

[57] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. “Challenges of real-world reinforcement learning”. In: *arXiv preprint arXiv:1904.12901* (2019).

[58] Nathan Eagle and Alex Sandy Pentland. “Eigenbehaviors: Identifying structure in routine”. In: *Behavioral Ecology and Sociobiology* 63.7 (2009), pp. 1057–1066.

[59] Asli Enez Darcin et al. “Smartphone addiction and its relationship with social anxiety and loneliness”. In: *Behaviour & Information Technology* 35.7 (2016), pp. 520–525.

[60] Karoly Farkas et al. “Crowdsending based public transport information service in smart cities”. In: *IEEE Communications Magazine* 53.8 (2015), pp. 158–165.

[61] Shu Farmer et al. “Mobile Phone Ecological Momentary Assessment of Daily Stressors among People Living With HIV: Elucidating Factors Underlying Health-Related Challenges in Daily Routines”. In: *Journal of the Association of Nurses in AIDS Care* (2017).

[62] Arsham Farshad, Mahesh K Marina, and Francisco Garcia. “Urban WiFi characterization via mobile crowdsensing”. In: *2014 IEEE Network Operations and Management Symposium (NOMS)*. IEEE. 2014, pp. 1–9.

[63] Matthew Faulkner et al. “Community sense and response systems: Your phone as quake detector”. In: *Communications of the ACM* 57.7 (2014), pp. 66–75.

[64] Katya C Fernandez, Matthew R Johnson, and Thomas L Rodebaugh. “TelEMA: a low-cost and user-friendly telephone assessment platform”. In: *Behavior research methods* 45.4 (2013), pp. 1279–1291.

[65] Denzil Ferreira et al. “Contextual experience sampling of mobile application micro-usage”. In: *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM. 2014, pp. 91–100.

[66] Davide Figo et al. “Preprocessing techniques for context recognition from accelerometer data”. In: *Personal and Ubiquitous Computing* 14.7 (2010), pp. 645–662.

[67] Gabriel Filios et al. "Hierarchical algorithm for daily activity recognition via smartphone sensors". In: *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE. 2015, pp. 381–386. URL: http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=7389084&tag=1.

[68] Joel Fischer. "Understanding receptivity to interruptions in mobile human-computer interaction". PhD thesis. University of Nottingham, 2011. URL: http://eprints.nottingham.ac.uk/12499/1/PhDThesis-JoelFischer.pdf.

[69] Andreas Fsr, Jim Torresen, et al. "Smartphone accelerometer data used for detecting human emotions". In:

[70] Raghu K Ganti, Fan Ye, and Hui Lei. "Mobile crowdsensing: current state and future challenges". In: *IEEE Communications Magazine* 49.11 (2011), pp. 32–39.

[71] Yusong Gao et al. "How smartphone usage correlates with social anxiety and loneliness". In: *PeerJ* 4 (2016), e2197.

[72] Tarek F Gharib et al. "An efficient algorithm for incremental mining of temporal association rules". In: *Data & Knowledge Engineering* 69.8 (2010), pp. 800–815.

[73] Karen Glanz, Barbara K Rimer, and Kasisomayajula Viswanath. *Health behavior and health education: theory, research, and practice*. John Wiley & Sons, 2008.

[74] Manish Kumar Goel, Pardeep Khanna, and Jugal Kishore. "Understanding survival analysis: Kaplan-Meier estimate". In: *International journal of Ayurveda research* 1.4 (2010), p. 274.

[75] Thomas Goetz, Madeleine Bieg, and Nathan C Hall. "Assessing Academic Emotions via the Experience Sampling Method". In: *Methodological Advances in Research on Emotion and Education*. Springer, 2016, pp. 245–258.

[76] Jiaqi Gong et al. "Understanding behavioral dynamics of social anxiety among college students through smartphone sensors". In: *Information Fusion* 49 (2019), pp. 57 –68. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.

2018.09.002. URL: http://www.sciencedirect.com/science/article/pii/S1566253517307352.

[77] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. “Understanding individual human mobility patterns”. In: *nature* 453.7196 (2008), pp. 779–782.

[78] Eric Granholm et al. “Mobile Assessment and Treatment for Schizophrenia (MATS): a pilot trial of an interactive text-messaging intervention for medication adherence, socialization, and auditory hallucinations”. In: *Schizophrenia bulletin* (2011), sbr155.

[79] Nils Y Hammerla et al. “On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution”. In: *Proceedings of the 2013 International Symposium on Wearable Computers*. ACM. 2013, pp. 65–68.

