

1-1-2015

On the Selection of Just-in-time Interventions

Luis Gabriel Jaimes

University of South Florida, ljames@mail.usf.edu

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>

 Part of the [Electrical and Computer Engineering Commons](#)

Scholar Commons Citation

Jaimes, Luis Gabriel, "On the Selection of Just-in-time Interventions" (2015). *Graduate Theses and Dissertations*.
<http://scholarcommons.usf.edu/etd/5506>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

On the Selection of Just-in-time Interventions

by

Luis G. Jaimes

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Andrew Raij, Ph.D.
Wilfrido Moreno, Ph.D.
Nasir Ghani, Ph.D.
Carlos Reyes, Ph.D.
Federico Giovannetti, M.S.

Date of Approval:
March 20, 2015

Keywords: mHealth, Affective Computing, Ubiquitous Computing, Reinforcement Learning

Copyright © 2015, Luis G. Jaimes

DEDICATION

To my mother, and my father.

ACKNOWLEDGMENTS

I would like to thank Dr. Andrew Rajj for his invaluable guidance and support during the development of this work. I would also like to thank the graduate committee members, Dr. Wilfrido Moreno, Dr. Nasir Ghani, Dr. Carlos Reyes, and MS. Federico Giovannetti, for taking the time to be a part of my committee. I would like to thank classmate and friends, Idalides Vergara, Martin Llofriu, Juan Calderon, Juan Lopez and Yueng de la Hoz, for their support and all these great experiences over the past years.

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	v
ABSTRACT	vii
CHAPTER 1 INTRODUCTION	1
1.1 Motivation	2
1.2 Problem Statement	3
1.3 A Proposed System Architecture for Stress Interventions	3
1.4 Contributions	5
1.5 Structure of the Dissertation	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 Note to the Reader	7
2.2 Background	7
2.3 Related Work and Background	9
2.4 Hardware Architecture for CPS for JITI	10
2.4.1 Network Architecture	10
2.4.2 Wearable Body Area Network (WBAN)	11
2.4.2.1 Wearable Sensors	11
2.4.2.2 Mobile Devices	11
2.4.2.3 Communication	12
2.4.2.4 Cloud Storage	12
2.5 CPS-JITIS System Architecture	12
2.5.1 Sensing and Human Condition Recognition Layer	12
2.5.2 Feedback Control System Layer	14
2.5.2.1 Modeling the How to Choose the Right Intervention	14
2.5.2.2 Modeling the Doses	15
2.5.2.3 Intervention Dosage Control System	16
2.5.2.4 Modeling the Timing to Deliver an Intervention	18
2.5.3 Theoretical Foundation for Development of Mobile Interventions	19
2.6 Other Design Issues	19
2.6.1 Privacy	19
2.6.2 Incentive Mechanism to Encourage User Engagement	19
2.7 Remarks	20

CHAPTER 3	A MODEL-BASED MECHANISM FOR INTERVENTION SELECTION	21
3.1	Note to the Reader	21
3.2	Introduction	21
3.3	Modeling the Components	21
3.3.1	Simulating the Patient Reaction to Interventions	21
3.3.2	Modeling an Episode	22
3.3.3	Modeling Rewards	22
3.4	Algorithm Framework	22
3.4.1	Learning the TPMs	23
3.4.2	The Algorithm	25
3.5	Performance Evaluation	27
3.5.1	Modeling the OSEPJITI System Components	28
3.5.2	Experiment Parameters	28
3.5.3	Experiments	29
3.6	Remarks	31
CHAPTER 4	A MODEL-BASED FREE METHOD FOR INTERVENTION SELECTION	32
4.1	Note to the Reader	32
4.2	Introduction	32
4.3	Adaptive Just-in-Time Interventions	34
4.3.1	Modeling the Intervention Dynamics	34
4.3.2	Modeling a Stress Episode	34
4.3.2.1	Time Complexity for the Intervention Selection Algorithm	35
4.3.3	Modeling the Patient Reaction to Interventions	36
4.3.4	Design Considerations	37
4.4	Relationship Among Interventions, Treatments and Q-states, a Short Example	38
4.5	Predicting and Forecasting Stress	40
4.5.1	Data	40
4.5.2	Basic Definitions	40
4.5.3	Prediction Model	42
4.5.4	Prediction Algorithm	42
4.5.4.1	Forecasting Algorithm Time Complexity	45
4.6	Remarks	45
CHAPTER 5	PERFORMANCE EVALUATION	46
5.1	Note to the Reader	46
5.2	Forecasting Experiments	46
5.2.1	Experiment Parameters	49
5.2.2	Experiment Results	49
5.3	Stress Intervention Experiments	51
5.3.1	Simulation Settings	51
5.3.2	QL vs Random Policy	52
5.3.3	Experiment Parameters	52
5.3.4	Experiment Results and Discussion	53
5.3.5	QL vs CALMA	54
5.3.6	Experiment Parameter	56

5.3.7	Experiment Results and Discussion	56
5.4	Remarks	56
CHAPTER 6	CONCLUSIONS	58
6.1	Summary of Results and Findings	58
REFERENCES		60
APPENDICES		67
Appendix A	Permission for Reuse	68

LIST OF TABLES

Table 2.1	Common terms and abbreviations used in this dissertation.	9
Table 3.1	<i>OSEPJITI</i> parameter search data.	29
Table 3.2	Beta distribution parameters for no stress probability sampling.	29
Table 4.1	Beta distribution parameters for no stress probability sampling.	36
Table 4.2	Demographics of participants in the two studies [1].	41
Table 5.1	Evaluation metrics used in this dissertation.	49
Table 5.2	Forecasting initial parameters common to all experiments.	50
Table 5.3	Forecasting experiment initial specific parameters.	50
Table 5.4	Forecasting experiment results.	50
Table 5.5	QL parameter search data.	53

LIST OF FIGURES

Figure 1.1	A three layer architecture for stress just-in-time interventions.	4
Figure 1.2	Flow of the top-down stress three layer architecture.	4
Figure 2.1	Mobile interventions vs non-intervention.	8
Figure 2.2	Network topology.	11
Figure 2.3	Sensing layer process flow.	13
Figure 2.4	CPS with a human in the loop.	14
Figure 2.5	Patient condition representation.	16
Figure 2.6	Effect of disturbances and interventions on the patient condition.	17
Figure 2.7	Control feed-back system.	18
Figure 2.8	Feedback block diagram.	18
Figure 3.1	Random policy Vs OSEPJITI, and OSEPJITI-update $TrainingSize = 40$.	30
Figure 3.2	Average of 100 runs, random policy Vs, OSEPJITI, OSEPJITI-update.	30
Figure 4.1	Hardware architecture of a context-aware mobile system.	33
Figure 4.2	Beta distribution for probability sampling.	37
Figure 4.3	Graphical models of the patient's stress.	38
Figure 4.4	Q-states transition graph.	39
Figure 4.5	HMM graphical model, $k = T + h$, and h is the forecasting horizon.	41
Figure 5.1	One step-ahead forecasting using a four hidden states Poisson HMM.	47
Figure 5.2	Two step-ahead forecasting using a four hidden states Poisson HMM.	47
Figure 5.3	Three step-ahead forecasting using a four hidden states Poisson HMM.	47
Figure 5.4	Capturing the data trend.	48

Figure 5.5	Average number of intervention to relieve stress.	54
Figure 5.6	Cumulative average of the number of interventions per episode to relieve stress.	55
Figure 5.7	Average number of intervention to relieve stress, variability per episode, 100 repetitions.	55
Figure 5.8	Random policy Vs QL Vs value iteration with a <i>TrainingSize</i> =20.	56
Figure 5.9	Cumulative average random policy Vs QL Vs value iteration with a <i>TrainingSize</i> =20.	57
Figure 5.10	Cumulative average random policy Vs QL Vs value iteration with a <i>TrainingSize</i> =40.	57

ABSTRACT

A deeper understanding of human physiology, combined with improvements in sensing technologies, is fulfilling the vision of affective computing, where applications monitor and react to changes in affect. Further, the proliferation of commodity mobile devices is extending these applications into the natural environment, where they become a pervasive part of our daily lives. This work examines one such pervasive affective computing application with significant implications for long-term health and quality of life adaptive just-in-time interventions (AJITIs). We discuss fundamental components needed to design AJITIs based for one kind of affective data, namely stress. Chronic stress has significant long-term behavioral and physical health consequences, including an increased risk of cardiovascular disease, cancer, anxiety and depression. This dissertation presents the state-of-the-art of Just-in-time interventions for stress. It includes a new architecture that is used to describe the most important issues in the design, implementation, and evaluation of AJITIs. Then, the most important mechanisms available in the literature are described, and classified. The dissertation also presents a simulation model to study and evaluate different strategies and algorithms for interventions selection. Then, a new hybrid mechanism based on value iteration and monte carlo simulation method is proposed. This semi-online algorithm dynamically builds a transition probability matrix (TPM) which is used to obtain a new policy for intervention selection. We present this algorithm in two different versions. The first version uses a pre-determined number of stress episodes as a training set to create a TPM, and then to generate the policy that will be used to select interventions in the future. In the second version, we use each new stress episode to update the TPM, and a pre-determined number of episodes to update our selection policy for interventions. We also present a completely online learning algorithm for intervention selection based on Q-learning with eligibility traces. We show that this algorithm could be used by an affective computing system to select and deliver in mobile environments. Finally, we conduct post-hoc experiments and simulations to demonstrate feasibility of both real-time stress forecasting and stress intervention adaptation and optimization.

CHAPTER 1

INTRODUCTION

Advancements in pervasive computing are rapidly changing preventative healthcare. Under the status quo, the average healthy individual visits the doctor rarely, perhaps just once a year. The doctor assesses the patient and then may prescribe medications and recommend behavior changes (reduce fat consumption, exercise more, etc.). One year later, the patient returns and this process is repeated. In the emerging new model of health care, the patient carries sensors that monitor health in real-time, as the patient goes about normal daily life [2]. A smartphone and cloud-based services assess monitored data at a much higher frequency (on the order of minutes or seconds, if needed), allowing health interventions to be prescribed and delivered more frequently and in the natural environment. This vision of intervention in the natural environment is sometimes called Ecological Momentary Intervention (EMI) [2].

An important variation on EMI are adaptive just-in-time interventions (AJITIs) [3]. AJITIs leverage real-time and historical information about the user to maximize the success of the intervention. The JITI component of AJITI refers to the idea that the intervention is delivered precisely when needed (neither too early nor too late; just in time). The A component of AJITIs refers to algorithms that aim to maximize intervention success via real-time adaption of interventions to the user and their context. AJITIs search the space of intervention parameters (i.e., intervention type, timing, and dose) to select the intervention that would best address the user's health at the current moment in time. Murphy and Chakraborty examined the use of reinforcement learning [4] for this optimization. Rivera [5] models the problem using dynamic systems and control theory. These optimization approaches have been used as a treatment mechanism for variety of disorders such as: Smoking cessation, weight loss, anxiety reduction, and eating disorder reduction [6].

In this dissertation, we examine AJITIs for chronic stress. Stress is a "silent killer," in that the negative impacts of stress on the body are not instantaneously noticeable. Rather, the effects of stress accumulate

over time and lead to significant wear and tear on the cardiovascular system. A well-designed AJITI system could reduce this accumulation of negative effects by helping individuals reduce stress level on daily basis.

Building on previous work on the timing constraints imposed by AJITI [7], we identify a need for dedicated intervention management as well as the ability to forecast health state in advance. The latter enables intervention before a sudden increase in stress. We then propose a three layer architecture for AJITIs: a continuous sensing layer, a real time stress recognition and forecasting layer, and an adaptive intervention management module. Several mHealth systems include continuous sensing and health state recognition [8,9]. However, to our knowledge, forecasting and real-time adaptive intervention components have not yet been investigated in the literature. We investigate these two new components of the architecture more closely with post-hoc simulations on real-world data.

1.1 Motivation

Several factors are driving a slow but steady shift in medicine. Today, doctors, hospitals, and clinics are the center of healthcare. Optimistically, people visit the doctor maybe once per year, where they receive expert advice, nudges (you really need to reduce your fat intake), treatments, and action plans, all within a short 10-15 minute meeting with the doctor. Afterward, the patient goes back to his/her normal daily life and the doctor moves on to the next patient.

Inexpensive wearable sensors, always-on internet connections, and computationally powerful smartphones are changing this model. Thanks to these technologies, the practice, distribution and delivery of healthcare is becoming democratized, patient-driven, personalized, inexpensive, timely, and - perhaps most important - preventative. The greatest potential of these new technologies is in preventing health problems long before they happen. Imagine a smartphone that knows when a patient is craving a cigarette, and then intervenes in some way to convince and prevent the user from smoking a cigarette. If we apply this vision to other health problems, we may be able to short-circuit almost an entire class of common healthcare challenges, including obesity, diabetes, and some cancers. While the technology components exist to enable this vision, several core challenges remain to make robust, mobile, cyber-physical health systems achieve it.

1.2 Problem Statement

According to Murphy [4] AJITIs follow the same rules of any medical treatment. Additionally, it should be possible to administrate these treatments beyond medical facilities, and at any time, hopefully without interrupting the patient daily routines.

As any medical treatment, AJITIs include factors such as: Timing for administration or frequency, doses or amount, and choosing the appropriate medication given the patient symptoms. Of course, instead of considering the patient's symptoms in AJITIs we are concerned about the user temporal context.

Thereby, the last factor can be re-stated as how to choose the intervention or set of them that minimize the average number steps needed to relieve a patient from stress given his/her temporal context.

In this dissertation we address this last factor, and postpone the timing and doses for future work. Thereby, we make an online selection of an effective personalized intervention given either the current user temporal context or a predicted user stress state. In the first case, allowing us to deliver a stress mitigation intervention, and in the second case a preventive one.

We address the first challenge by the use of reinforcement learning and propose a solution in the form of Q-learning algorithm with eligibility traces. Besides, we show the feasibility of our proposal by a set of simulations.

In order to address the second challenge, we use the framework constructed to address challenge one, plus a mechanism to stress forecasting based on Hidden Markov Models. We explain our proposed solution and again we show the feasibility of our solution, this time working with real data.

1.3 A Proposed System Architecture for Stress Interventions

We propose a three layer architecture for JITAIIs that includes the following layers: Sensing and inference, prediction or forecasting, and online intervention selection. Figure 1.1 depicts the proposed architecture.

As in other context-aware systems, the first layer is where sensing of user behavior and health state occurs. It consists of sensing hardware and software to pass sensed samples on to the next layer. For

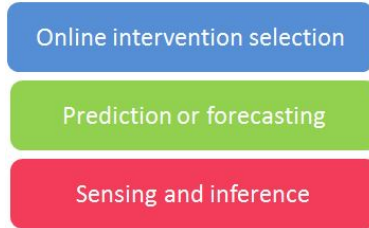


Figure 1.1: A three layer architecture for stress just-in-time interventions.

example, in the experimental work described in later sections, the sensing layer consists of a two-lead ECG and a Bluetooth connection to transmit captured samples to a smart phone.

The second layer has two components, a real-time recognizer and a real-time forecaster. The recognizer processes samples from the sensing layer to determine the health state of the user. The forecaster uses the current user's health state (as determined by the recognizer) and other contextual information to predict what the user's health state will be in the near future. Thus the forecaster has a temporal dependency on the recognizer. First, the user's health state at time t is determined. Then, the forecaster can predict what this health state will be at time $t + \delta$. This information is then passed on to the intervention layer, where the decision to select a particular intervention is made.

For example, in the experimental work described in later sections, raw ECG data is passed from the continuous sensing layer to the stress recognition layer. There, the ECG data is processed into heart rate variability (HRV), a proxy for continuous stress. At the same time, HRV data is passed to a stress forecaster. For every minute t , the forecaster produces a prediction of stress level at $t + 1$, $t + 2$, and $t + 3$ minutes in the future.

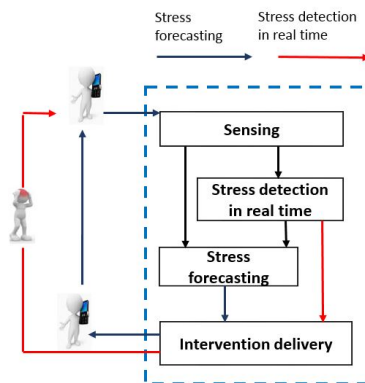


Figure 1.2: Flow of the top-down stress three layer architecture.

Figure 1.2 outlines the relationship between the inputs and output of the different layers, emphasizing the advantages of using predicted stress values as input into intervention layer. Here, black lines represent the flow and connection between the architecture layers when forecasted values are used. The goal of anticipating an stressful event is to prevent the user from going through a traumatic experience.

Here, the first layer and the inference module in the second layer have the form of AutoSense [10], a wearable sensor suit for stress recognition in real time that has been widely documented and tested in several studies [11]. The details about these two components are out of the scope of this dissertation, however, the second and third layer as well as the interaction between them are widely documented in Chapter 3, Chapter 4, and Chapter 5, respectively.

1.4 Contributions

The main contributions of this dissertation are presented next. Each one of these contributions will be elaborated in depth in subsequent chapters.

- *A taxonomy for the classification of the Just-in-time intervention systems.* The first contribution of this dissertation is the presentation of the state-of-the-art in Just-in-time interventions systems (JITI). After a general description of JITI systems and its main components, this dissertation defines the most important issues to consider in the design, implementation, and evaluation of just-in-time intervention mechanisms. Then, a three-level architecture is presented for classifying the mechanisms [12].
- *A simulation framework.* The second contribution of this dissertation is a simulation framework to study and evaluate different strategies and algorithms for interventions selection. This modular framework is based on an episodic modeling approach, and provides elements to simulate patient reaction to intervention as well as a set of rules for the application of of intervention [13].
- *A model-based mechanism for intervention selection based on Monte carlo simulation and value iteration.* The fourth contribution of this dissertation is the introduction of a semi-online model based on Monte carlo simulation and value iteration for intervention selection. This method builds a transition probability matrix (TPM) using a pre-determined number of episodes as a training set. This

TPM can be used to create the policy that will be used during the whole process, or it can be updated periodically in order to capture the changes in the user behavior over the time [13].

A model-based free mechanism for intervention selection based on Q-learning with eligibility traces.

The fifth contribution of this dissertation corresponds to a method for intervention selection based on Q-learning with eligibility traces. Unlike the value iteration methods, Q-learning is an online method with a time complexity that makes it suitable for mobile environments [14].

- *A forecasting algorithm for psychological stress.* The sixth contribution of this work corresponds to a mechanism for stress forecasting based on Hidden Markov Models (HMM). This mechanism uses the posterior predictive distribution density to make predictions with one, two, and three minutes in ahead. We use Heart Rate Variability (HRV) a common proxy for stress as the input of our algorithm [14,15].

1.5 Structure of the Dissertation

The rest of the dissertation is structured as follows. Chapter 2 contains the related work, presenting the three-level taxonomy of just-in-time interventions along with a qualitative evaluation of them. Chapter 3 presents a model-based method for intervention selection based on value iteration. Chapter 4 presents a model free method for intervention selection based on Q-learning. Chapter 5 the performance evaluation. Finally, Chapter 6 concludes the dissertation and presents possible areas of future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Note to the Reader

Part of this chapter was published in the IEEE-Internet of Things Journal [16] and the proceedings of following conferences: IEEE-PerCom 2012 [17], IEEE-PerCom CrowdSensing WS 2014 [18], IEEE-LatinCom 2014 [19], and IEEE-SoutheastCon 2015 [12](to appear) .

2.2 Background

Thanks to the advent of mobile technologies and sensor miniaturization, it is possible for us to track a patient's health in real time [2]. Physiological data, such as skin temperature (ST), heart rate variability (HRV), and respiration rate (RR), are usually collected through a set of sensors attached to the body of the patient. Another set of less invasive sensors, such as accelerometers, gyroscopes, proximity sensors, GPS, and microphones (which can be found in almost every smart phone), are often used to infer a user's physical activity as well as to provide some clues about the context of the user. Finally, a set of virtual sensors, such as social sensors (which collect data from the user's agenda) and incoming or outgoing calls, can be used in combination with the array of sensors described previously to infer social behaviors.

Data streams coming from these sensors are used as the input of inference models. These inference models usually have the form of machine learning algorithms, which are trained using ground true data to recognize the condition of a patient [20]. For instance, by using respiration rate and accelerometer data, it is possible to infer chest expansion. Similarly, by monitoring arm movement, we can infer smoking behaviors [21]. Lastly, a stress model could use the arousal of physiological signals such as HRV, RR, and ST to infer stress [8].

JITI systems use these context-aware systems to recognize human conditions or behaviors (e.g., stress, smoking) as a first layer of their architectures. On top of that layer is the recommender layer, where the system takes as input continuous data streams coming from the first layer. Data from the first layer might include inferences about the patients conditions, such as the likelihood of the user to be under stress, the users approximate location, and the social context where the user might be.

Based on the information provided by the first layer, a recommender system in the second layer takes the decision of whether or not to intervene. Additionally, if the decision is to intervene, the system has to decide in an autonomous way when to intervene (i.e., the timing), the intervention doses, and chooses the intervention that maximizes the effectiveness of the treatment.

The whole process described above is operated through a Cyber-Physical System (CPS). A CPS integrates sensor perception, communication networks, and control feedback systems to control physical entities [22]. In this work, we analyze the characteristics and potential of CPS with human-in-the-loop. In this dissertation, the element to control has the form of human behaviors (e.g., eating disorders, alcohol and smoking addictions, etc) or human mental states (e.g, psychological stress). In this dissertation, we will often use the control of psychological stress as a reference example. We selected this challenging task because it will help us show the most recent advances in the design of MCPS-JITI systems. Figure 2.1 compares the advantage of using MCPS-JITI to deliver an intervention for stress. In the picture, after identifying the right time to interrupt the patient, the system delivers a mobile intervention, which helps the patient to cope with a stressful situation and continue with his or her daily activities.

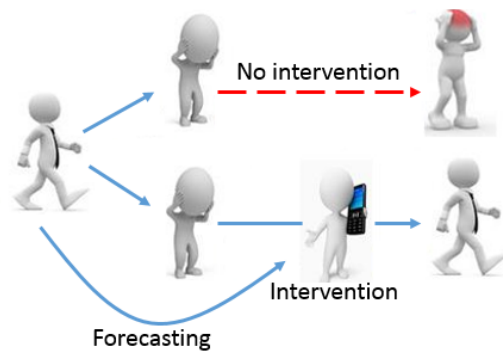


Figure 2.1: Mobile interventions vs non-intervention.

Table 2.1: Common terms and abbreviations used in this dissertation.

Abbreviation	Description
Users	Participant of an intervention system
Participant	Participant of an intervention system
CPS	Cyber-physical Systems
MCPS	Mobile CPS
JITI	Just-in-time intervention
AJITI	Adaptive JITI
MCPS-JITI	System that combine MCPS and JITI
MDP	Markov decision process
MABP	Multi-armed bandit problem
ST	Skin Temperature
RIP	respiratory inductive plethysmograph
ECG	Electrocardiogram
EEG	Electroencephalogram
GSR	Galvanic skin response
GPS	Global positioning system
SVM	Support vector machines
HRV	Heart rate variability
RR	Respiration rate
EMA	Ecological momentary assessment
EMI	Ecological momentary intervention
DTRAI	Dynamic treatment regimes and adaptive interventions
WSN-MN	Wireless sensor network with mobile nodes
BAN	Body area networks

2.3 Related Work and Background

Delivery of supportive therapies and treatments often takes place in medical facilities such as hospitals and rehabilitation institutions. The disadvantage of this traditional approach is the limited access that the average population has to public or private health systems. In addition, in some cases, the capacity of health institutions is not enough to follow up with treatments for the entire patient population. Therefore, institutions often have to prioritize based on the severity of the conditions and the availability of medical personal and budget.

A first step to address this problem was the development of what is called *Ecological Momentary Assessment* (EMA) [23]. EMA was developed by the social behavioral community to gather user's feedback in real time. This feedback often has the form of descriptions about feelings or circumstances that took place when a patient experienced a crisis(e.g., smoking cravings, stress). This feedback, which gathers a natural environment and records in real time, is used later for treatment support. The logic about the use of EMA is that patients in follow up sessions usually forget the details about the exact circumstances about what happened when the crisis event took place.

A second milestone toward treatment support in natural environments was the development of Ecological Momentary Intervention (EMI) [2]. The goal of EMI is to complement and reinforce the treatments that take place in the health institutions by the use of interventions in natural environments. EMI-EMA is used to gather the user's feedback (i.e., written descriptions, non-sensor data), whereas EMI is used to provide treatment in semi-real time. This support has the form a phone call or a text message sent by the physician or caregiver. This combination has been used as a treatment mechanism for a variety of disorders such as: Smoking cessation [21], weight loss [24], anxiety reduction [25] as well as eating disorder reduction [6]. In all the cases, the intervention works in an asynchronous way (i.e., they are not just-in-time interventions), they respond when the user requires help, and not by the activation of an automatic sensor-based system.

A recent research field, which deals with the creation of strategies for interventions and the evaluation of their effectiveness, is called Dynamic Treatment Regimes and adaptive interventions (DTRAI) [3]. The main components of DTRAI are factors such as the decision about whether or not to intervene, the timing, and the intensity that maximizes the intervention's effectiveness. An appealing characteristic of this approach is its adaptive component, which takes into account the differences among patients. That means that treatments (i.e., sequence of interventions in a period of time) are tailored according to the characteristics of each patient. Research projects in this field include the work of Murphy and Chakraborty [3, 4, 26] in reinforcement learning as well as that of Rivera [5], who modeled the problem using dynamic systems and control theory.

2.4 Hardware Architecture for CPS for JITI

2.4.1 Network Architecture

Figure 2.2 shows the network model for a CPS-JITI, which corresponds to a Wireless Sensor Network with Mobile Nodes (WSN-MN). In this figure, each mobile node represents a Body Area Network (BAN) that collects physiological sensor data from a patient (e.g. ECG) and sends these data or inferences made on these data using a JITI server.

The WSN-MN operates in discrete time with a unit time slot $t \in \{1, \dots, n\}$, and at each time slot t the mobile nodes can send sensor data to the JITI server by switching among any transmission method (e.g., WiFi direct, Bluetooth, and LTE direct) in order to minimize transmission costs.

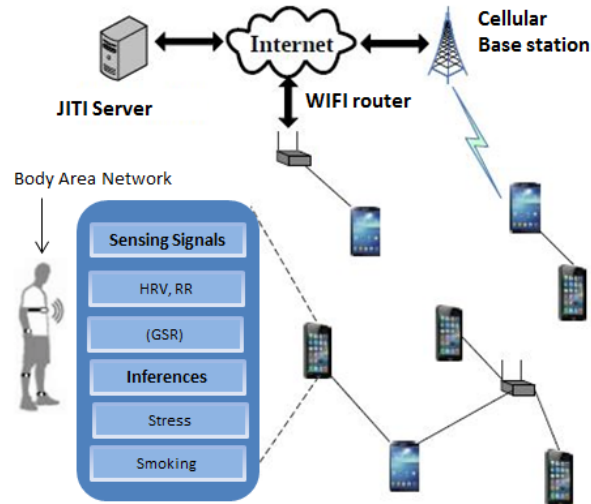


Figure 2.2: Network topology.

2.4.2 Wearable Body Area Network (WBAN)

2.4.2.1 Wearable Sensors

The task of these small and unobtrusive devices is to continuously measure a user's physiological signals and any other user-contextual information. They are often located in chest bands, waist bands, and embedded in smart devices. The collected data is usually transmitted via Wireless Personal Area Network (WPAN) to a mobile device for aggregation and processing.

2.4.2.2 Mobile Devices

The role of these smart devices is to receive and consolidate data from a variety of wearable sensors. Additionally, the smart device may offer its own sensor data, such as acceleration, video, or speech. The whole task of processing, inference (i.e., stress recognition), and user feedback (i.e., intervention management and delivery) can take place in the smart device (i.e., compact model). Alternatively, the mobile device can be used as a bridge where data from the different sensors is consolidated and then transmitted to the cloud (i.e., extended model).

2.4.2.3 Communication

Here, communication protocol depends on the systems hardware architecture model, the system may use two different models. In the case of the compact model, the system can use a WPAN, which is usually based on Bluetooth communication protocols.

2.4.2.4 Cloud Storage

This component only plays a role if we use an extended architectural model. The cloud transmits the intervention data to the user's mobile device, which in turn displays the intervention to the user via smart device (e.g., smart phone, smart glasses).

2.5 CPS-JITIS System Architecture

To describe the main components of CPS-JITIS and their inner connections, we use a four layer architectural approach. For CPS-JITIS, the first layer includes elements such as sensing, signal processing, and feature extraction. The second layer comprises the recognizer. This layer uses the features computed in the first layer along with specific models to infer a human condition (e.g., stress, smoking). Often, these models take the form of a machine learning algorithm trained and tested using ground true data. Layer three is based on the feedback system, which takes as input the output of the recognizer. Layer three is where the system decides when to intervene, intervention dosage, selection, and delivery. Finally, the fourth layer corresponds to the set of interventions. Because the development of these therapeutic elements is usually assigned to behavioral scientists and psychologists, it is out of the scope of this work. In this dissertation, we briefly describe some psychological theories that motivate the JITIs.

2.5.1 Sensing and Human Condition Recognition Layer

The CPS-JITIS sensing layer is the architecture's physical layer. The goal of this component is to continuously sense the user's vital signals in order to obtain signals at sampling rates that are adequate for data analysis and inference. These signals are sent either to the mobile phone (e.g., via Bluetooth) or to the

JITI server (e.g, via TCP/IP), where processing and feature extraction take place. Figure 2.3 shows the flow chart of the process [27].

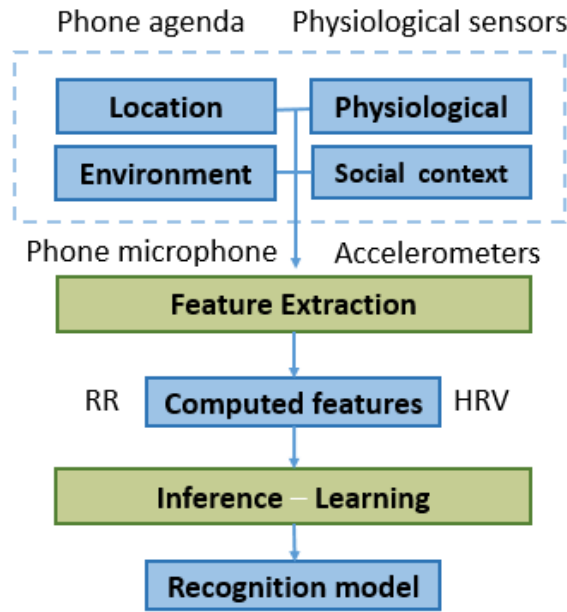


Figure 2.3: Sensing layer process flow.

Commonly used sensors for detecting human condition such as stress or smoking behaviors include: Electrocardiogram (ECG) sensors, which consist of electrodes that measure the electrical pulses of the heart; respiratory inductive plethysmograph (RIP) sensors, which measure relative lung volume and breathing rate; electroencephalogram (EEG) sensors, which measure the electrical signals produced by neural activity in the brain; and also galvanic skin response (GSR) sensors [28] and skin temperature thermistors. Sharma and Gedeon [27] survey these sensors and metrics used to recognize stress.

Another set of sensors often embedded in the mobile phone include: Global Positioning System (GPS), tri-axis accelerometers, gyroscopes, microphones, light sensors, humidity sensors, proximity sensors, and thermometers. This set of sensors is used to infer activities based on location, environmental conditions, and social context. A set of systems that combines the previous set of sensors to infer stress includes [8,9,29,30].

Finally, a set of models use as input a combination of physiological features, location, and social context to infer physical activities [31] and mental states and emotions [20]. Examples include, smoking behaviors [32], and psychological stress models [20]. These statistical models are often created using machine

learning techniques such as decision trees, support vector machines (SVM), bayesian classifiers, clustering [33, 34], and hidden markov models [27].

2.5.2 Feedback Control System Layer

This is one of the most important layers, and the focus of this dissertation. This layer takes as input the output of the recognition layer (e.g., stress, non-stress) at discrete time slots $t \in \{1, \dots, n\}$. Depending on the user's state at time t , a control feedback system decides whether or not to intervene. The output of this layer is a policy or set of rules p that dictates the time, dosage, and the intervention to be delivered, given user's state at time t .

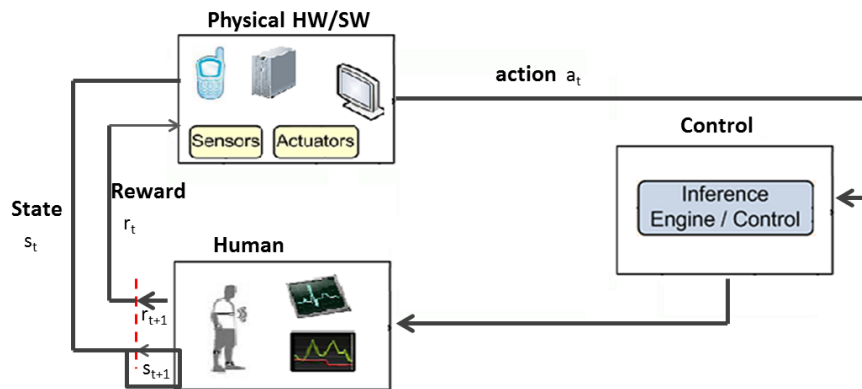


Figure 2.4: CPS with a human in the loop.

2.5.2.1 Modeling the How to Choose the Right Intervention

The problem of selecting an intervention a , given the user's state at time t , is often modeled as a multi-arm bandit problem (MABP) [35, 36]. Weber *et al.* [37] describes MABP using a metaphor in which a gambler has the opportunity to play any of n slot machines (known as one-armed bandits). The gambler has to decide which machines to play, how many times to play each machine, and the order in which to play the machines in order to maximize his or her total discounted reward. The main objective of this modeling approach is to select the intervention (i.e., the slot machine arm) and then maximize the average reward in the long term. Thus, the system delivers this one intervention at a specific time on the basis of its experience and observations to date.

The MABP is formally described in terms of Markov Decision Process (MDP) [38]. In this modeling framework (i.e., MDP), the user is referred to as the environment, and it is modeled as a random variable that can switch between one or several states (e.g., stress, non-stress). These states are also influenced by a control signal (i.e., intervention) provided by an agent or observer. The system or observer is continuously sensing the environment reaction to the control signal with the objective of stabilizing the system.