[80] Gabriella M Harari et al. “Patterns of behavior change in students over an academic term: A preliminary study of activity and sociability behaviors using smartphone sensing methods”. In: *Computers in Human Behavior* 67 (2017), pp. 129–138. URL: http://www.sciencedirect.com/science/article/pii/S0747563216307282.

[81] Gabriella M Harari et al. “Smartphone sensing methods for studying behavior in everyday life”. In: *Current Opinion in Behavioral Sciences* 18 (2017), pp. 83–90.

[82] Gabriella M Harari et al. “Using Smartphones to Collect Behavioral Data in Psychological Science Opportunities, Practical Considerations, and Challenges”. In: *Perspectives on Psychological Science* 11.6 (2016), pp. 838–854. URL: http://pps.sagepub.com/content/11/6/838.full.pdf+html.

[83] David Hasenfratz et al. “Participatory air pollution monitoring using smartphones”. In: *Mobile Sensing* 1 (2012), pp. 1–5.

[84] Kristin E Heron and Joshua M Smyth. “Ecological momentary interventions: incorporating mobile technology into psychosocial and health behaviour treatments”. In: *British journal of health psychology* 15.1 (2010), pp. 1–39.

[85] Stefan Hey et al. "Recent developments of ambulatory assessment methods". In: *Zeitschrift für Neuropsychologie* (2014).

[86] Joyce Ho and Stephen S Intille. "Using context-aware computing to reduce the perceived burden of interruptions from mobile devices". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2005, pp. 909–918.

[87] Wilhelm Hofmann and Paresh V Patel. "SurveySignal: A convenient solution for experience sampling research using participants' own smartphones". In: *Social Science Computer Review* 33.2 (2015), pp. 235–253.

[88] Eric Horvitz, Andy Jacobs, and David Hovel. "Attention-sensitive alerting". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1999, pp. 305–313.

[89] Seyed Amir Hoseini-Tabatabaei, Alexander Gluhak, and Rahim Tafazolli. "A survey on smartphone-based systems for opportunistic user context recognition". In: *ACM Computing Surveys (CSUR)* 45.3 (2013), p. 27.

[90] Yitong Huang et al. "Designing for Human Sustainability: The Role of Self-Reflection". In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. ACM. 2015, pp. 1042–1045.

[91] Galen Chin-Lun Hung et al. "Predicting Negative Emotions Based on Mobile Phone Usage Patterns: An Exploratory Study". In: *JMIR Research Protocols* 5.3 (2016).

[92] Tâm Huynh, Mario Fritz, and Bernt Schiele. "Discovery of activity patterns using topic models". In: *Proceedings of the 10th international conference on Ubiquitous computing*. ACM. 2008, pp. 10–19. URL: http://www.mobvis.org/publications/ubicomp2008.pdf.

[93] Stephen S Intille et al. "Tools for studying behavior and technology in natural settings". In: *International Conference on Ubiquitous Computing*. Springer. 2003, pp. 157–174.

[94] Prem Prakash Jayaraman et al. "Here-n-now: A framework for context-aware mobile crowdsensing". In: *Proc. of the Tenth International Conference on Pervasive Computing*. Citeseer. 2012.

[95] Wenjun Jiang et al. "Towards Personalized Learning in Mobile Sensing Systems". In: *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2018, pp. 321–333.

[96] Junqi Jin et al. "Real-time bidding with multi-agent reinforcement learning in display advertising". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 2193–2201.

[97] Jong Hee Kang et al. "Extracting places from traces of locations". In: *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*. ACM. 2004, pp. 110–118.

[98] Aman Kansal et al. "The latency, accuracy, and battery (LAB) abstraction: programmer productivity and energy efficiency for continuous mobile context sensing". In: *ACM SIGPLAN Notices*. Vol. 48. 10. ACM. 2013, pp. 661–676.

[99] Robert M Kaplan and Arthur A Stone. "Bringing the laboratory and clinic to the community: mobile technologies for health promotion and disease prevention a". In: *Annual Review of Psychology* 64 (2013), pp. 471–498.

[100] James Kelly et al. "Intelligent real-time therapy: Harnessing the power of machine learning to optimise the delivery of momentary cognitive–behavioural interventions". In: *Journal of Mental Health* 21.4 (2012), pp. 404–414.

[101] Adil Mehmood Khan et al. "Activity recognition on smartphones via sensor-fusion and kda-based svms". In: *International Journal of Distributed Sensor Networks* 10.5 (2014), p. 503291.

[102] Aftab Khan et al. "Optimising sampling rates for accelerometer-based human activity recognition". In: *Pattern Recognition Letters* ().

[103] Kyu-Han Kim et al. “Improving energy efficiency of Wi-Fi sensing on smartphones”. In: *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, pp. 2930–2938.

[104] Sunyoung Kim et al. “Creek watch: pairing usefulness and usability for successful citizen science”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 2125–2134.

[105] David Kimhy et al. “Mobile assessment guide for research in schizophrenia and severe mental disorders”. In: *Schizophrenia bulletin* (2012), sbr186.

[106] Petra Klumb, Achim Elfering, and Christiane Herre. “Ambulatory assessment in industrial/organizational psychology: Fruitful examples and methodological issues”. In: *European Psychologist* 14.2 (2009), pp. 120–131.

[107] Jens Kober, J Andrew Bagnell, and Jan Peters. “Reinforcement learning in robotics: A survey”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274.

[108] Ingrid Kramer et al. “A therapeutic application of the experience sampling method in the treatment of depression: a randomized controlled trial”. In: *World Psychiatry* 13.1 (2014), pp. 68–77.

[109] Andreas Krause et al. “Trading off prediction accuracy and power consumption for context-aware wearable computing”. In: *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*. IEEE. 2005, pp. 20–26.

[110] Hyeokhyen Kwon, Gregory D Abowd, and Thomas Plötz. “Adding structural characteristics to distribution-based accelerometer representations for activity recognition using wearables”. In: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM. 2018, pp. 72–75.

[111] Constantino M Lagoa et al. “Designing adaptive intensive interventions using methods from engineering.” In: *Journal of consulting and clinical psychology* 82.5 (2014), p. 868.

[112] Nicholas D Lane et al. "A survey of mobile phone sensing". In: *IEEE Communications magazine* 48.9 (2010), pp. 140–150.

[113] Nicholas D Lane et al. "Deepx: A software accelerator for low-power deep learning inference on mobile devices". In: *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*. IEEE Press. 2016, p. 23.

[114] Neal Lathia et al. "Contextual dissonance: Design bias in sensor-based experience sampling methods". In: *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. 2013, pp. 183–192.

[115] Neal Lathia et al. "Happier people live more active lives: Using smartphones to link happiness and physical activity". In: *PloS one* 12.1 (2017), e0160589.

[116] Nadja Lederer, Thomas Grechenig, and René Baranyi. "unCUT: bridging the gap from paper diary cards towards mobile electronic monitoring solutions in borderline and self-injury". In: *Serious Games and Applications for Health (SeGAH), 2014 IEEE 3rd International Conference on*. IEEE. 2014, pp. 1–8.

[117] Jonathan Lester et al. "A Hybrid Discriminative/Generative Approach for Modeling Human Activities". In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. IJCAI'05. Edinburgh, Scotland: Morgan Kaufmann Publishers Inc., 2005, pp. 766–772. URL: http://dl.acm.org/citation.cfm?id=1642293.1642416.

[118] Xueying Li et al. "Learning to infer the status of heavy-duty sensors for energy-efficient context-sensing". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 3.2 (2012), p. 35.

[119] Yuxi Li. "Deep reinforcement learning: An overview". In: *arXiv preprint arXiv:1701.07274* (2017).

[120] Celine Liang et al. "Learning Treatment Policies for Mobile Health Using Randomized Least-Squares Value Iteration". In: ().

[121] Yunji Liang et al. “Energy-efficient motion related activity recognition on mobile devices for pervasive healthcare”. In: *Mobile Networks and Applications* 19.3 (2014), pp. 303–317.

[122] Peng Liao et al. “Just-in-Time but Not Too Much: Determining Treatment Timing in Mobile Health”. In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 2.4 (2018), p. 179.

[123] Robert LiKamWa et al. “Moodscope: Building a mood sensor from smartphone usage patterns”. In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM. 2013, pp. 389–402.

[124] Kaisen Lin et al. “Energy-accuracy trade-off for continuous mobile device location”. In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM. 2010, pp. 285–298.

[125] Miao Lin and Wen-Jing Hsu. “Mining GPS data for mobility patterns: A survey”. In: *Pervasive and mobile computing* 12 (2014), pp. 1–16.

[126] Miao Lin, Wen-Jing Hsu, and Zhuo Qi Lee. “Detecting modes of transport from unlabelled positioning sensor data”. In: *Journal of Location Based Services* 7.4 (2013), pp. 272–290.

[127] Miao Lin, Wen-Jing Hsu, and Zhuo Qi Lee. “Predictability of individuals’ mobility with high-resolution positioning data”. In: *Proceedings of the 2012 ACM conference on ubiquitous computing*. ACM. 2012, pp. 381–390.