An MDP is characterized by four main elements: A set of states S , a set of actions A , and a set of rewards R . Thereby, we are interested in $P_a(s, s')$ (see Equation 3.1), the probability to reach the state s' at time $t + 1$, given that action a was taken at time t .

$$P_a(s, s') = Pr(s_{s+1} = s' | s_t = s, a_t = a) \quad (2.1)$$

In addition, we observe $R_a(s, s')$, the immediate reward received after switching from state s to state s' to evaluate the effectiveness of applying intervention a .

There are two well defined methods to solve an MDP: The model-free, and model-based approaches. The model-free method is utilized when the effect of applying an intervention on the user's state is unknown. This is the case when we have to learn a policy by using experience. Common approaches include the use of reinforcement learning algorithms such Q-learning and monte carlo methods [39]. These methods take advantage from the experience and do not need $P_a(s, s')$. Recommendation systems based on these techniques include PopTherapy [40], MoodWings [41], and Calma [14].

On the other hand, when we know in advance the intervention effect on the user's states, the common solution is to use dynamic programming techniques such as policy or value iteration [39]. Systems following this model include LED [42] and work of Yasavur [43] and Reza [44] on Alcohol Interventions.

2.5.2.2 Modeling the Doses

Control theory (CT) offers an alternative modeling framework for the design and implementation of behavioral health interventions. In this framework, it is common to represent a patient condition with a fluid contained in a tank or reservoir. Figure 2.5 shows how a patient's condition $PF(t)$ is dependent on $I(t)$, and $D(t)$, especially from the set of interventions and disturbances respectively.

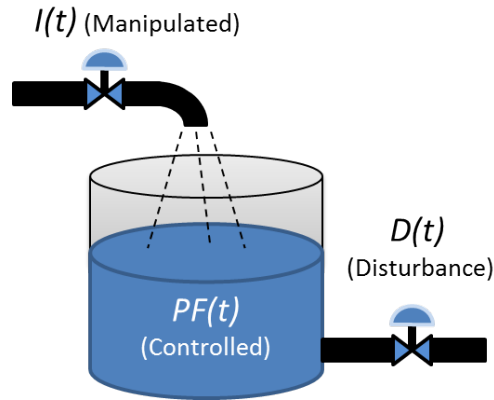


Figure 2.5: Patient condition representation.

This model associates the idea of maintaining an adequate fluid level in the tank with a user’s health condition and wellbeing. Equation 2.2 shows that a patient’s condition after a time interval T is dependent on factors such as the patient’s current condition $PF(t)$; the sum of the applied intervention at the time interval T times a constant k_l ; and the sum of all disturbances or problems that reduce the fluid level at the same time interval (see Equation 2.3).

$$PF(t + T) = pf(t) + K_l l(t) - D(t) \quad (2.2)$$

$$D(t) = \sum_{i=1}^n i^2 4 \quad (2.3)$$

Figure 2.6 shows the effect of disturbances $D(t)$ (i.e. layoffs and illnesses) and interventions $I(t)$ on the PF level. Notice how the control system stabilizes the patient’s condition (solid blue curb) by using interventions. In the figure, the red line represents a healthy patient or a patient under normal conditions.

2.5.2.3 Intervention Dosage Control System

Once the model representation (e.g., tank model) of the variable of interest(e.g., stress) is chosen, we can proceed with the design of the control system. Figure 2.7 shows a block diagram for a simple feedback control system.

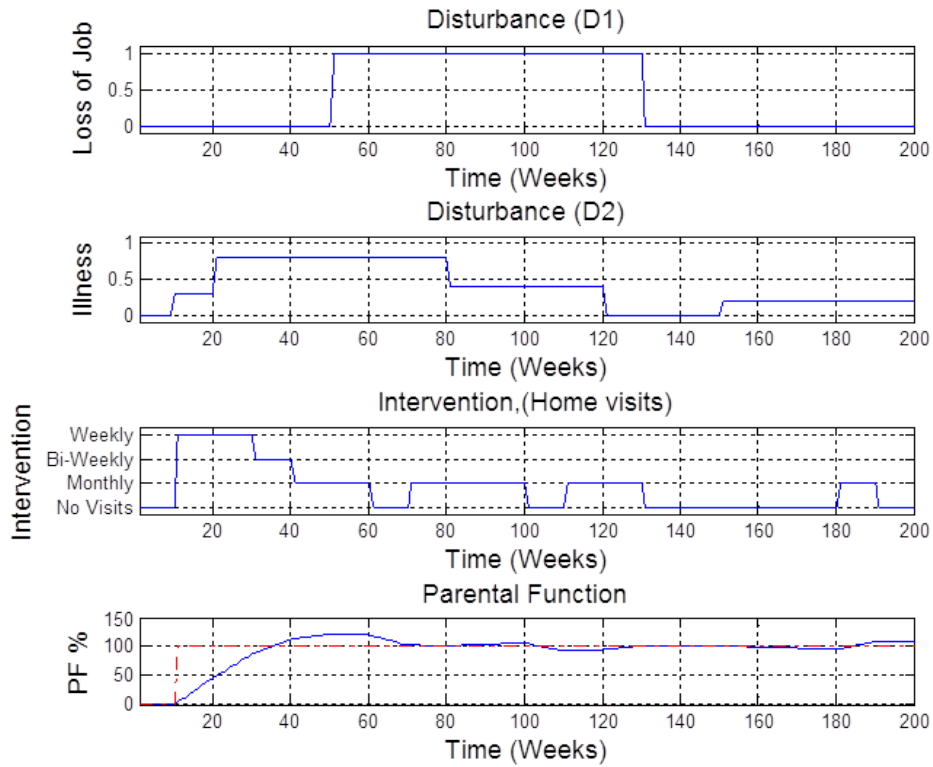


Figure 2.6: Effect of disturbances and interventions on the patient condition.

Figure 2.8 shows the previous block diagram, illustrated now as a control model based on the tank representation. This controller [45] applies a set of decision rules to determine the dosage of the intervention. In the figure, sensor LT continuously senses the tank's fluid level and notifies the controller the time to open or close the flow of interventions. Notice that the disturbances D are independent from the set of other variables of the system, and the role of the control mechanism is to compensate for the negative effects of D .

Other techniques for designing advanced control systems such as predictive, adaptive, and optimal control have been used in the design of intervention systems for HIV and smoking cessation [45–47]. Finally, Doboecq and Bergerman [48, 49] use the tank reservoir model to represent the negative impact of stress in older populations.

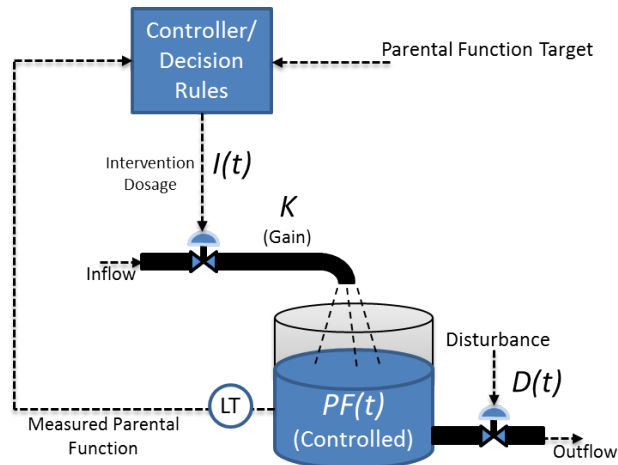


Figure 2.7: Control feed-back system.

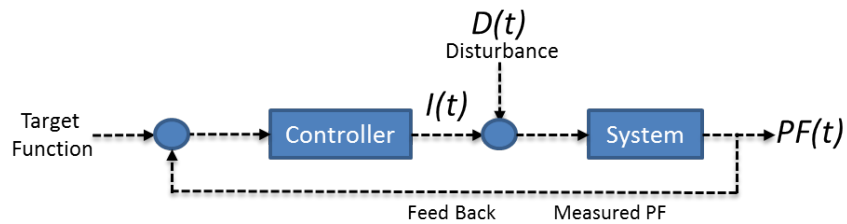


Figure 2.8: Feedback block diagram.

2.5.2.4 Modeling the Timing to Deliver an Intervention

Mobile phones have changed the way to obtain information. By using search engines such as Google or Yahoo, users can access information located anywhere at anytime. However, as Pejovic and Musolesi [50] emphasize, *accessibility does not necessarily imply reachability*. Studies such as [51, 52] reveal that interruptions at wrong times might increase the user’s stress, reduce a worker’s performance, and increase chances of error. In the mobile environment interruptions have the form of notifications through text messages, emails, and calendar reminders that often pop up in the phone interface.

Recent works in anticipatory mobile computing [53–55] show that user’s contextual information such as location, physical and social context, activities, and emotions determine the user’s interruptivity. Hence, the challenge is to develop intelligent applications that leverage the user’s contextual information to deliver notifications without causing unpleasant interruptions to recipients.

Some projects in mobile notification and interruptivity include the work of Pejovic [50] and Sarker *et al.* [55]. In the first case, the authors present, InterruptMe an interruption management library for android smart phones. The library uses a set of interruptivity models that have been trained and tested using post-mortem analysis of interruption traces. In the second case, the authors present an exploratory research in the context notification delivery for just-in-time interventions (JITI). This work explores the factors that influence the delay or quick response to JITI notifications. In addition, the authors propose an approach to determine a user's availability to engage in JITI's tasks that include the use of micro-incentives.

2.5.3 Theoretical Foundation for Development of Mobile Interventions

We briefly describe some of the psychological theories that motivate the development of mobile interventions, particularly for stress reduction. The work of [56] is a system inspired by the Proteus Effect, which states that manipulating the avatar's appearance and behavior affects the user's behavior. Other projects in this direction include the work of Murray and Jaimes [56,57]. Other psychological theories such as positive psychology [58], and Cognitive Behavioral Theory [59] have been used in [40] and [60].

2.6 Other Design Issues

2.6.1 Privacy

An important design issue is to address the privacy concerns of patients regarding the disclosure of continuously-collected physiological, behavioral, and psychological data. In fact, Raij *et al.* [61] found that people do not understand the potential privacy threats associated with having their data recorded. In another study Taylor *et al.* [62] explore the vulnerability of pervasive health systems in terms of data sensitivity, especially when multiple devices communicate and share the data. Finally, Jaimes *et al.* [63] explore the role of trust in personal informatics systems.

2.6.2 Incentive Mechanism to Encourage User Engagement

A key factor for the success of any strategy that involves the use of Mobile interventions is user engagement. Health interventions tend to be activities that aim to help patients to cope with a harmful situation

(e.g., stressful event). However, if the patient does not accomplish the task, the intervention will not have any effects. In a recent study that involved 30 participants in a week-long field study, Rajj *et al.* [64] explored the effects of micro-incentives on participants in crowd sourcing schemes. Sarker *et al.* [55] found that micro-incentives and compensations affect the interest of participants studies that aimed to observe Just-in-time interventions. Other studies include the work of Jaimes *et al.* [17–19, 65] in the context of crowd sensing incentive techniques.

2.7 Remarks

To the best of our understanding, this is the first work that surveys the state of the art of design techniques for construction of MCPS-JITI systems. Our work includes the following contributions: First, we propose a set of design issues which can be used as metrics to evaluate MCPS-JITI systems. Then, we propose a three-layer architecture for MCPS-JITI systems and explain the engineering concepts behind the different design alternatives. Third, we survey current MCPS-JITI systems based on the similarities between each system and each layer of our proposed architecture. Finally, we propose a set recommendations and guidelines for future research. Therefore, this dissertation becomes an important tool for the selection of appropriate techniques in the design of Mobile Cyber-Physical Systems for Health interventions.

CHAPTER 3

A MODEL-BASED MECHANISM FOR INTERVENTION SELECTION

3.1 Note to the Reader

Part of this chapter was already accepted for publication and will appear in the proceedings of IEEE-SoutheastCon 2015 [13].

3.2 Introduction

In this chapter we propose an algorithm framework based on Value Iteration and Monte Carlo Methods to choose the intervention or set of interventions that minimize the average number of steps needed to relieve a patient from a health condition. Our algorithm is designed to work with data streams as those produced by applications such as AutoSense, where every t seconds a new fresh data-inference (e.g., stress, non-stress) shows up. In this context, our system decides what the best treatment to administer is, given a patient's state.

Value iteration is a well-known model-based method used to solve MDPs when a Transition Probability Matrix (TPM) is available, especially when we know the effect of the interventions on the patients. A key contribution of this dissertation is the introduction of a new algorithm based on Monte Carlo (MC) methods, which allows to learn a TPM from a few episodes.

3.3 Modeling the Components

3.3.1 Simulating the Patient Reaction to Interventions

We simulate a patient's reaction to an intervention by creating a distribution $p(s|t_1, \dots, t_N)$ that corresponds to the probability of not having stress under a set of treatments t_i . We sample from a beta distribution with the parameters shown in Table 4.1, where most of the probability mass is concentrated near zero. Thus,

most of the treatment combinations will be inefficient, while a few of them will have a greater chance of relieving the patient. The rationale behind using this distribution, is to show the ability of our framework to learn the most effective treatments from just few successful examples.

3.3.2 Modeling an Episode

There are two main approaches to model a patient’s condition in the reinforcement learning framework, namely episodic and continuous [39]. In the former, a patient’s condition (e.g., patient stress) naturally breaks down into a sequence of separate episodes. In the latter, a patient is seen as a subject of continuous study that changes from one state to another from the time of his/her birth until the time of his/her death.

In this work we model a patient’s condition such as stress in terms of episodes. We consider an episode as the duration of time between the moment when stress is detected and the moment when the patient shows no more sign of stress, or when there are no more interventions to deliver.

We state the following design constraints: 1) Just one intervention of each type can be delivered in each episode and 2) the order of the delivery does not affect the outcome.

3.3.3 Modeling Rewards

A positive reward $r_{relieve}$ is given whenever the system succeeds in treating the patient; a zero reward $r_{intervention}$ is given after an intervention ends in a non-terminal or no-absorbing state; and a negative reward is given when the episode ends in a terminal or absorbing state.

Thereby, the problem is then to find a policy $\pi : S \times A \rightarrow A$ that maximizes the expected reward, which is equivalent to relieving the patient’s stress while minimizing the number of interventions. Here, A corresponds to the number of available interventions and S to the number of q-states.

3.4 Algorithm Framework

This proposed framework includes four algorithms: Online Selection of an Effective Personalized Just-in-Time Intervention (*OSEPJITI*), *LearningSTP*, *ComputingPolicy*, and *ApplyIntervention* algorithms. *OSEPJITI* can be seen as a combination of the other algorithms. While *LearningSTP* tackles the problem of learning a TPM, *ComputingPolicy* uses that TPM to learn the policy that will guide the process of choosing

interventions for a set of episodes. Finally, these interventions are applied using the *ApplyIntervention* algorithm. Here, we present *OSEPJITI* in two different flavors: Using the same policy during the whole set of episodes or updating it every fixed number of episodes. (*OSEPJITI*-update). The remainder of this section explains the details about the algorithms and how they work together.

3.4.1 Learning the TPMs

So far, we have discussed how to model our problem as a MDP. However, solving an MDP using a model-based optimization approach, such as value-iteration, requires knowledge about the system model (i.e., TPM). We use First-visit MC method [39] to solve the MDP. Following a random policy, First-visit MC generates a pre-defined number of episodes (i.e., training) of different length.

Then, for each state s appearing in the episode, it computes the return following the first occurrence of s , accumulates it, and averages it as the ratio between the number of times we took action a in state s and got to s' , and the number of times we took action a in state s . As a result, we obtain the maximum likelihood estimates for the TPM, see Equation 3.1

$$P_{sas'} = \frac{\text{times we took action } a \text{ in state } s \text{ and got to } s'}{\text{times we took action } a \text{ in state } s} \quad (3.1)$$

Algorithm 1 depicts the estimation process. Let us, define A as the number of interventions, and $S = 2^A + 1$ as the number q-states (i.e., intervention-actions states) plus a stress relief state. Therefore, at the beginning, the multi-array $P_{sas'}$ of dimensions $S \times S \times A$ is created to store the values of TPMs. Initially, $P_{sas'}$, which works as a counter, is set to zero. Then, the system runs for a predetermined number of episodes (i.e., *TrainingSize*) choosing interventions according to the random policy π . The chosen intervention a is applied using the *ApplyIntervention()* function, resulting in a transition from state s to state s' . This transition is recorded and accumulated in $P_{sas'}(xindex, yindex, intervention)$. After the end of the for-loop, $P_{sas'}$ is updated using the *HandleConstraints()* function, which addresses the cases when the value of equation 3.1 is $\frac{0}{0}$. Finally, the ratio shown in equation 3.1 is computed by the *Normalize()* function, which normalizes to 1 the rows of $P_{sas'}$.

Algorithm 1: LearningSTP.

```
input : RandomPolicy, S, A, TrainingSize
output:  $P_{sas'}$ 
begin
  A  $\leftarrow$  numInterventions;
  qS  $\leftarrow$   $2^A + 1$ ;
   $P_{sas'}$   $\leftarrow$  array[S][S][A];
   $P_{sas'}$   $\leftarrow$  0 ;
  for  $i \leftarrow 1$  to TrainingSize do
    stress  $\leftarrow$  True;
    treatment  $\leftarrow$  All available no intervention applied so far;
    while stress  $\leftarrow$  True and Treatment  $\leftarrow$  Available do
      intervention = RandomPolicy (treatment);
      (newstress, newtreatment)= ApplyIntervention (stress, treatment, model);
      xindex = bi2de(newtreatment) +1;
      if newstress = True then
        | yindex = xindex;
      else
        | yindex = S;
      end
       $P_{sas'}(xindex, yindex, intervention)++$ ;
      stress=newstress;
      treatment = newtreatment;
    end
  end
   $P_{sas'}$   $\leftarrow$  HandleConstrains ( $P_{sas'}$ );
   $P_{sas'}$   $\leftarrow$  Normalize ( $P_{sas'}$ );
return  $P_{sas'}$ ;
end
```

3.4.2 The Algorithm

Algorithm 2: OSEPJITI algorithm.

```

input : S, A,  $\gamma$ , TrainingSize, numEpisodes
output: A state transition probability matrix STPM

begin
  for each combination a of interventions and s states do
    | model  $\leftarrow$  sampleBeta(alpha=.1, beta=1);
  end
   $P_{sas'} \leftarrow$  LearningSTP (S, A, TrainingSize,  $\pi$ );
  Policy  $\leftarrow$  ComputingPolicy ( $P_{sas'}$ ,  $R_{sas'}$ , S, A,  $\gamma$ );
  intercount  $\leftarrow$  array(1,numEpisodes);
  for  $i \leftarrow 1$  to numEpisodes do
    | stress  $\leftarrow$  true;
    | treatment  $\leftarrow$  All available no intervention applied so far;
    | while stress  $\leftarrow$  True and Treatment  $\leftarrow$  All-available do
      | | intervention  $\leftarrow$  Policy(treatment);
      | | (newstress, newtreatment)= ApplyIntervention (stress, treatment, model);
      | | /* Optional update  $P_{sas'}$  */
      | | // $P_{sas'} \leftarrow$  LearningSTP (S, A, i);
      | | stress=newstress;
      | | treatment = newtreatment;
      | | iterationPerEpisode++;
    | end
    | /* Optional update Policy */
    | if  $i \% 10 = 0$  then
      | | //Policy  $\leftarrow$  ComputingPolicy ( $P_{sas'}$ ,  $R_{sas'}$ , S, A,  $\gamma$ );
    | else
    | end
    | intercount( $i$ )  $\leftarrow$  iterationPerEpisode;
  end
  return intercount;
end

```

Algorithm 2 depicts our online mechanism to select Just-in-time interventions. At the beginning, we create a model that represents the conditional probability of not having stress s' , given that the patient received an intervention a or set of interventions when he/she was in state s i.e., $p(s' = relieved | s, a_1, \dots, a_n)$. This model has the form of an array structure of dimensions $ST \times ST \dots \times ST$, where ST is the number of stress

states (different from the q -states), and number of interventions is the number of times that ST appears in this product.

At the heart of the OSEPJITI algorithm is the *LearningTPM* function, which is executed for a pre-determined number of times (*TrainingSize*) with the objective of learning the system's model or TPM. The output of *LearningTPM* $P_{sas'}$ is used as input of *ComputingPolicy* function. This function, represented by the Algorithm 3, uses value iteration, a dynamic programming approach to find a semi-optimal policy. This policy indicates the intervention that will be applied to the patient given his/her clinical story (i.e., treatments applied so far). The variable *intercount* counts how many times per episode the system takes to find the correct intervention or combination of interventions that relieve patients from stress.

Algorithm 3: ComputingPolicy.

input : $P_{sas'}, R_{sas'}, S, P_{sas'}, A, \gamma$
output: semi-optimal Policy *Policy*

begin
 foreach q -state **do**
 | $Q(s, a) \leftarrow 0$;
 end
 repeat
 | **forall the** q -states, s, a **do**
 | $Q_k[s, a] = \sum_{s'} P_{sas'}(R_{sas'} + \gamma \max_{a'} Q_{k-1}[s', a'])$;
 | **end**
 until $(Q_k[s, a] - Q_{k-1}[s, a]) < \epsilon$;
 foreach state s **do**
 | $Policy[s] \leftarrow \operatorname{argmax}_a Q[s, a]$;
 end
 return *Policy*;
end

Setting stress to true indicates that an episode only starts when a patient is in *stress* state. In addition, the *treatment* variable is set to *all available*, which means that no intervention has been applied so far.

The body of the while-loop corresponds to an episode. Thus, while the patient is in stress state and there are interventions available to apply, the algorithm picks an intervention using the *Policy*. The chosen *intervention* is used as part of the input of the *ApplyIntervention* function. This function, represented by Algorithm 4, uses the model (i.e., sampled for a *Beta* distribution) and a probabilistic roulette approach to

Algorithm 4: ApplyIntervention.

```
input : model,treatment, intervention
output: newstress, newtreatment
begin
  newtreatment  $\leftarrow$  treatment;
  newtreatment(treatment) $\leftarrow$  true;
  index  $\leftarrow$  treatment;
   $P_{relieved} \leftarrow 1 - \text{model}(\text{index})$ ;
  if  $\text{rand} < P_{relieved}$  then
    | newstress = true;
  else
    | newstress = false;
  end
  return newstress newtreatment;
end
```

determine whether the patient was relieved from stress or not and his/her new treatment status. Finally, the stress and treatment states are updated as a result of having applied the intervention.

After the while-loop block, there is an optional step that consists of updating the state's transition probabilities $P_{sas'}$ and the system the policy. We can reduce the computational cost of computing these values in every episode by keeping counts for both the number of times action a was taken in state s and went to state s' , and the number of times action a was taken in state s . Then as we observe more episodes, we can simply keep accumulating those counts. On the other hand, having an updated $P_{sas'}$ it is easy to recompute the policy. In fact, we do not need to re-compute it on every episode. Instead, this can be done sporadically to improve the prediction power of the policy.

3.5 Performance Evaluation

This section describes how we modeled the different parts of the intervention system and presents the results of the performance evaluation.

3.5.1 Modeling the OSEPJITI System Components

The OSEPJITI mechanism works in episodes as follows. First of all, an episode only begins when the user emotional state is recognized as in stress. Initially, all the possible treatments are set to \emptyset or available. The mechanism works in two main stages. The first stage accumulates experience about the system dynamics by the creation of a state transition probability matrix (TPM) see Algorithm 1. This experience is optimized by Algorithm 3 which uses value iteration, a dynamic programming technique to produce a semi-optimal policy.

The second stage is represented by Algorithm 2, which is in charge of applying the policy obtained in first stage by Algorithm 4. The second stage might take two different approaches. The first approach uses the policy computed during the first stage to choose interventions given the current user state during the entire process. In contrast, in the second approach, the TPM is re-computed on every iteration and the policy is updated every fixed number of episodes as shown at the end of Algorithm 2. The second approach is meant to capture the possible change in the user behavior in the long run.

The first and second stage use the same mechanism to apply interventions (see Algorithm 4). An important parameter of this algorithm is the user's reaction to interventions model, $p(s = \text{relieved} | t_1, \dots, t_k)$, which corresponds to the conditional probability of relieve given the user has received a set of interventions. These probabilities are simulated by a β distribution with parameters shown in Table 4.1, in which most of the mass is accumulated near to zero.

3.5.2 Experiment Parameters

An exhaustive search for the best set of parameters for the *OSEPJITI* algorithm was made. Table 5.5 summarizes the parameters, the search space, and the best result. Each set of parameters was evaluated by running the *OSEPJITI* system over ten simulations with different random patient models. The mean number of intervention was the chosen metric to minimize. The obtained set of parameters were used for the remaining experiments.

On the other hand, Table 4.1 shows the parameters of the β distribution. For OSEPJITI and OSEPJITI, γ value was 0.9.

Table 3.1: *OSEPJITI* parameter search data.

Parameter	Min Value	Step	Max Value	Best value
γ	0.7	0.05	1	-6
relief reward	10	10	100	.8

Table 3.2: Beta distribution parameters for no stress probability sampling.

Parameter	Value
α	0.1
β	1

3.5.3 Experiments

A set of experiments are included. It is meant to compare the performance of *OSEPJITI* and *OSEPJITI*-update against a random policy. We compare the average number of steps per episode that these methods take in order to find the correct intervention or combination of interventions that relieve patients from a condition such as stress. Here, the x-axis corresponds to the number of episodes and the y-axis to the average number of steps needed to find the right treatment that relieves the patient from stress.

We explore different values for the *TrainingSize* parameter which corresponds to the number of episodes used to create the TPM.

Figure 3.1 compares the performance of a random policy against the policies obtained by the *OSEPJITI* and *OSEPJITI*-update algorithms. Here, *OSVOJITI*-update upgrades the TPM on every episode and the policy on every ten episodes, both algorithms are trained using 40 episodes. We observe that *OSVOJITI*-update slightly outperforms *OSVOJITI* in terms of the average number of steps per episode needed to relieve a patient from a condition such as stress.

Finally, we use a histogram to compare the performance of *OSEPJITI*, *OSEPJITI*-update, and random policy. In order to reach statistical significance, we run 500 episodes 100 times. Figure 5.7 shows the average and standard deviation of the 100 runs. Again, we used 40 episodes to train both algorithms, and *OSVOJITI*-update upgrades the TPM on every episode and the policy on every ten episodes.

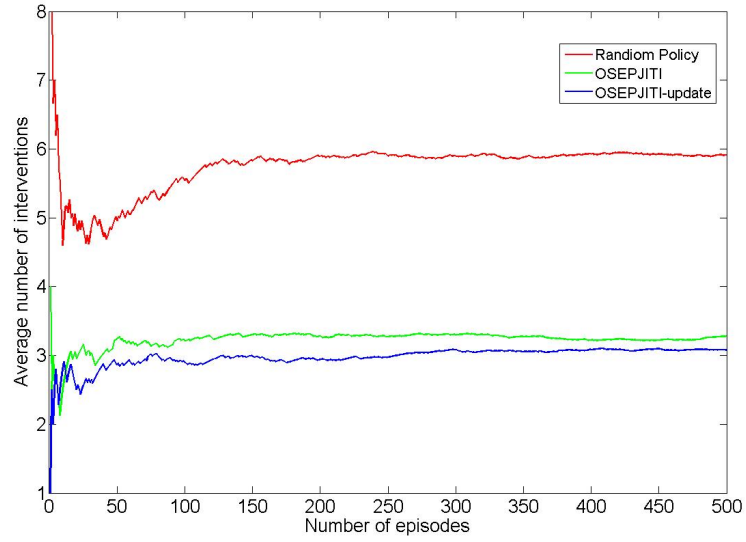


Figure 3.1: Random policy Vs OSEPJITI, and OSEPJITI-update $TrainingSize = 40$.

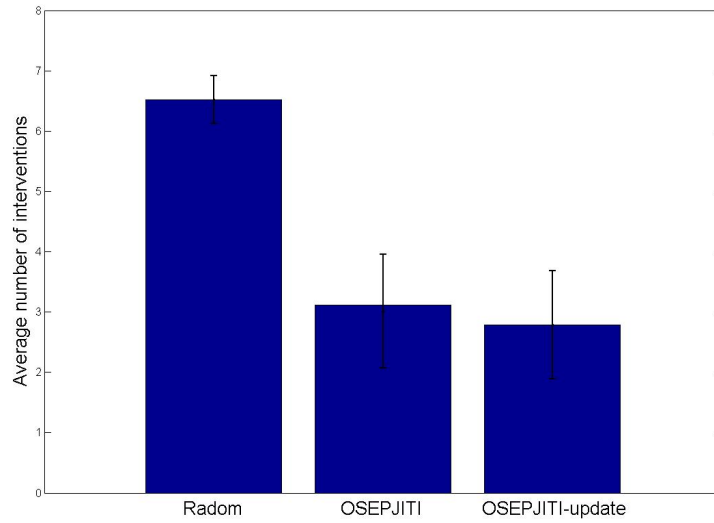


Figure 3.2: Average of 100 runs, random policy Vs, OSEPJITI, OSEPJITI-update.

3.6 Remarks

To the best of our understanding, this is the first work that uses value iteration to build transition probability matrix for just-in-time interventions.

CHAPTER 4

A MODEL-BASED FREE METHOD FOR INTERVENTION SELECTION

4.1 Note to the Reader

Part of this chapter was published in the proceedings of the 10th International Conference on Body Area Networks BodyNets 2014 [14]. Appendix A includes the permission to reuse such work in the dissertation.

4.2 Introduction

Several factors are driving a slow but steady shift in medicine. Today, doctors, hospitals, and clinics are the center of healthcare. Optimistically, people visit the doctor maybe once per year, where they receive expert advice, nudges (“you really need to reduce your fat intake”), treatments, and action plans, all within a short 10-15 minute meeting with the doctor. Afterward, the patient goes back to his/her normal daily life and the doctor moves on to the next patient.

Inexpensive wearable sensors, always-on internet connections, and computationally powerful smartphones are changing this model. Thanks to these technologies, the practice, distribution and delivery of healthcare is becoming democratized, patient-driven, personalized, inexpensive, timely, and perhaps most important, preventative. The greatest potential of these new technologies is in preventing health problems long before they happen. Imagine a smartphone that knows when a patient is craving a cigarette, and then intervenes in some way to convince and prevent the user from smoking a cigarette. If we apply this vision to other health problems, we may be able to short-circuit almost an entire class of common healthcare challenges, including stress, obesity, diabetes, and some cancers. While the technology components exist to enable this vision, several core challenges remain to make robust, mobile, cyber-physical health systems achieve it.



Figure 4.1: Hardware architecture of a context-aware mobile system.

In this dissertation, we examine JITAI for chronic stress. Stress has a negative impact on the human physiology not instantaneously noticeable. Rather, the effects of stress accumulate over time and lead to significant wear and tear on the cardiovascular system. A well-designed JITAI system could reduce this accumulation of negative effects by helping individuals reduce stress level on a daily basis.

Building on previous work on the timing constraints imposed by JITAI [7], we identify a need for dedicated intervention management as well as the ability to forecast health state in advance. The latter enables intervention before a sudden increase in stress. We then propose a three layer architecture for JITAI: a continuous sensing layer, a real time stress recognition and forecasting layer, and an adaptive intervention management module. Several mHealth systems include continuous sensing and health state recognition [8,9]. However, to our knowledge, forecasting and real-time adaptive intervention components have not yet been investigated in the literature. We investigate these two new components of the architecture more closely with post-hoc simulations on real-world data.

We ran a set experiments to show that it is feasible to forecast physiological variables associated with stress with up to 3 minutes in advance. We developed a forecasting mechanism based Hidden Markov Models (HMM) that computes the posterior predictive distribution to forecast the most like elements in the sequence. We evaluated the forecast error using two well known error indexes mean absolute error (MAE), and root mean square error (RMSE). We found that the value of these errors is less than half of the standard deviation of the actual data which according to author such Singh [66] is an indication of an appropriate model. In addition, we reached an agreement of 89% between the actual and forecasted data using three hidden states and training the HMM with one thousand elements.

Another key innovation of this dissertation is the use of Q-Learning to select any number of interventions at any number of time points. In previous work, Nahum-Shani et al [67] introduced the notion of using Q-learning for adaptive intervention. They used Q-Learning with linear regression to choose from two

interventions at two points in time. This intervention frequency and quantity was chosen because of the nature of medical practice, in which health professionals and patient interact infrequently, and there are little opportunities to deliver known interventions and try new ones. With the expansion of pervasive technology into our daily lives, it is now possible to deliver health interventions at far more frequent intervals and to try a much larger set of interventions over a longer period of time. We adapt Q-Learning to enable selection and delivery of interventions via pervasive, always-on, always-available technologies by introducing eligibility traces. The adaptations we describe here are an important step toward taking advantage of the characteristics of pervasive systems to improve health care.

4.3 Adaptive Just-in-Time Interventions

4.3.1 Modeling the Intervention Dynamics

We model the problem of selecting the intervention that maximizes the intervention effectiveness, as a Markov Decision Process (MDP). Here, at every decision point t , the system observes the user stress state s , and based on the observation selects an intervention a from a set A of available ones. As result, the user makes a transition to a new stress state s' with transition probability $P(s'|s, a)$ and returns a reward $R(a, s)$ [39].

Initially, we don't assume any knowledge about the intervention's effect on the user's stress state. Thereby, the algorithm has to learn and optimize the selection of the right intervention in real time, namely we do not consider a face of training and another of testing.

Reinforcement Learning (RL) [39] seems to be an appealing modeling framework to optimize the effectiveness of stress interventions, because it is designed to solve multi-stage delayed-result decision problems. In particular, we use an episodic Q-Learning(λ) (QL(λ)) algorithm.

4.3.2 Modeling a Stress Episode

We model our problem in terms of episodes or rounds. We consider an stress episode as the interval between the time when stress is detected (or forecasted) and the time when the patient shows no more sign

of stress. Additionally, only one intervention of each type can be delivered in each episode. Thus, an episode is also considered terminated if no other intervention is left to be delivered.

Let $\bar{A} = (a_1 \dots a_M)$ be the possible treatments that can be delivered to the patient during an episode of stress. Let $A_t = a_i$ the set of treatments given to the patient so far in the episode. Since an episode ending implies the patient has been relieved from stress, those interventions, namely the ones applied before the ending of the episode, are considered to have failed in curing the patient.

Then, we define the QL state at time t as $S_t = A$. We interpret A as a set instead of an ordered list. The underlying assumption is that the order in which treatments are delivered is not important. This assumption greatly reduces the state space. Notice, however, that the number of possible states still grows exponentially with the number of interventions.