[128] Jinwei Liu et al. “A survey of mobile crowdsensing techniques: A critical component for the internet of things”. In: *ACM Transactions on Cyber-Physical Systems* 2.3 (2018), pp. 1–26.

[129] Karen Kay-Lynn Liu. “A personal, mobile system for understanding stress and interruptions”. PhD thesis. Citeseer, 2004.

[130] Ye Liu et al. "From action to activity: Sensor-based activity recognition". In: *Neurocomputing* (2015). URL: http://www.sciencedirect.com/science/article/pii/S0925231215016331.

[131] Hong Lu et al. "SoundSense: scalable sound sensing for people-centric applications on mobile phones". In: *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM. 2009, pp. 165–178. URL: http://cvrr.ucsd.edu/ece285/papers/Lu_soundsense.pdf.

[132] Hong Lu et al. "Speakersense: Energy efficient unobtrusive speaker identification on mobile phones". In: *International conference on pervasive computing*. Springer. 2011, pp. 188–205.

[133] Hong Lu et al. "Stresssense: Detecting stress in unconstrained acoustic environments using smartphones". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 351–360.

[134] Hong Lu et al. "The Jigsaw continuous sensing engine for mobile phone applications". In: *Proceedings of the 8th ACM conference on embedded networked sensor systems*. ACM. 2010, pp. 71–84.

[135] Huadong Ma, Dong Zhao, and Peiyan Yuan. "Opportunities in mobile crowd sensing". In: *IEEE Communications Magazine* 52.8 (2014), pp. 29–35.

[136] Yuanchao Ma et al. "Daily mood assessment based on mobile phone sensing". In: *2012 ninth international conference on wearable and implantable body sensor networks*. IEEE. 2012, pp. 142–147.

[137] Elsa Macias, Alvaro Suarez, and Jaime Lloret. "Mobile sensing systems". In: *Sensors* 13.12 (2013), pp. 17292–17321.

[138] George MacKerron and Susana Mourato. "Happiness is greater in natural environments". In: *Global Environmental Change* 23.5 (2013), pp. 992–1000.

[139] Hongzi Mao et al. "Resource management with deep reinforcement learning". In: *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. 2016, pp. 50–56.

[140] Gabriela Marcu et al. "Collecting and Managing Data in Public Health Research on Autism". In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM. 2017, pp. 2767–2773.

[141] Panos Markopoulos, Nikolaos Batalas, and Annick Timmermans. "On the use of personalization to enhance compliance in experience sampling". In: *Proceedings of the European Conference on Cognitive Ergonomics 2015*. ACM. 2015, p. 15.

[142] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. "PrefMiner: mining user's preferences for intelligent mobile notification management". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 1223–1234.

[143] Abhinav Mehrotra and Mirco Musolesi. "Using autoencoders to automatically extract mobility features for predicting depressive states". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.3 (2018), p. 127.

[144] Abhinav Mehrotra, Veljko Pejovic, and Mirco Musolesi. "SenSocial: a middleware for integrating online social networks and mobile sensing data streams". In: *Proceedings of the 15th International Middleware Conference*. ACM. 2014, pp. 205–216.

[145] Abhinav Mehrotra et al. "Ask, but don't interrupt: the case for interruptibility-aware mobile experience sampling". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM. 2015, pp. 723–732. URL: http://dl.acm.org/citation.cfm?id=2804397.

[146] Abhinav Mehrotra et al. "Designing content-driven intelligent notification mechanisms for mobile applications". In: *Proceedings of the 2015 ACM International*

Joint Conference on Pervasive and Ubiquitous Computing. ACM. 2015, pp. 813–824.

[147] Abhinav Mehrotra et al. "My Phone and Me: Understanding People's Receptivity to Mobile Notifications". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. 2016, pp. 1021–1032.

[148] Marianne Menictas et al. "Artificial intelligence decision-making in mobile health". In: *The Biochemist* 41.5 (2019), pp. 20–24.

[149] G Mikelsons et al. "Towards Deep Learning Models for Psychological State Prediction using Smartphone Data: Challenges and Opportunities". In: *CoRR*. 2017.

[150] Jun-Ki Min et al. "Toss'n'turn: smartphone as sleep and sleep quality detector". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2014, pp. 477–486.

[151] Prashanth Mohan, Venkata N Padmanabhan, and Ramachandran Ramjee. "Nericell: rich monitoring of road and traffic conditions using mobile smartphones". In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*. 2008, pp. 323–336.

[152] David C Mohr, Mi Zhang, and Stephen M Schueller. "Personal sensing: understanding mental health using ubiquitous sensors and machine learning". In: *Annual review of clinical psychology* 13 (2017), pp. 23–47.