A positive reward r_{relief} is given whenever the system succeeds in treating the patient and a negative reward $r_{intervention}$ is given after each intervention. The problem is then to find the policy $\pi : S \times A \rightarrow A$ that maximizes the expected reward, which is equivalent to relieving the patient's stress while minimizing the number of interventions.

Eligibility traces were implemented to improve the algorithm learning rate using a memory decay factor λ . Thus, the implemented algorithm is a QL(λ) algorithm. No discount factor γ was used given the episodic nature of the problem.

Exploration and exploitation were balanced through the use of an ϵ -greedy strategy, where ϵ depended on the relation between the value of the best known choice and the sum of all values for that state. Thus, exploration was performed with a probability as stated in Equation 4.1.

$$p(\text{exploration}, s) = \left(1 - \frac{Q(s, a^*)}{\sum_a Q(s, a)}\right) \quad (4.1)$$

where a^* corresponds to the action with the highest estimated value. Algorithm 5 summarizes one iteration of the simultaneous decision-making and learning process of the QL(1) algorithm.

4.3.2.1 Time Complexity for the Intervention Selection Algorithm

Koenig and Simmons [68] showed that the worse-case complexity of reaching a goal state has a bound of $\mathcal{O}(n^3)$ for Q-learning, and $\mathcal{O}(n^2)$ action executions for value-iteration, with n the number of states. In

Algorithm 5: QL(1) Online selection algorithm.

```

input : stress levels detected or forecasted
output: the set of interventions to select
begin
  forall the state action do
    | value(state, action) = 1
  end
  while stress and there are more interventions to perform do
    | i = chooseIntervention (treatment, value);
    | stress = applyIntervention (treatment, intervention);
    | r = reward (treatment, intervention, stress);
    | maxVal = max(value([treatment + intervention]));
    | updateETrace (treatment, intervention);
    | applyQLLearningRule (treatment, intervention, reward, maxVal);
  end
end

```

our case, the number of q-states or *state-actions* in Algorithm 5 is dependent on the number of interventions A , namely the number of state-actions is $N = 2^A + 1$. Thereby, in the worst-case scenario the loop will go up to A , causing a worst-case time complexity of $\mathcal{O}(A2^{3A})$, namely an exponential complexity in terms of the number of interventions.

4.3.3 Modeling the Patient Reaction to Interventions

We modeled a patient reaction to an intervention by creating an artificial distribution $p(s|t_1, \dots, t_N)$ corresponding to probability of not having stress under a set of treatments. We sample from a beta distribution with the parameters shown in Table 4.1.

Table 4.1: Beta distribution parameters for no stress probability sampling.

Parameter	Value
α	0.1
β	1

Figure 4.2 shows the plot for this distribution. As one can see, most of the probability mass is concentrated near zero. Thus, most of the treatment combinations will be inefficient, while a few of them will have a greater chance of relieving the patient.

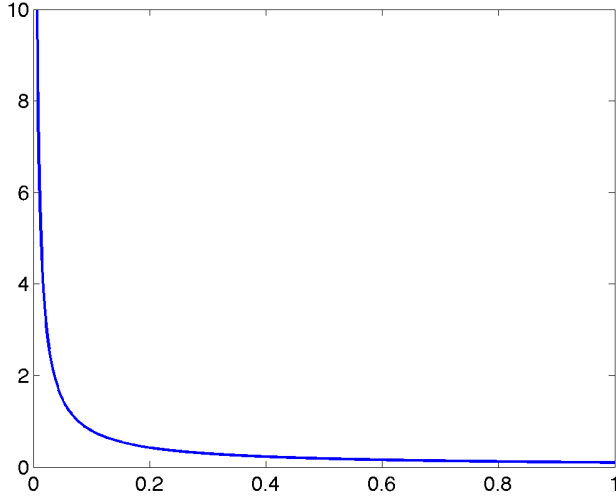


Figure 4.2: Beta distribution for probability sampling.

4.3.4 Design Considerations

The question of whether intervention's outcomes are independent of each other or not plays an important role in the design of JITIs systems because it affects the difficulty of the problem. If the interventions' outcomes are independent of each other, simple statistical approaches can be used to find the set of most effective interventions for a given patient. Then, the system would always try to deliver those interventions in decreasing order of effectiveness. However, in the medical setting, there is consensus that previous interventions may influence the effectiveness of new ones. We argue in favor of this statement with a graphical model of the stochastic outcome of different interventions. Figure 4.3 depicts the model in which the stress state s depends on the combination of N independent interventions i_j and a set of M attributes a_k that define the patient.

The joint probability $p(s, a_1, \dots, a_M, i_1, \dots, i_N)$ is then split into different factors expressing the effectiveness of each intervention under the patient attributes, according to Equation 4.2. Note that, in this case, intervention outcomes are indeed independent of each other.

$$p(s, a_1, \dots, a_M, i_1, \dots, i_N) = \prod_{j=1}^{j=N} \phi(s, a_1, \dots, a_M, i_j) \quad (4.2)$$

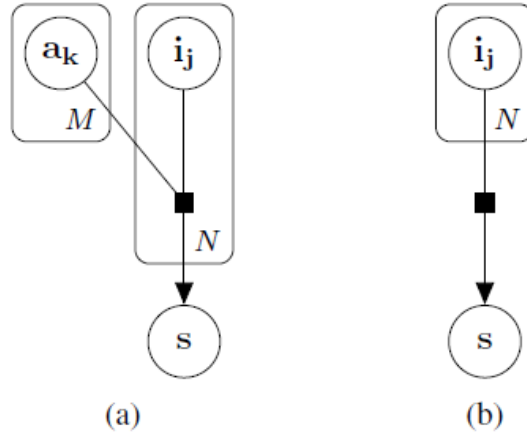


Figure 4.3: Graphical models of the patient's stress.

As the complete set of attributes can never be known, i.e. the factors that affect the effectiveness of each intervention cannot be fully determined, we would like to work with the marginal distribution $p(s, i_1, \dots, i_N)$. Then, due to marginalization of the attribute variables, the graphical model turns into the one shown in Figure 4.3. Under this assumption, the effectiveness of each intervention becomes dependent on one another.

Consequently, the decision regarding which intervention to apply at a given moment must take into account the outcomes of previous given ones.

The previous insight also suggests that the solution to this problem could be improved by introducing additional context. Firstly, in the case of JITIs for stress, context information such as the time of the day and the patient's location could give clues about the cause for the stress, which could be taken into account when selecting the interventions to apply. In addition, extra context such as whether physical activity is being performed by the patient may be important to distinguish a true or false event of stress (i.e. physical activity and stress have similar effects in the physiology). Finally, an a priori questionnaire could gather information of the patient's preferences of possible interventions.

4.4 Relationship Among Interventions, Treatments and Q-states, a Short Example

The following example using just two interventions illustrates our approach.

We consider two interventions a , and b . Then the state of the QL algorithm is fully determined by the applied interventions plus the relieved state (r), resulting in the set of possible states $\{\emptyset, a, b, ab, r\}$.

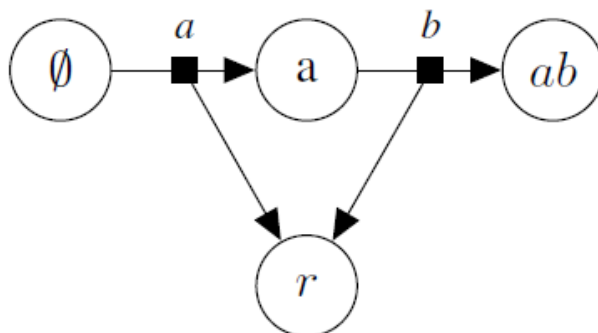


Figure 4.4: Q-states transition graph.

Figure 4.4 shows the transition between Q-states (i.e., ovals) given that an intervention has been applied (i.e., squares). In this example, the system chose between intervention a and b following some policy π . The transition graph shows that the policy chose a and the user might have gone to the Q-state a with probability P_a or to the *relieved* state with probability $1 - P_a$. If patient was not relieved and he/she is in state a a reward of zero is assigned to this transition; on the other hand, if the patient is relieved a positive reward is assigned. At this point, the only available intervention is b , because our design constraints indicate that the same intervention cannot be applied more than once. Thereby, the system applied intervention b , and again, the user might have gone to the terminal state ab in which case it would be penalized with a negative reward, or it might have gone to the *relieved* state in which case it would be rewarded with a positive reward. Here, $1 - p_a$ and $1 - p_b$ are sampled from a beta distribution $\beta(0.1, 1)$.

This short example shows the relationship between interventions, and Q-states. In addition, it illustrates the relationship between the patient states, namely $\{stress, non - stress\}$, and the Q-states $\{\emptyset, a, b, ab, r\}$. Thus, if the user is in $\{r\}$ Q-state, namely *relieve*, he/she is considered in the user $\{non-stress\}$ state. However, if he/she is in any of the remaining $\{\emptyset, a, b, ab\}$ Q-states, then he/she is considered in $\{stress\}$ user state.

Additional concepts, such as how the mechanism uses the user model to determine the patient new treatment and the new emotional state, and how to obtain a policy and optimize it are addressed in the rest of this dissertation.

4.5 Predicting and Forecasting Stress

The purpose of this layer is to switch the system to its preventive mode and turning it into a *Preventer*. Here, the system makes predictions about future stress values and these values are in turn used by the stress *management layer* to deliver preventive interventions as shown in Figure 1.2.

In this case, we use Heart Rate Variability (HRV) which is one of the sensing layer outputs. At every time t the *sensing layer* provides a new fresh HRV value, and the forecaster uses it along with the previous $\{k, \dots, t - 1\}$ to predict the HRV values at times $t + 1$, $t + 2$, and $t + 3$.

This layer takes the form of a forecasting algorithm which is based on a first-order Hidden Markov Model that uses the posterior predictive distribution density as the prediction method. Section 4.5.3, and Algorithms 6 and 7 depict this approach.

4.5.1 Data

We use a real-world continuous physiological dataset in our experiments described below. This data was collected as part of a larger study on the inferring stress from a suite of physiological measures [1]. Participants wore the AutoSense [8] system underneath their clothes for four weeks as they went about their normal daily lives. Table 4.2 depicts the demography of the participant of the study. In this dissertation, we only examine the ECG data and use Heart Rate Variability (HRV) as a proxy for stress. We used this measure because it has been widely used in the literature [69–71] as one of the most important features for stress recognition. HRV measurements are reported as averages over consecutive one minute windows.

In this work, we only examine the ECG data and use Heart Rate Variability (HRV) as a proxy for stress. We used this measure because it has been widely used in the literature [69–71] as one of the most important features for stress recognition. HRV measurements are reported as averages over consecutive one minute windows.

4.5.2 Basic Definitions

HMMs are one of the probabilistic graphical approaches most widely used to model physical, behavioral, and social phenomena. They are useful to model situations which involve time and processes whose states

Table 4.2: Demographics of participants in the two studies [1].

	Statistics	Drug users	Daily Smokers
	of Participants	40	30
	of Males	29	15
	of Females	11	15
	Age	41+10	24.25+6.25
Race	White	19	19
	African-American	20	8
	Asian	0	3
	Refuse	1	0
Education status	High School Grad	40	0
	University Grad	0	30
Employment status	Full Time	15	6
	Part Time	10	14

are not evident, but whose manifestations can be observed through sensor readings. A HMM is characterized for the following elements:

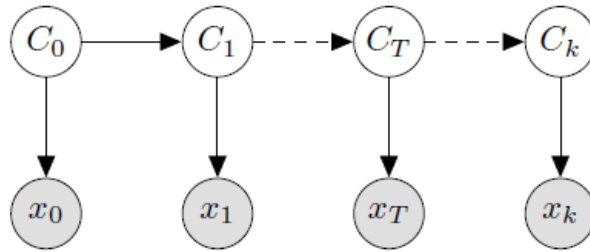


Figure 4.5: HMM graphical model, $k = T + h$, and h is the forecasting horizon.

- The set of hidden states, $C^T = \{C_0 = i, C_1, \dots, C_T\}$.
- The observable process, $X^T = \{x_0, x_1, \dots, x_T\}$.
- The initial state probabilities, $\delta_0 = Pr(C_0 = i)$.
- The transition state probabilities, $\gamma_{ij}(t) = Pr(C_{s+t} = j | C_s = i)$.

- The emission probabilities, $\gamma_{ij}(t) = Pr(C_{s+t} = j | C_s = i)$.

Here, transition, emission, and initial state probabilities are stochastic matrices.

4.5.3 Prediction Model

Our prediction model is based on the posterior predictive distribution, in which the set observations $X_T = \{x_1, \dots, x_T\}$ are given, and we want to predict the x_{T+h} distribution. The predictive distribution can be computed as follows:

$$\begin{aligned}
P(x_{T+h} = i | X_{1:T}) &= \sum_{c_{T+h}} Pr(x_{T+h}, c_{T+h} | X_{1:T}) \\
&= \sum_{c_{T+h}} Pr(x_{T+h} | c_{T+h}) Pr(c_{T+h} | X_{1:T}) \\
&= \sum_{c_{T+h}} Pr(x_{T+h} | c_{T+h}) \sum_{c_T} Pr(c_{T+h}, c_T | X_{1:T}) \\
&= \sum_{c_{T+h}} Pr(x_{T+h} | c_{T+h}) \sum_{c_T} Pr(c_{T+h} | c_T) Pr(c_T | X_{1:T}) \\
&= \sum_{c_{T+h}} Pr(x_{T+h} | c_{T+h}) \sum_{c_T} Pr(c_{T+h} | c_T) \frac{Pr(c_T, X_{1:T})}{Pr(X_{1:T})} \\
&= \frac{1}{Pr(X_{1:T})} \sum_{c_{T+h}} Pr(x_{T+h} | c_{T+h}) \sum_{c_T} Pr(c_{T+h} | c_T) \alpha(c_T)
\end{aligned}$$

Let us express $\frac{Pr(c_T, X_{1:T})}{Pr(X_{1:T})}$ as ratio of likelihoods.

$$\phi_T \leftarrow \frac{\alpha(c_T)}{\sum_i \alpha(c_T(i))} = \frac{Pr(c_T, X_{1:T})}{Pr(X_{1:T})}$$

Let us express the transitions, and emissions probabilities in matrix notation.

$$\Gamma^h \leftarrow Pr(x_{T+h} | c_{T+h}), \quad \Lambda(x) \leftarrow Pr(c_{T+h} | c_T)$$

Finally, we express the posterior predictive distribution in a matrix notation.

$$P(x_{T+h} = i | X_{1:T}) = \sum_i \phi \Gamma(i)^h \lambda_i(x) = \phi_T \Gamma^h \Lambda(x) \mathbf{1}'$$

This distribution is computed by Algorithm 6 and Algorithm 7, respectively.

4.5.4 Prediction Algorithm

The HMM forecasting algorithm takes the form of Algorithms 6 and 7. Data processing, and HMM parameter estimation take place in Algorithm 6. Here, the signal is discretized by using the Symbolic Aggregate approximation algorithm (SAX) [72], which transforms the HRV time series in a new discrete series

Algorithm 6: Parameter estimation.

input : i a new HRV value at time t , Time window X_{N+1}^{N+k} of size k , number of states m , initial estimations of Λ_0, Γ_0 , and δ_0

output: Λ, Γ , and δ

begin

$Y \leftarrow X_{N+1}^{N+k}$

/* Discretizing a new HRV element */

$s_i \leftarrow \text{SAX}(i)$

/* Adding the new s_i value at the end of the time window */

$Y \leftarrow Y \cup \{s_i\}$

/* removing s_j at the beginning of the time window */

$Y \leftarrow Y \setminus \{s_j\}$

/* Parameter estimation using Expectation Maximization algorithm */

$(\Lambda, \Gamma, \delta) \leftarrow \text{EM}(Y, \Lambda_0, \Gamma_0, \delta_0, m);$

return $(\Lambda, \Gamma, \delta);$

end

of alphabet size 20. Thereby, every minute, the sensing and stress recognition layer provides a new HRV value i which is transformed by SAX into a new symbol s_i and it is added as a new element of the training set. At the same time, the first element s_k of the training set is removed (i.e., sliding window movement) resulting in the new training set $x_{N+1} \dots x_{N+k}$. Finally, the parameters of the HMM are estimated by using the Expectation Maximization (EM) Algorithm. The parameters of the forecasting function include: The current time window Y , the initial HMM parameters δ_0 , the TMP Γ_0 , and the emission probability matrix Λ_0 .