[153] Megan A Moreno et al. "Depression and internet use among older adolescents: an experience sampling approach". In: *Psychology* 3.09 (2012), p. 743.

[154] Inbal Nahum-Shani, Eric B Hekler, and Donna Spruijt-Metz. "Building health behavior models to guide the development of just-in-time adaptive interventions: A pragmatic framework." In: *Health Psychology* 34.S (2015), p. 1209.

[155] Inbal Nahum-Shani et al. "Just in time adaptive interventions (jitais): An organizing framework for ongoing health behavior support". In: *Methodology Center technical report* 2014 (2014), pp. 14–126.

[156] Felix Naughton et al. "A context-sensing mobile phone app (Q sense) for smoking cessation: a mixed-methods study". In: *JMIR mHealth and uHealth* 4.3 (2016), e106.

[157] Tadashi Okoshi et al. "Attelia: Reducing User's Cognitive Load due to Interruptive Notifications on Smart Phones". In: (2015), pp. 96–104. URL: https://www.ht.sfc.keio.ac.jp/~slash/papers/Okoshi2015a_Attelia1_PerCom15.pdf.

[158] Tadashi Okoshi et al. "Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 475–486. URL: http://s3.amazonaws.com/academia.edu.documents/38984036/2_Attelia-Ubicomp2015-CameraReady_20150721_v2.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1476305179&Signature=MN61L6pJaPUQufX8vIofNaKyj2k%3D&response-content-disposition=inline%3B%20filename%3DReducing_Users_Perceived_Mental_Effort_d.pdf.

[159] Margreet Oorschot et al. "Mobile assessment in schizophrenia: a data-driven momentary approach". In: *Schizophrenia bulletin* 38.3 (2011), pp. 405–413.

[160] Jim van Os et al. "Testing an mHealth momentary assessment Routine Outcome Monitoring application: a focus on restoration of daily life positive mood states". In: *PLoS One* 9.12 (2014), e115254.

[161] Thomas Olutoyin Oshin, Stefan Poslad, and Athen Ma. "Improving the energy-efficiency of GPS based location sensing smartphone applications". In: *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE. 2012, pp. 1698–1705.

[162] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. "Energy-efficient rate-adaptive GPS-based positioning for smartphones". In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM. 2010, pp. 299–314.

[163] Jasper E Palmier-Claus et al. “Experience sampling research in individuals with mental illness: reflections and guidance”. In: *Acta Psychiatrica Scandinavica* 123.1 (2011), pp. 12–20.

[164] Veljko Pejovic and Mirco Musolesi. “InterruptMe: Designing intelligent prompting mechanisms for pervasive applications”. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2014, pp. 897–908. URL: http://ubhave.org/publications/pejovic_ubicomp14.pdf.

[165] Veljko Pejovic, Mirco Musolesi, and Abhinav Mehrotra. “Investigating The Role of Task Engagement in Mobile Interruptibility”. In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. ACM. 2015, pp. 1100–1105.

[166] Veljko Pejovic et al. “Mobile-based experience sampling for behaviour research”. In: *Emotions and Personality in Personalized Services*. Springer, 2016, pp. 141–161.

[167] Charith Perera et al. “Energy-efficient location and activity-aware on-demand mobile distributed sensing platform for sensing as a service in IoT clouds”. In: *IEEE Transactions on Computational Social Systems* 2.4 (2015), pp. 171–181.

[168] Martin Pielot et al. “Didn’t you see my message?: predicting attentiveness to mobile instant messages”. In: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM. 2014, pp. 3319–3328. URL: http://pielot.org/pubs/Pielot2014-CHI-AttPred.pdf.

[169] Skyler Place et al. “Behavioral Indicators on a Mobile Sensing Platform Predict Clinically Validated Psychiatric Symptoms of Mood and Anxiety Disorders”. In: *Journal of Medical Internet Research* 19.3 (2017).

[170] Thomas Plötz and Yu Guan. “Deep Learning for Human Activity Recognition in Mobile Computing”. In: *Computer* 51.5 (2018), pp. 50–59.

[171] Thomas Plötz, Nils Y Hammerla, and Patrick L Olivier. "Feature learning for activity recognition in ubiquitous computing". In: *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.

[172] Stephen J Preece et al. "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data". In: *IEEE Transactions on Biomedical Engineering* 56.3 (2008), pp. 871–879.

[173] Bodhi Priyantha, Dimitrios Lymberopoulos, and Jie Liu. "Enabling energy efficient continuous sensing on mobile phones with littlerock". In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM. 2010, pp. 420–421.

[174] Alexander J Quinn and Benjamin B Bederson. "Human computation: a survey and taxonomy of a growing field". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2011, pp. 1403–1412.

[175] Moo-Ryong Ra et al. "Improving energy efficiency of personal sensing applications with heterogeneous multi-processors". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 1–10.

[176] Mashfiqui Rabbi et al. "MyBehavior: automatic personalized health feedback from user behaviors and preferences using smartphones". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 707–718.

[177] Mashfiqui Rabbi et al. "SARA: A Mobile App to Engage Users in Health Data Collection". In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. UbiComp '17. Maui, Hawaii: ACM, 2017, pp. 781–789. ISBN: 978-1-4503-5190-4. DOI: 10.1145/3123024.3125611. URL: http://doi.acm.org/10.1145/3123024.3125611.

[178] Kiran K Rachuri, Mirco Musolesi, and Cecilia Mascolo. "Energy-accuracy trade-offs in querying sensor data for continuous sensing mobile systems". In: *Proc. of Mobile Context-Awareness Workshop*. Vol. 10. 2010.

[179] Kiran K Rachuri et al. "Sociablesense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing". In: *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM. 2011, pp. 73–84.

[180] Valentin Radu et al. "Towards multimodal deep learning for activity recognition on mobile devices". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*.

[181] Reza Rawassizadeh et al. "Energy-efficient integration of continuous context sensing and prediction into smartwatches". In: *Sensors* 15.9 (2015), pp. 22616–22645.

[182] Sasank Reddy et al. "Using mobile phones to determine transportation modes". In: *ACM Transactions on Sensor Networks (TOSN)* 6.2 (2010), p. 13.

[183] Ulrich Reininghaus, Colin A Depp, and Inez Myin-Germeys. "Ecological interventionist causal models in psychosis: targeting psychological mechanisms in daily life". In: *Schizophrenia bulletin* 42.2 (2016), pp. 264–269.

[184] HARRY T Reis et al. "Methods for studying everyday experience in its natural context". In: *Handbook of research methods in social and personality psychology* 2 (2014), pp. 373–403.

[185] William T Riley et al. "Health behavior models in the age of mobile interventions: are our theories up to the task?" In: *Translational behavioral medicine* 1.1 (2011), pp. 53–71.

[186] Megan L Robbins, Thomas Kubiak, and DI Mostofsky. "Ecological monetary assessment in behavioral medicine: research and practice". In: *The handbook of behavioral medicine* 46 (2014).

[187] Sohrab Saeb et al. "Mobile phone detection of semantic location and its relationship to depression and anxiety". In: *JMIR mHealth and uHealth* 5.8 (2017), e112.

[188] Sohrab Saeb et al. "Mobile phone sensor correlates of depressive symptom severity in daily-life behavior: an exploratory study". In: *Journal of medical Internet research* 17.7 (2015).

[189] Kartik Sankaran et al. "Using mobile phone barometer for low-power transportation context detection". In: *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM. 2014, pp. 191–205. URL: http://projects.csail.mit.edu/wiki/pub/LSPgroup/PublicationList/baro_sensys2014.pdf.

[190] Iqbal H Sarker, Alan Colman, and Jun Han. "RecencyMiner: mining recency-based personalized behavior from contextual smartphone data". In: *Journal of Big Data* 6.1 (2019), p. 49.

[191] Iqbal H Sarker and Flora D Salim. "Mining user behavioral rules from smartphone data through association analysis". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2018, pp. 450–461.

[192] JA van Schendel et al. "A Matter of Time: The influence of Context-Based Timing on Compliance With Well-Being Triggers". In: *Eindhoven University of Technology* (2015).

[193] Christian M Schneider et al. "Unravelling daily human mobility motifs". In: *Journal of The Royal Society Interface* 10.84 (2013), p. 20130246.

[194] Fuschia Serre et al. "Ecological momentary assessment in alcohol, tobacco, cannabis and opiate dependence: A comparison of feasibility and validity". In: *Drug and alcohol dependence* 126.1 (2012), pp. 118–123.

[195] Sandra Servia-Rodríguez et al. "Mobile Sensing at the Service of Mental Wellbeing: a Large-scale Longitudinal Study". In: (2017).

[196] Saul Shiffman, Arthur A Stone, and Michael R Hufford. "Ecological momentary assessment". In: *Annu. Rev. Clin. Psychol.* 4 (2008), pp. 1–32.

[197] Erez Shmueli et al. "Sensing, Understanding, and Shaping Social Behavior". In: *IEEE Transactions on Computational Social Systems* 1 (2014), pp. 22–34. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6804686.

[198] Noam Shoval, Yonatan Schvimer, and Maya Tamir. "Real-Time Measurement of Tourists' Objective and Subjective Emotions in Time and Space". In: *Journal of Travel Research* (2017), p. 0047287517691155.