Algorithm 7 corresponds to the forecasting component. At the beginning, we store all the different symbols of the new discrete time series in array *range*. Then, the HMM parameters are estimated by calling the function `ParEstimate(Y, δ_0, Γ_0, m)` which corresponds to Algorithm 6. We then call the function `Forward()` which uses as input the output of function `ParEstimate` to compute the forward probability. The next three lines compute ϕ a ratio between the likelihood α_T and the summation of the likelihood components $\alpha_T 1'$. Next, in the outer for-loop we compute $\sum_i \phi \Gamma(, i)$, which corresponds

Algorithm 7: Posterior predictive distribution.

input : i a new HRV value at time t , X the time series, number of states m , initial estimations of Λ_0, Γ_0 , and δ_0 , H forecast horizon

output: *forecastD* a posterior predictive distribution

```
begin
  /* range of symbols  $s$  of the discrete time series          */
  range  $\leftarrow \{s^1, \dots, s^k\}$ ;
   $(\Lambda, \Gamma, \delta, s_i) \leftarrow \text{ParEstimate}(Y, \Lambda_0, \Gamma_0, \delta_0, m, i)$ ;
  /* computing the forward probability                          */
   $\alpha \leftarrow \text{Forward}(Y, \Lambda, \Gamma, \delta, m)$ ;
   $c \leftarrow \max(\alpha[n])$ ;
  /* Scaling the likelihood                                    */
   $\text{logk} \leftarrow c + \log(\text{sum}(\exp(\alpha[n] - c)))$ ;
   $\text{left} \leftarrow \exp(\alpha[n] - \text{logk})$ ;
   $\text{right} \leftarrow \text{diag}(m)$ ;
  /*  $m \times H$  matrix to store the predictive distributions    */
   $\text{storage} \leftarrow \text{zeros}(m, H)$ ;
   $\text{prePredic} \leftarrow \text{zeros}(m, H)$ ;
  for  $i \leftarrow 1$  to  $H$  do
     $\text{right} \leftarrow \text{right} \times \Gamma$ ;
     $\text{storage}(:, i) \leftarrow \text{storage}(:, i) + \text{left} \times \text{right}$ ;
    for  $j \leftarrow 1$  to  $|range|$  do
       $\text{forecast} \leftarrow \text{forecast} + \text{storage}(:, i) \times \lambda_{ij}$ 
    end
  end
  return  $\text{forecast}$ ;
end
```

to the state prediction distribution $Pr(C_{T+i} = x|X_{1:T})$. The resulting columns' vectors are then stored in the $m \times H$ matrix *storage*, where m is the number of hidden states, and H is the forecasting horizon. In the inner for-loop, we compute the posterior distribution $\sigma(T+h) = \sum_j \phi\Gamma(\cdot, i) \times \lambda_{ij}, \forall s_j, j = 1, \dots, |range|$. Thereby, for each symbol s the next h -step probability $\sigma(T+h) = Pr(X_{T+h} = i|X_{1:T})$ is computed by $\sum_j \phi\Gamma^h(\cdot, i) \times \Lambda$. Finally, the most probable symbol is computed as $s_{T+h}^* = \arg \max[\sigma_{T+h}(s)] \forall s$.

4.5.4.1 Forecasting Algorithm Time Complexity

The time complexity for our forecasting proposal depends on the complexity of Algorithms 6 and 7, respectively. In the former case, the time complexity is given by the algorithm's most expensive computation, namely the estimation of the HMM parameters by using the EM() algorithm. In the latter case, the algorithm calls the function `ParEstimate()` which corresponds to Algorithm 6 and then calls the `Forward()` algorithm to compute the predictive state density. Khreich *et al.* [73] and Salojärvi *et al.* [73] showed that given a time series of length T and a N state HMM, the time complexity per iteration for the EM and `Forward` algorithms is $\mathcal{O}(N^2T)$. Thereby, the time complexity of our forecasting algorithm is driven by the sum of the time complexities of EM and `Forward` algorithms, namely, $\mathcal{O}(N^2T)$.

4.6 Remarks

This chapter presents the core of this dissertation, namely a preventive just-in-time intervention system. This is possible thanks to the combination of our forecasting algorithm and our model free method to select interventions.

CHAPTER 5

PERFORMANCE EVALUATION

5.1 Note to the Reader

Part of this chapter was submitted to IEEE-Transactions on Affective Computing [15] Journal and it is still under review.

5.2 Forecasting Experiments

In this section we describe the metrics and experiment used to evaluate our proposal. In a first part, we run a set of experiment to show that it is feasible to have an intuition about the user's future stress behavior. Using real HRV data, we compute the posterior predictive distribution density for one, two, and three minutes-ahead. We use the Mean Absolute Error, the Index of Agreement, and the Coefficient of determination to measure the performance of our forecasting methodology.

In the second part, we evaluate our mechanism to select interventions. This evaluation is done in two stages, in the first stage, we compare the performance of the policy learned by using our algorithm vs a random policy in relieving a patient from stress. In the second stage, we compare the performance of using our methodology to select interventions vs CALM [13] a proposal for intervention selection based on value iteration. We also discuss, additional issues such as the time complexity as well as the learning rate of both methodologies.

A set of three experiments were carried out to measure the ability of our forecasting mechanism to predict stress. In all the experiments, the training set corresponds to a sliding window of a thousand minutes which runs through a HRV time series.

Thereby, every minute, a training set Y along with the initial parameters described in Table 5.3 are used as input of the Expectation Maximization algorithm (EM) [38] to optimize the initial parameters and

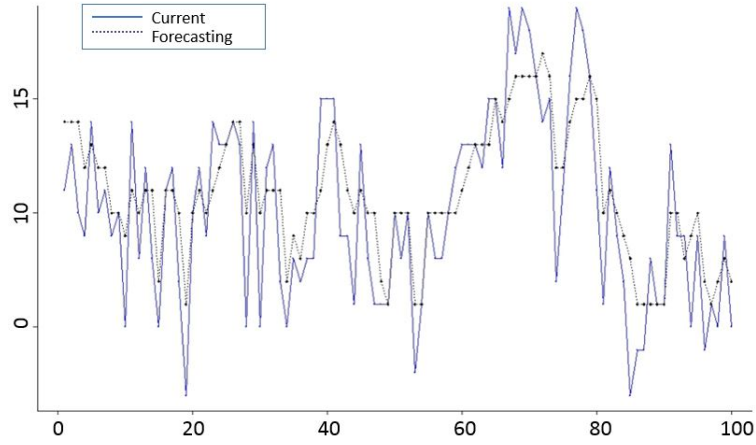


Figure 5.1: One step-ahead forecasting using a four hidden states Poisson HMM.

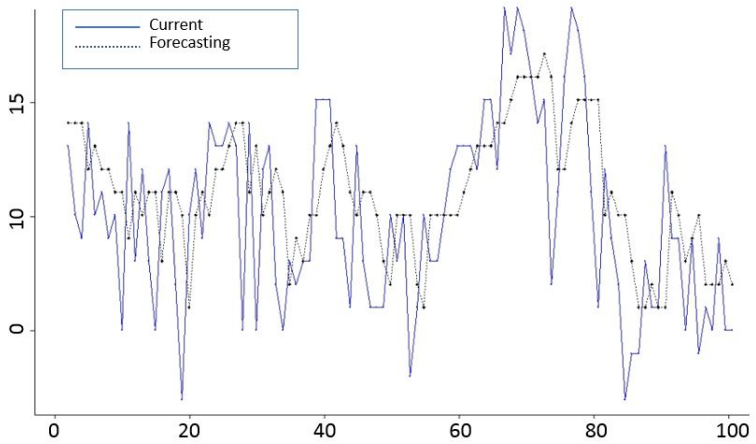


Figure 5.2: Two step-ahead forecasting using a four hidden states Poisson HMM.

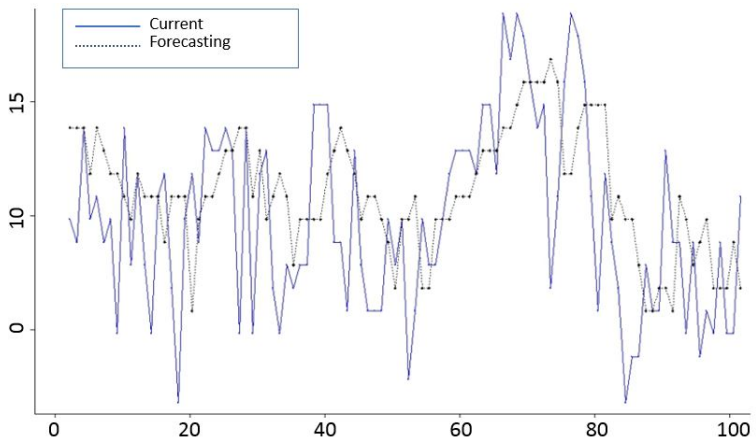


Figure 5.3: Three step-ahead forecasting using a four hidden states Poisson HMM.

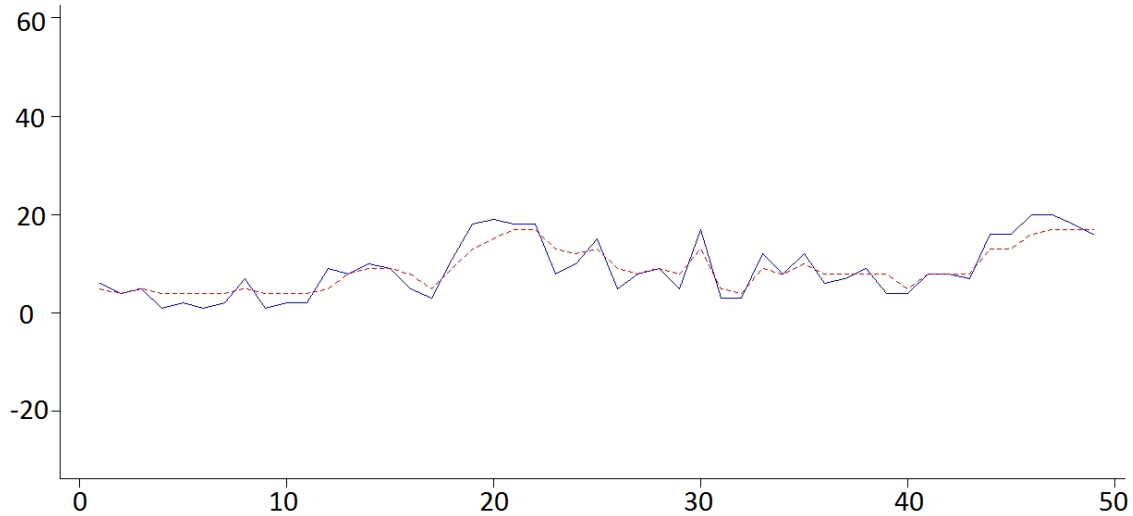


Figure 5.4: Capturing the data trend.

re-compute the HMM model. Then, the new parameters are used as input of Algorithm 7 to compute the likelihood for the potential candidates to be the next sequence element in the series. Finally, the one with the maximum likelihood is selected as the predicted element. Each experiment was repeated fifteen times to reach statistical significance. An instance or repetition consists of forecasting a hundred minutes (e.g., using one, two, or three minutes-ahead forecasting approach) of HRV using the parameters specified in Table 5.3.

Furthermore, the contrast between the observed values of the raw series and the predicted values obtained through our algorithm were compared to determine the efficacy the our forecasting methodology.

Error indexes such mean absolute error (MAE), and root mean square error (RMSE) were selected as the evaluation metrics because they are widely used in literature to understand the magnitudes of the forecasting errors and the accuracy of the models [74–77]. The values of MAE and RMSE fluctuate from 0 to $+\infty$, and their units are the same as the units of the actual data. Singh [66] stated that the RMSE and MAE values less than half of the actual data standard deviation may be considered low and good indicator of an appropriate model.

In addition, we selected the index of agreement d developed by Willmot [78] as a standardized measure of the degree of model prediction error. This index represents the ratio between the mean square error and the potential error, and varies between 0 and 1, a value of 1 indicates a perfect agreement between the actual and forecasted values, while 0 indicates total disagreement.

Finally, we used the coefficient of determination r^2 , this well known index refers to forecasting errors in relation to the mean. R^2 fluctuates between 0 and 1 and since it is the outcome of a ratio it tells us the percentage of the total variation (errors square) explained by the forecasting method in relation to the mean. Higher values indicating less error variance, and typically values greater than 0.5 are considered acceptable [79, 80].

These indexes were computed following the definition stated in Table 5.1. Here, S_t and O_t correspond to the predicted and observed values at time t , and N to the number of predictions.

Table 5.1: Evaluation metrics used in this dissertation.

Metric	Description	Definition
MAE	Mean Absolute Error	$\frac{1}{N} \sum_{i=1}^N (S_t - O_t) $
RMSE	Root Mean Square Error	$\sqrt{\frac{1}{N} \sum_{i=1}^N (S_t - O_t)^2}$
d	Index of Agreement	$1 - \frac{\sum_{t=1}^N (O_t - S_t)^2}{\sum_{i=t}^N (S_t - \bar{O} + O_t - \bar{O})^2}$
r^2	Coefficient of determination	$1 - \frac{SS_{res}}{SS_{tot}}$

5.2.1 Experiment Parameters

We carried out a set of three experiments, and for each experiment, we used the one, two, and three minutes-ahead forecasting approach. We repeated experiments one, two, and three with four, three, and two hidden states. Table 5.2 summarizes the initial parameters common to all experiments, while Table 5.3 shows the specific set of parameters for each experiment.

5.2.2 Experiment Results

Table 5.4 summarizes the results for the three experiments. The first, second, third, and four table rows show the experiment number, the forecasting approach (i.e., one, two, and three minutes-ahead), the number of hidden states used for that experiment, and the metric values computed for each experiment.

Table 5.2: Forecasting initial parameters common to all experiments.

Parameters	Value
Number of experiments	3
Number of participats	10
Number of repetitions per experiment	15
Time series size	4506 minutes of HRV
Distribution	Poisson
Parameter estimation	EM Algorithm
Forecasting range	100
Forecasting horizon (h)	1-3
Training set size	1000
Type of training set	Sliding Window
Lambda	kmeans(training-set, m)

Table 5.3: Forecasting experiment initial specific parameters.

Parameters	Experiment 1	Experiment 2	Experiment 3
Hidden states (m)	4	3	2
Transition-Prob matrix	$\begin{pmatrix} 0.7 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.7 \end{pmatrix}$	$\begin{pmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{pmatrix}$	$\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$
Delta	(0.25, 0.25, 0.25, 0.25)	(0.333, 0.333, 0.333)	(0.5, 0.5)

Table 5.4: Forecasting experiment results.

Predict approach (mins-ahead)	Experiment 1			Experiment 2			Experiment 3		
	1	2	3	1	2	3	1	2	3
Hidden States	4			3			2		
MAE	1.88	2.99	3.19	1.91	3.12	3.49	2.16	3.14	3.27
RMSE	2.36	3.66	3.94	2.35	4.00	4.30	2.66	3.95	4.13
d	0.87	0.67	0.60	0.89	0.64	0.56	0.87	0.72	0.69
r^2	0.71	0.26	0.18	0.7	0.2	0.12	0.59	0.24	0.2

After discretization the minimum and maximum values of the time series were 3 and 20 respectively, with standard deviation sd of 6.83. These values help us to make sense about the error indexes MAE and RMSE. For the first experiment where we used 4 hidden states, and a training set of 1000 samples, we obtained RMSE values of 1.88 2.99 3.19 for one, two, and three minutes ahead predictions. All these values are still below of half of sd of the actual data. This values represents the sd deviation of the errors, namely the difference between the actual and predicted values, while MAE measures the variance of this errors.

The index of agreement keeps above 0.5 in all the experiment, and especially high for one step ahead prediction. Finally, in all the experiment the coefficient of determination r^2 drops for predictions with horizon greater than one. As can be observed in Table 5.4 there is little difference in terms of the coefficient of determination (r^2) and Index of Agreement (d) when the HMM forecaster was used with four and three hidden states. However, in terms of computational cost, prediction using four states is more costly. Thus, using three states offers a good trade-off between prediction accuracy and computational cost.

The type of forecasting used in this project is not recursive, namely the forecasted value x_t was not used to predict the next element x_{t+1} . Instead, the sensing and stress recognition layer provides us a new x_t value every minute, which in turns is used to forecast the $\{x_{t+1}, x_{t+2}, x_{t+3}\}$ values at the same time. Figure 5.4 shows a zoom out of the first 50 minutes of forecasting, here we can see that even though the algorithm does not predict exactly the next point at times x_{t+2}, x_{t+3} , the system predicts the distribution or the trend of the future values.

5.3 Stress Intervention Experiments

5.3.1 Simulation Settings

We simulated the intervention system using a mathematical model of the patient reaction to the interventions. Carrying out simulations before testing with actual subjects is important due to the online nature of the experiments. Namely, the system must be fully functional and optimized before performing costly experiments on real human beings. The platform also provides a suitable testbed to try out new ideas before deployment with real users.

We modeled a patient by a beta distribution $p(s|t_1, \dots, t_N)$ corresponding to the factor graph shown in Figure 4.3. The probability of not having stress under a set of interventions was sampled from a beta distribution with the parameters $\alpha = 0.1$ and $\beta = 1$.

Algorithm 8: Simulated experiments for the intervention system.

```

begin
  for experiment in 1:100 do
    for each combination c of interventions do
      | model(s—c) = sampleBeta (alpha=.1, beta=1);
    end
    for episode in 1:1000 do
      | treatment(:) = 0;
      | while stress and remaining interventions do
      | | intervention = QLpickIntervention (treatment);
      | | treatment(intervention) = 1;
      | | stress = sampleBinom (model(treatment));
      | | QLlearn (treatment, intervention, stress);
      | end
    end
  end
end

```

5.3.2 QL vs Random Policy

The goal of this experiment is to compare the performance of a random policy vs that of a policy obtained by using QL in relieving a patient from stress. In order to do that, we compare the average number of steps per episode that each of these policies takes to find the intervention that relieves the patient. One thousand episodes were executed and the experiment was repeated 100 times. Finally the results were averaged. In the second experiment we compare the performance of two policies using 1000 episodes, and each episode was computed 100 times to measure its variability.

5.3.3 Experiment Parameters

A brute force search of the QL algorithm parameters was made. Table 5.5 summarizes the parameters, their search space and the best resulting combination. Each set of parameters was evaluated over ten simu-

Table 5.5: QL parameter search data.