[199] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), p. 484.

[200] Ida Sim. "Mobile devices and health". In: *New England Journal of Medicine* 381.10 (2019), pp. 956–968.

[201] Alexander W Sokolovsky, Robin J Mermelstein, and Donald Hedeker. "Factors predicting compliance to ecological momentary assessment among adolescent smokers". In: *nicotine & tobacco research* 16.3 (2014), pp. 351–358.

[202] Donna Spruijt-Metz et al. "Building new computational models to support health behavior change and maintenance: new opportunities in behavioral research". In: *Translational behavioral medicine* 5.3 (2015), pp. 335–346.

[203] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[204] David Taniar and John Goh. "On mining movement pattern from mobile users". In: *International Journal of Distributed Sensor Networks* 3.1 (2007), pp. 69–86. URL: http://www.tandfonline.com/doi/abs/10.1080/15501320601069499.

[205] Kyle Taylor and Laura Silver. "Smartphone ownership is growing rapidly around the world, but not always equally". In: *Pew Research Center* (2019).

[206] C Telford et al. "Experience sampling methodology studies of depression: the state of the art". In: *Psychological medicine* 42.06 (2012), pp. 1119–1129.

[207] Kevin P Timms et al. "A dynamical systems approach to understanding self-regulation in smoking cessation behavior change". In: *nicotine & tobacco research* 16.Suppl_2 (2013), S159–S168.

[208] Debra Trampe, Jordi Quoidbach, and Maxime Taquet. "Emotions in everyday life". In: *PloS one* 10.12 (2015), e0145450.

[209] Achyut Mani Tripathi, Diganta Baruah, and Rashmi Dutta Baruah. "Acoustic sensor based activity recognition using ensemble of one-class classifiers". In: *Evolving and Adaptive Intelligent Systems (EAIS), 2015 IEEE International Conference on*. IEEE. 2015, pp. 1–7. URL: http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=7368798.

[210] Timothy J Trull and Ulrich Ebner-Priemer. "Ambulatory assessment". In: *Annual review of clinical psychology* 9 (2013), pp. 151–176.

[211] Timothy J Trull and Ulrich W Ebner-Priemer. "Using experience sampling methods/ecological momentary assessment (ESM/EMA) in clinical assessment and clinical research: introduction to the special section." In: (2009).

[212] Vincent WS Tseng et al. "Assessing mental health issues on college campuses: preliminary findings from a pilot study". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM. 2016, pp. 1200–1208.

[213] Christopher Twigg. "Catmull-rom splines". In: *Computer* 41.6 (2003), pp. 4–6.

[214] Hugo Vachon et al. "Repeated self-evaluations may involve familiarization: An exploratory study related to Ecological Momentary Assessment designs in patients with major depressive disorder". In: *Psychiatry Research* 245 (2016), pp. 99–104.

[215] Simone JW Verhagen et al. "Use of the experience sampling method in the context of clinical trials". In: *Evidence Based Mental Health* 19.3 (2016), pp. 86–89.

[216] Sudip Vhaduri and Christian Poellabauer. "Design Factors of Longitudinal Smartphone-based Health Surveys". In: *Journal of Healthcare Informatics Research* (2017), pp. 1–40.

[217] Sudip Vhaduri and Christian Poellabauer. "Human Factors in the Design of Longitudinal Smartphone-based Wellness Surveys". In: *Healthcare Informatics (ICHI), 2016 IEEE International Conference on*. IEEE. 2016, pp. 156–167.

[218] Blake Wagner III et al. "e wrapper: operationalizing engagement strategies in mHealth". In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM. 2017, pp. 790–798.

[219] Leye Wang et al. "effSense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45.12 (2015), pp. 1549–1563.

[220] Leye Wang et al. "Sparse mobile crowdsensing: challenges and opportunities". In: *IEEE Communications Magazine* 54.7 (2016), pp. 161–167.

[221] Rui Wang et al. "CrossCheck: toward passive sensing and detection of mental health changes in people with schizophrenia". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 886–897.

[222] Rui Wang et al. "Predicting Symptom Trajectories of Schizophrenia using Mobile Sensing". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (2017), p. 110.

[223] Rui Wang et al. "SmartGPA: how smartphones can assess and predict academic performance of college students". In: *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. ACM. 2015, pp. 295–306.

[224] Rui Wang et al. "StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones". In: *Proceedings of the*

2014 ACM international joint conference on pervasive and ubiquitous computing. ACM. 2014, pp. 3–14.

[225] Rui Wang et al. "Tracking depression dynamics in college students using mobile phone and wearable sensing". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (2018), p. 43.