Parameter	Min Value	Step	Max Value	Best value
α	0.6	0.05	0.9	.8
λ	0.7	0.05	1	.8
relief reward	10	10	100	10
intervention cost	-10	2	0	-6

lations with different random patient models. The mean number of interventions was the chosen metric to minimize. The obtained set of parameters was used for the remaining experiments.

After that, 100 different models were generated using this sampling technique. For each of the models, 1000 episodes were simulated. In each episode, the patient stress was set and the QL system was executed. For each intervention, the probability of stress was taken from the conditional probability of the patient model. Algorithm 8 summarizes this. We also implemented a simple system that picked interventions at random. This system served as a way of measuring how difficult it was to relieve stress under the current models.

5.3.4 Experiment Results and Discussion

Figure 5.5 corresponds to an experiment instance, namely the result of running 1000 episodes without repetition. In this experiment the x-axes correspond to the number of episodes, and the y-axes to the average number of steps per episode needed to relieve a patient from stress. Here, at the second episode both policies delivered around 7 out of 10 interventions to relieve the patient from stress. This agrees with the fact that the system does not know anything about the effect of the intervention on the patient’s stress states. However, at the 50 episode the QL just need to deliver 5 interventions, and at episode 100 around 2.5 interventions.

Figure 5.6 shows the result of running the previous experiment 100 times, the average per every 1000 execution is computed and at the end, the average of the 100 executions is calculated. Finally, the standard deviation of the 100 exclusions is shown at the top of each bar.

Finally, Figure 5.7 shows the average and standard deviation per episode performed by QL (blue) vs random (red) algorithms over 1000 iterations.

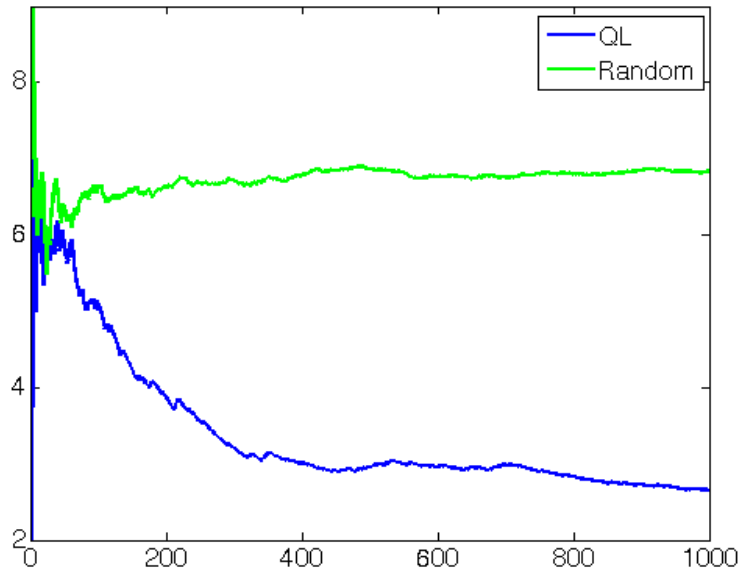


Figure 5.5: Average number of intervention to relieve stress.

A Student's t-test was performed on the data obtained from all repetitions of the experiment. The number of interventions was found to be significantly different for the QL and Random algorithms with a $p < 1e - 15$.

5.3.5 QL vs CALMA

The goal of this experiment is to compare the performance of CALMA [13] vs QL. Here, as before, we compare the average number of steps per episode that these methods take to find the correct intervention or combination of them that relieves patients from stress.

Unlike QL CALMA is not completely an online learning mechanism. Thereby, this algorithm which is based on value iteration has to create a transition probability matrix (TPM) using a pre-determined number of episodes. The intervention selection during the training stage follows a random policy, which means that the TPM is learned from a random policy.

As in the previous set of experiments, we compare the performance of the policies obtained by QL vs CALMA using training sets of size 20 and 40, during 1000 episodes. Again, the experiments were repeated 100 times, and the average, and standard deviation of the 100 repetitions was computed.

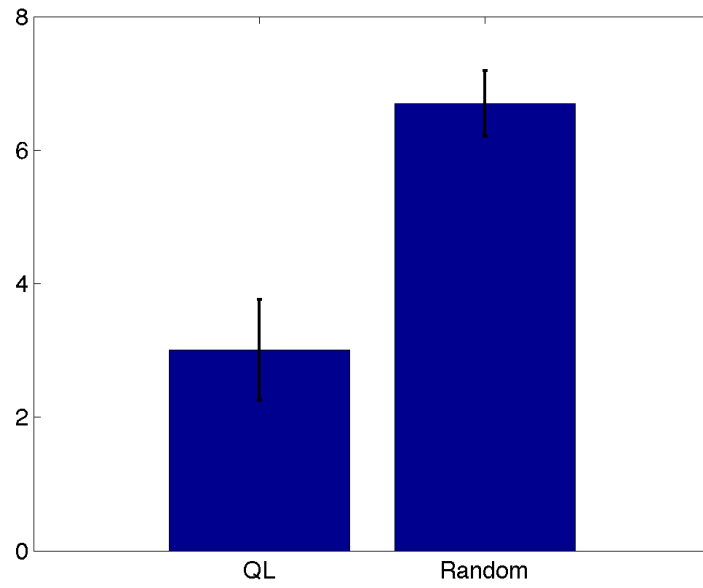


Figure 5.6: Cumulative average of the number of interventions per episode to relieve stress.

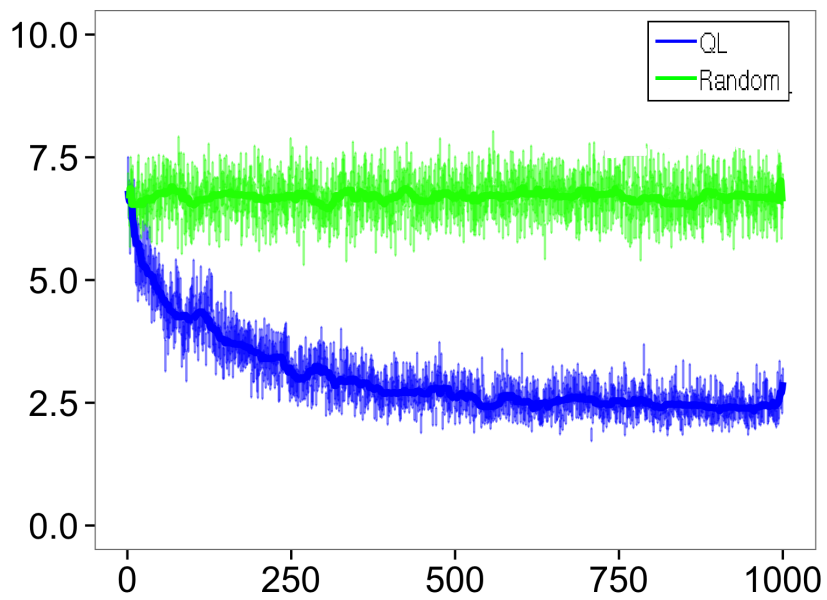


Figure 5.7: Average number of intervention to relieve stress, variability per episode, 100 repetitions.

5.3.6 Experiment Parameter

For QL and CALMA we use the same set of parameters, namely, the one listed in Table 5.5. In addition, we use training sets of size 20, and 40 episodes to train CALMA.

5.3.7 Experiment Results and Discussion

Figure 5.8 and Figure 5.10 show the accumulated average in graph bars, additionally the standard deviation among the 100 repetition is shown at the top of each bar. Here, the results are very similar, however the running time of CALMA is significantly greater than the running time of QL. The creation of TMP adds an extra complexity time complexity that make the algorithm not suitable to run in the mobile environment.

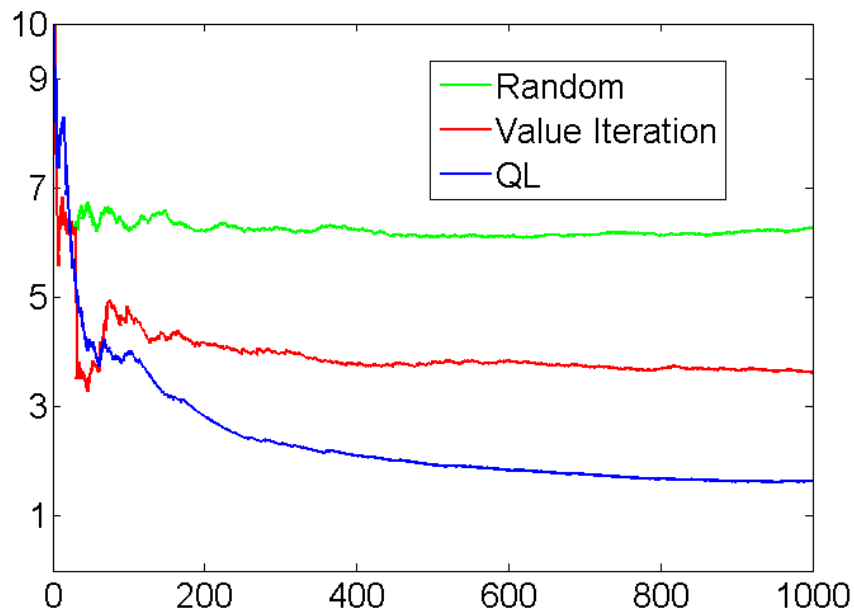


Figure 5.8: Random policy Vs QL Vs value iteration with a $TrainingSize = 20$.

5.4 Remarks

This chapter presents the performance evaluation of our proposed preventive just-in-time intervention.

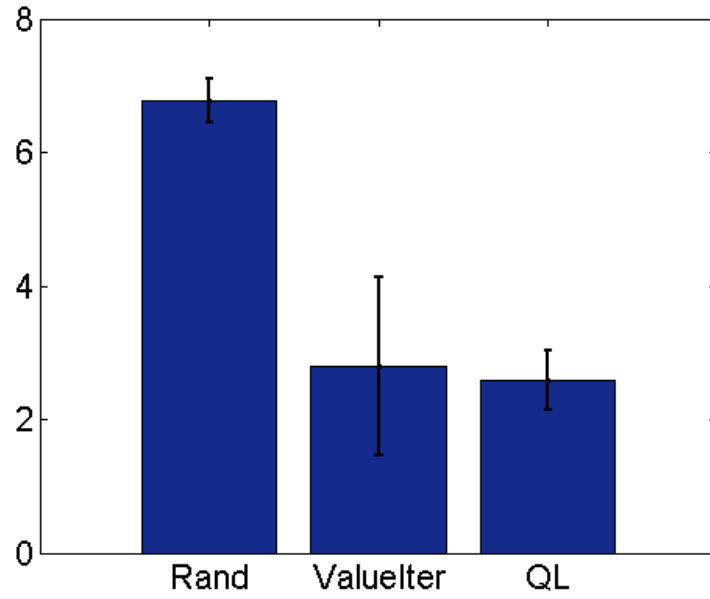


Figure 5.9: Cumulative average random policy Vs QL Vs value iteration with a *TrainingSize* =20.

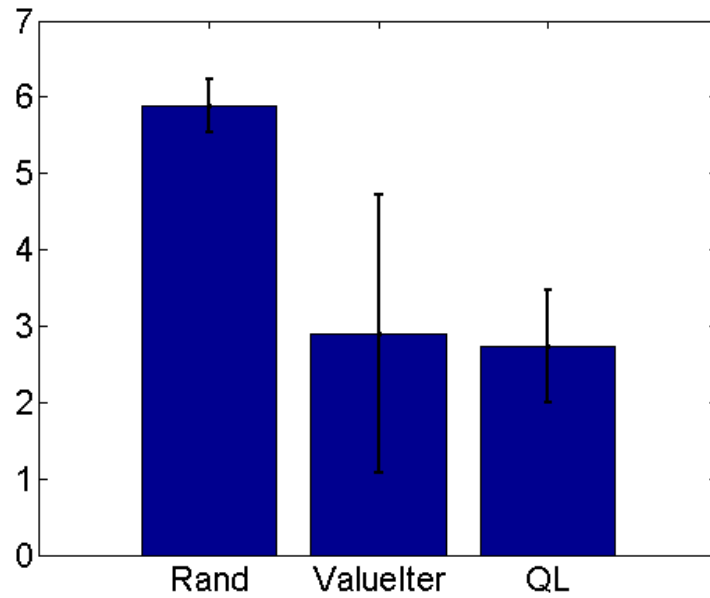


Figure 5.10: Cumulative average random policy Vs QL Vs value iteration with a *TrainingSize* =40.

CHAPTER 6

CONCLUSIONS

This chapter summarizes the most relevant findings and results of this dissertation and includes a number of ideas for future research in privacy-preserving mechanisms for participatory sensing.

6.1 Summary of Results and Findings

In the first case, we used a real data set from the AutoSense [8] project, this system developed by the University of Memphis is able to recognize stress in natural environments with up to 90% of accuracy. We developed a forecasting mechanism based on a Poisson Hidden Markov model that uses this dataset to compute the posterior predictive distribution to predict HRV values with one, two, and three minutes ahead. We ran a set of three experiments using two, three, and four hidden states. In all the cases, the Mean Absolute Error (MAE) was less than half of standard deviation of the actual data, and the agreement between the actual and predicted time series represented by the index of agreement d was greater or equal than 87%, 67%, 60% for one, two, and three minutes ahead respectively, using four hidden states, and 1000 units of HRV as training set.

In the second case, we showed a reinforcement learning based approach to optimize the selection of the interventions in order to maximize the effectiveness of the treatments. We did not assume any knowledge about the effect of applying an intervention on the patient stress state. In order to simulate this effect, we used a model based on a beta distribution. We sampled from $\beta(0.1, 1)$ to assign probabilities P and $1 - P$ to the transitions between the *stress* and *relieve* states, and *stress* and *stress* states, respectively. Thus, most of the time P will be close to zero, meaning that is very likely that the application of an intervention will fail to relieve a patient from stress. Despite this adverse scenario, our algorithm learned a policy for intervention

selection that is able to relieve a patient from stress using an average 2.8 out of 10 interventions, unlike the random policy that required an average 7.

Thereby, we have proved that it is possible to build an intervention system with the capability of not only treating patients from current stress episodes, but also with the ability to predict and prevent potential stressful episodes.

As a future work, we would like to better characterize the stress signal shape. A better understanding of the stress signal behavior may help us to understanding when, and for how long to intervene, as well as better assess the effectiveness of the interventions.

We also want to address the challenges in terms of online learning in both the prediction and the intervention layers. We believe that the introduction of a priori knowledge of the patient preferences for treatments and context information is important to accelerate the intervention layer's learning process and increase each intervention success rate.

Besides, we would like to implement this system in a mobile environment to explore the system's usability by the designing and running of experimental studies.

Finally, we want to explore the effect of burden on the intervention's user experience (i.e. what is level of the user tolerance to false positives) as well as the role of user incentives to encourage the user participation and retention.

REFERENCES

- [1] M. M. Rahman, R. Bari, A. A. Ali, M. Sharmin, A. Raij, K. Hovsepian, S. M. Hossain, E. Ertin, A. Kennedy, D. H. Epstein *et al.*, “Are we there yet? feasibility of continuous stress assessment via wireless physiological sensors,” 2014.
- [2] K. E. Heron and J. M. Smyth, “Ecological momentary interventions: incorporating mobile technology into psychosocial and health behaviour treatments,” *British journal of health psychology*, vol. 15, no. 1, pp. 1–39, 2010.
- [3] B. Chakraborty and S. A. Murphy, “Dynamic treatment regimes,” *Statistics*, vol. 1, 2014.
- [4] S. A. Murphy, “Optimal dynamic treatment regimes,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, no. 2, pp. 331–355, 2003.
- [5] K. P. Timms, D. E. Rivera, L. M. Collins, and M. E. Piper, “A dynamical systems approach to understand self-regulation in smoking cessation behavior change,” *Nicotine & Tobacco Research*, p. ntt149, 2013.
- [6] S. Robinson, S. Perkins, S. Bauer, N. Hammond, J. Treasure, and U. Schmidt, “Aftercare intervention through text messaging in the treatment of bulimia nervosa feasibility pilot,” *International Journal of Eating Disorders*, vol. 39, no. 8, pp. 633–638, 2006.
- [7] L. G. Jaimes and A. Raij, “Forecasting biorhythms for preventative stress interventions,” in *Proceedings of the ACM Workshop of Biological rhythms and technology, at CHI 2014*. ACM, 2014.
- [8] E. Ertin, N. Stohs, S. Kumar, A. Raij, M. al’Absi, and S. Shah, “Autosense: unobtrusively wearable sensor suite for inferring the onset, causality, and consequences of stress in the field,” in *SenSys*, 2011, pp. 274–287.
- [9] H. Lu, D. Frauendorfer, M. Rabbi, M. S. Mast, G. T. Chittaranjan, A. T. Campbell, D. Gatica-Perez, and T. Choudhury, “Stresssense: Detecting stress in unconstrained acoustic environments using smartphones,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 351–360.
- [10] E. Ertin, N. Stohs, S. Kumar, A. Raij, M. al’Absi, and S. Shah, “Autosense: unobtrusively wearable sensor suite for inferring the onset, causality, and consequences of stress in the field,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2011, pp. 274–287.
- [11] K. Plarre, A. Raij, S. M. Hossain, A. A. Ali, M. Nakajima, M. Al’absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott *et al.*, “Continuous inference of psychological stress from sensory measurements collected in the natural environment,” in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*. IEEE, 2011, pp. 97–108.