[226] Weichen Wang et al. "Sensing Behavioral Change over Time: Using Within-Person Variability Features from Mobile Sensing to Predict Personality Traits". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.3 (2018), p. 141.

[227] Yi Wang et al. "A framework of energy efficient mobile sensing for automatic user state recognition". In: *Proceedings of the 7th international conference on Mobile systems, applications, and services.* ACM. 2009, pp. 179–192.

[228] Yi Wang et al. "The tradeoff between energy efficiency and user state estimation accuracy in mobile sensing". In: *International Conference on Mobile Computing, Applications, and Services.* Springer. 2009, pp. 42–58.

[229] Marieke Wichers et al. "From affective experience to motivated action: Tracking reward-seeking and punishment-avoidant behaviour in real-life". In: *PloS one* 10.6 (2015), e0129722.

[230] Emma H Wood and Jonathan Moss. "Capturing emotions: experience sampling at live music events". In: *Arts and the Market* 5.1 (2015), pp. 45–72.

[231] Kieran Woodward et al. "Beyond Mobile Apps: A Survey of Technologies for Mental Well-being". In: *arXiv preprint arXiv:1905.00288* (2019).

[232] Tyler B Wray, Jennifer E Merrill, and Peter M Monti. "Using ecological momentary assessment (EMA) to assess situation-level predictors of alcohol use and alcohol-related consequences". In: *Alcohol research: current reviews* 36.1 (2014), p. 19.

[233] Congyu Wu et al. "Improving momentary stress measurement and prediction with bluetooth encounter networks". In: *Smart Health* 9 (2018), pp. 219–231.

[234] Congyu Wu et al. "Vector Space Representation of Bluetooth Encounters for Mental Health Inference". In: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. UbiComp '18. Singapore, Singapore: ACM, 2018, pp. 1691–1699. ISBN: 978-1-4503-5966-5. DOI: 10.1145/3267305.3267535. URL: http://doi.acm.org/10.1145/3267305.3267535.

[235] Pang Wu, Jiang Zhu, and Joy Ying Zhang. "Mobisens: A versatile mobile sensing platform for real-world applications". In: *Mobile Networks and Applications* 18.1 (2013), pp. 60–80.

[236] Haoyi Xiong et al. "Sensus: a cross-platform, general-purpose system for mobile crowdsensing in human-subject studies". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 415–426.

[237] Ye Xu et al. "Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns". In: *Proceedings of the 2013 International Symposium on Wearable Computers*. ACM. 2013, pp. 69–76.

[238] Zhixian Yan et al. "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach". In: *Wearable Computers (ISWC), 2012 16th International Symposium on*. Ieee. 2012, pp. 17–24.

[239] Shuochao Yao et al. "DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing". In: *arXiv preprint arXiv:1611.01942* (2016). URL: https://arxiv.org/pdf/1611.01942.pdf.

[240] Yang Ye et al. "Mining individual life pattern based on location history". In: *2009 tenth international conference on mobile data management: Systems, services and middleware*. IEEE. 2009, pp. 1–10.

[241] Yang Yu. "Towards Sample Efficient Reinforcement Learning." In: *IJCAI*. 2018, pp. 5739–5743.

[242] O Yurur and Wilfrido Moreno. "Energy efficient sensor management strategies in mobile sensing". In: *Proc. ISTEC Gener. Assem* (2011), pp. 66–72.

[243] Eftim Zdravevski et al. "Robust histogram-based feature engineering of time series data". In: *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE. 2015, pp. 381–388.

[244] Xinglin Zhang et al. "Incentives for mobile crowd sensing: A survey". In: *IEEE Communications Surveys & Tutorials* 18.1 (2015), pp. 54–67.

[245] Guanjie Zheng et al. "DRN: A deep reinforcement learning framework for news recommendation". In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 167–176.

[246] Yu Zheng et al. "Learning transportation mode from raw gps data for geographic applications on the web". In: *Proceedings of the 17th international conference on World Wide Web*. ACM. 2008, pp. 247–256.

[247] Changqing Zhou et al. "Discovering personal gazetteers: an interactive clustering approach". In: *Proceedings of the 12th annual ACM international workshop on Geographic information systems*. ACM. 2004, pp. 266–273.

[248] Zhenpeng Zhou, Xiaocheng Li, and Richard N Zare. "Optimizing chemical reactions with deep reinforcement learning". In: *ACS central science* 3.12 (2017), pp. 1337–1344.

[249] Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. "Improving energy efficiency of location sensing on smartphones". In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM. 2010, pp. 315–330.