- [12] L. G. Jaimes, J. Calderon, J. Lopez, and A. Raij, "Trends in mobile cyber-physical systems for health just-in time interventions," in *IEEE - SoutheastCon, Fort Lauderdale FL*, 2015.
- [13] L. G. Jaimes, M. Llofriú, and A. Raij, "Calma, an algorithm framework for mobile just in time interventions," in *IEEE - SoutheastCon, Fort Lauderdale FL*, 2015.
- [14] L. Jaimes, M. Llofriú, and A. Raij, "A stress-free life: Just-in-time interventions for stress via real-time forecasting and intervention adaptation." *ICST*, 11 2014.
- [15] L. G. jaimes, M. Llofriú, and A. Raij, "Preventer, a selection mechanism for just-in-time preventive interventions," *T. Affective Computing*, 2015.
- [16] L. Jaimes, I. Vergara-Laurens, and A. Raij, "A survey on incentive techniques for mobile crowd sensing," *Internet of Things Journal, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.
- [17] L. G. Jaimes, I. J. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *PerCom*, 2012, pp. 103–108.
- [18] L. G. Jaimes, I. J. Vergara-Laurens, and A. Chakeri, "Spread, a crowd sensing incentive mechanism to acquire better representative samples," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom 2014 Workshops, Budapest, Hungary, March 24-28, 2014*, 2014, pp. 92–97. [Online]. Available: <http://dx.doi.org/10.1109/PerComW.2014.6815171>
- [19] L. G. Jaimes, I. Vergara-Laurens, and A. Raij, "A crowd sensing incentive algorithm for data collection for consecutive time slot problems," in *Communications (LATINCOM), 2014 IEEE Latin-America Conference on*, Nov 2014, pp. 1–5.
- [20] K. Plarre, A. Raij, S. M. Hossain, A. A. Ali, M. Nakajima, M. al' Absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott, D. P. Siewiorek, A. Smailagic, and L. E. Wittmers, "Continuous inference of psychological stress from sensory measurements collected in the natural environment," in *IPSN*, 2011, pp. 97–108.
- [21] A. Rodgers, T. Corbett, D. Bramley, T. Riddell, M. Wills, R.-B. Lin, and M. Jones, "Do u smoke after txt? results of a randomised trial of smoking cessation using mobile phone text messaging," *Tobacco control*, vol. 14, no. 4, pp. 255–261, 2005.
- [22] P. Derler, E. A. Lee, and A. L. Sangiovanni-Vincentelli, "Modeling cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2012. [Online]. Available: <http://dx.doi.org/10.1109/JPROC.2011.2160929>
- [23] A. A. Stone and S. Shiffman, "Ecological momentary assessment (ema) in behavioral medicine." *Annals of Behavioral Medicine*, 1994.
- [24] N.-S. Joo and B.-T. Kim, "Mobile phone short message service messaging for behaviour modification in a community-based weight control programme in korea," *Journal of Telemedicine and Telecare*, vol. 13, no. 8, pp. 416–420, 2007.
- [25] M. G. Newman, A. Consoli, and C. B. Taylor, "Computers in assessment and cognitive behavioral treatment of clinical disorders: Anxiety as a case in point," *Behavior Therapy*, vol. 28, no. 2, pp. 211–235, 1997.

- [26] M. Qian, I. Nahum-Shani, and S. A. Murphy, “Dynamic treatment regimes,” in *Modern Clinical Trial Analysis*. Springer, 2013, pp. 127–148.
- [27] N. Sharma and T. Gedeon, “Objective measures, sensors and computational techniques for stress recognition and classification: A survey,” *Computer Methods and Programs in Biomedicine*, vol. 108, no. 3, pp. 1287–1301, 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.cmpb.2012.07.003>
- [28] F. Horvath, “An experimental comparison of the psychological stress evaluator and the galvanic skin response in detection of deception.” *Journal of Applied Psychology*, vol. 63, no. 3, p. 338, 1978.
- [29] L. He, M. Lech, N. C. Maddage, and N. Allen, “Stress detection using speech spectrograms and sigma-pi neuron units,” in *Natural Computation, 2009. ICNC’09. Fifth International Conference on*, vol. 2. IEEE, 2009, pp. 260–264.
- [30] J. Zhai and A. Barreto, “Stress recognition using non-invasive technology.” in *FLAIRS Conference*, 2006, pp. 395–401.
- [31] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013. [Online]. Available: <http://dx.doi.org/10.1109/SURV.2012.110112.00192>
- [32] A. A. Ali, S. M. Hossain, K. Hovsepian, M. M. Rahman, K. Plarre, and S. Kumar, “mpuff: automated detection of cigarette smoking puffs from respiration measurements,” in *The 11th International Conference on Information Processing in Sensor Networks (co-located with CPS Week 2012), IPSN 2012, Beijing, China, April 16-19, 2012*, 2012, pp. 269–280. [Online]. Available: <http://doi.acm.org/10.1145/2185677.2185741>
- [33] L. G. Jaimes and V. Torra, “On the selection of parameter m in fuzzy c-means: A computational approach,” in *Integrated Uncertainty Management and Applications*. Springer, 2010, pp. 443–452.
- [34] L. G. Jaimes and F. V. Riveros, “Modelos clásicos de recuperación de la información,” *Revista Integración*, vol. 23, no. 1, 2005.
- [35] S. Bubeck and N. Cesa-Bianchi, “Regret analysis of stochastic and nonstochastic multi-armed bandit problems,” *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012. [Online]. Available: <http://dx.doi.org/10.1561/22000000024>
- [36] H. Lei, A. Tewari, and S. A. Murphy, “An actor-critic contextual bandit algorithm for personalized interventions using mobile devices,” 2014.
- [37] R. Weber *et al.*, “On the gittins index for multiarmed bandits,” *The Annals of Applied Probability*, vol. 2, no. 4, pp. 1024–1033, 1992.
- [38] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [39] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998, vol. 1, no. 1.
- [40] P. Paredes, R. Gilad-Bachrach, M. Czerwinski, A. Roseway, K. Rowan, and J. Hernandez, “Poptherapy: Coping with stress through pop-culture,” 2014.

- [41] D. L. MacLean, A. Roseway, and M. Czerwinski, "Moodwings: a wearable biofeedback device for real-time stress intervention," in *The 6th International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '13, Island of Rhodes, Greece, May 29-31, 2013*, 2013, p. 66. [Online]. Available: <http://doi.acm.org/10.1145/2504335.2504406>
- [42] S. J. Fakhri and T. K. Das, "LEAD: A methodology for learning efficient approaches to medical diagnosis," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 2, pp. 220–228, 2006. [Online]. Available: <http://doi.ieeecomputersociety.org/10.1109/TITB.2005.855538>
- [43] U. Yasavur, C. L. Lisetti, and N. Rishe, "Modeling brief alcohol intervention dialogue with mdps for delivery by ecas," in *Intelligent Virtual Agents - 13th International Conference, IVA 2013, Edinburgh, UK, August 29-31, 2013. Proceedings*, 2013, pp. 92–105. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40415-3_8
- [44] R. Amini, C. L. Lisetti, U. Yasavur, and N. Rishe, "On-demand virtual health counselor for delivering behavior-change health interventions," in *IEEE International Conference on Healthcare Informatics, ICHI 2013, 9-11 September, 2013, Philadelphia, PA, USA*, 2013, pp. 46–55. [Online]. Available: <http://dx.doi.org/10.1109/ICHI.2013.13>
- [45] N. N. Nandola and D. E. Rivera, "A novel model predictive control formulation for hybrid systems with application to adaptive behavioral interventions," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 6286–6292.
- [46] R. Zurkowski and A. R. Teel, "A model predictive control based scheduling method for hiv therapy," *Journal of Theoretical Biology*, vol. 238, no. 2, pp. 368–382, 2006.
- [47] K. P. Timms, D. E. Rivera, L. M. Collins, and M. E. Piper, "Control systems engineering for understanding and optimizing smoking cessation interventions," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 1964–1969.
- [48] P. R. Deboeck and C. Bergeman, "The reservoir model: A differential equation model of psychological regulation." *Psychological methods*, vol. 18, no. 2, p. 237, 2013.
- [49] C. Bergeman and P. R. Deboeck, "Trait stress resistance and dynamic stress dissipation on health and well-being: The reservoir model," *Research in Human Development*, vol. 11, no. 2, pp. 108–125, 2014.
- [50] V. Pejovic and M. Musolesi, "Interruptme: designing intelligent prompting mechanisms for pervasive applications," in *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14, Seattle, WA, USA, September 13-17, 2014*, 2014, pp. 897–908. [Online]. Available: <http://doi.acm.org/10.1145/2632048.2632062>
- [51] B. P. Bailey and J. A. Konstan, "On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state," *Computers in Human Behavior*, vol. 22, no. 4, pp. 685–708, 2006.
- [52] E. Horvitz and J. Apacible, "Learning and reasoning about interruption," in *Proceedings of the 5th international conference on Multimodal interfaces*. ACM, 2003, pp. 20–27.

- [53] V. Pejovic and M. Musolesi, “Anticipatory mobile computing for behaviour change interventions,” in *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*, 2014, pp. 1025–1034. [Online]. Available: <http://doi.acm.org/10.1145/2638728.2641284>
- [54] —, “Anticipatory mobile computing: A survey of the state of the art and research challenges,” *arXiv preprint arXiv:1306.2356*, 2013.
- [55] H. Sarker, M. Sharmin, A. A. Ali, M. M. Rahman, R. Bari, S. M. Hossain, and S. Kumar, “Assessing the availability of users to engage in just-in-time intervention in the natural environment,” in *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14, Seattle, WA, USA, September 13-17, 2014*, 2014, pp. 909–920. [Online]. Available: <http://doi.acm.org/10.1145/2632048.2636082>
- [56] T. Murray, L. G. Jaimes, E. Hekler, D. Spruijt-Metz, and A. Raij, “A glanceable mobile avatar for behavior change,” in *Wireless Health 2013, WH '13, Baltimore, MD, USA - November 01 - 03, 2013*, 2013, p. 16. [Online]. Available: <http://doi.acm.org/10.1145/2534088.2534093>
- [57] T. Murray, D. Hardy, D. Spruijt-Metz, E. Hekler, and A. Raij, “Avatar interfaces for biobehavioral feedback,” in *Design, User Experience, and Usability. Health, Learning, Playing, Cultural, and Cross-Cultural User Experience - Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part II*, 2013, pp. 424–434. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-39241-2_47
- [58] M. E. Seligman and M. Csikszentmihalyi, *Positive psychology: An introduction*. American Psychological Association, 2000, vol. 55, no. 1.
- [59] S. Folkman, *Stress, appraisal, and coping*. Springer Publishing Company LLC, 1984.
- [60] E. A. Carroll, M. Czerwinski, A. Roseway, A. Kapoor, P. Johns, K. Rowan, and M. M. C. Schraefel, “Food and mood: Just-in-time support for emotional eating,” in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013, Geneva, Switzerland, September 2-5, 2013*, 2013, pp. 252–257. [Online]. Available: <http://dx.doi.org/10.1109/ACII.2013.48>
- [61] A. Raij, A. Ghosh, S. Kumar, and M. B. Srivastava, “Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment,” in *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, 2011, pp. 11–20. [Online]. Available: <http://doi.acm.org/10.1145/1978942.1978945>
- [62] C. R. Taylor, K. Venkatasubramanian, and C. A. Shue, “Understanding the security of interoperable medical devices using attack graphs.” in *HiCoNS*, 2014, pp. 31–40.
- [63] L. G. Jaimes, T. Murray, and A. Raij, “Increasing trust in personal informatics tools,” in *Design, User Experience, and Usability. User Experience in Novel Technological Environments - Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part III*, 2013, pp. 520–529. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-39238-2_57
- [64] M. Musthag, A. Raij, D. Ganesan, S. Kumar, and S. Shiffman, “Exploring micro-incentive strategies for participant compensation in high-burden studies,” in *UbiComp*, 2011, pp. 435–444.



- [65] L. G. Jaimes, A. C. J. Lopez, and A. Raij, “A cooperative incentive mechanism for recurrent crowd sensing,” in *IEEE - SoutheastCon, Fort Lauderdale FL*, 2015.
- [66] J. Singh, H. Knapp, and M. Demissie, “Hydrologic modeling of the iroquois river watershed using hspf and swat. isws cr 2004-08. champaign, ill.: Illinois state water survey,” 2005.
- [67] I. Nahum-Shani, M. Qian, D. Almirall, W. E. Pelham, B. Gnagy, G. A. Fabiano, J. G. Waxmonsky, J. Yu, and S. A. Murphy, “Q-learning: A data analysis method for constructing adaptive interventions.” *Psychological methods*, vol. 17, no. 4, p. 478, 2012.
- [68] S. Koenig and R. G. Simmons, “Complexity analysis of real-time reinforcement learning,” in *Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, July 11-15, 1993.*, 1993, pp. 99–107. [Online]. Available: <http://www.aaai.org/Library/AAAI/1993/aaai93-016.php>
- [69] H. Seong, J. Lee, T. Shin, W. Kim, and Y. Yoon, “The analysis of mental stress using time-frequency distribution of heart rate variability signal,” in *Engineering in Medicine and Biology Society, 2004. IEMBS’04. 26th Annual International Conference of the IEEE*, vol. 1. IEEE, 2004, pp. 283–285.
- [70] J. Choi and R. Gutierrez-Osuna, “Using heart rate monitors to detect mental stress,” in *Sixth International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009, Berkeley, CA, USA, 3-5 June 2009*, 2009, pp. 219–223. [Online]. Available: <http://dx.doi.org/10.1109/BSN.2009.13>
- [71] J. Taelman, S. Vandeput, A. Spaepen, and S. Van Huffel, “Influence of mental stress on heart rate and heart rate variability,” in *4th European conference of the international federation for medical and biological engineering*. Springer, 2009, pp. 1366–1369.
- [72] J. Lin, E. J. Keogh, L. Wei, and S. Lonardi, “Experiencing SAX: a novel symbolic representation of time series,” *Data Min. Knowl. Discov.*, vol. 15, no. 2, pp. 107–144, 2007. [Online]. Available: <http://dx.doi.org/10.1007/s10618-007-0064-z>
- [73] W. Khreich, E. Granger, A. Miri, and R. Sabourin, “A survey of techniques for incremental learning of HMM parameters,” *Inf. Sci.*, vol. 197, pp. 105–130, 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.ins.2012.02.017>
- [74] C. J. Willmott and K. Matsuura, “Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance,” *Climate research*, vol. 30, no. 1, p. 79, 2005.
- [75] S. Makridakis, M. Hibon, and C. Moser, “Accuracy of forecasting: An empirical investigation,” *Journal of the Royal Statistical Society. Series A (General)*, pp. 97–145, 1979.
- [76] C. Christodoulos, C. Michalakelis, and D. Varoutas, “On the combination of exponential smoothing and diffusion forecasts: An application to broadband diffusion in the oecd area,” *Technological Forecasting and Social Change*, vol. 78, no. 1, pp. 163–170, 2011.
- [77] J. Salojärvi, K. Puolamäki, and S. Kaski, “Expectation maximization algorithms for conditional likelihoods,” in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 752–759.


- [78] C. J. Willmott, "On the validation of models," *Physical geography*, vol. 2, no. 2, pp. 184–194, 1981.
- [79] C. Santhi, J. G. Arnold, J. R. Williams, W. A. Dugas, R. Srinivasan, and L. M. Hauck, "Validation of the swat model on a large river basin with point and nonpoint sources," 2001.
- [80] M. Van Liew, J. Arnold, and J. Garbrecht, "Hydrologic simulation on agricultural watersheds: Choosing between two models," *Transactions of the ASAE*, vol. 46, no. 6, pp. 1539–1551, 2003.

APPENDICES

Appendix A: Permission for Reuse

The permissions below are for reuse of material in chapters 2, 3, and 4 respectively.

Home Create Account Help  Live Chat



IEEE
Requesting permission to reuse content from an IEEE publication

Title: A Survey on Incentive Techniques for Mobile Crowd Sensing

Author: Jaimes, L.G.; Vergara-Laurens, I.J.; Raj, A.

Publication: IEEE Internet of Things Journal (J-IOT)

Publisher: IEEE

Copyright © 1969, IEEE

LOGIN

If you're a [copyright.com](#) user, you can login to RightsLink using your [copyright.com](#) credentials. Already a [RightsLink](#) user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.htm to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Copyright © 2015 Copyright Clearance Center, Inc. All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#). Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Appendix A (Continued)



RightsLink®

Home

Create Account

Help



Title: SPREAD, a crowd sensing incentive mechanism to acquire better representative samples

Conference Proceedings: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on

Author: Jaimes, L.G.; Vergara-Laurens, I.; Chakeri, A.

Publisher: IEEE

Date: 24-28 March 2014

Copyright © 2014, IEEE

LOGIN

If you're a [copyright.com](#) user, you can login to RightsLink using your [copyright.com](#) credentials. Already a [RightsLink](#) user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Copyright © 2015 Copyright Clearance Center, Inc. All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#).
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Appendix A (Continued)



RightsLink®

Home

Create Account

Help



Title: A location-based incentive mechanism for participatory sensing systems with budget constraints

Conference Proceedings: Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on

Author: Jaimes, L.G.; Vergara-Laurens, I.; Labrador, M.A.

Publisher: IEEE

Date: 19-23 March 2012

Copyright © 2012, IEEE

LOGIN

If you're a [copyright.com](#) user, you can login to RightsLink using your [copyright.com](#) credentials. Already a [RightsLink](#) user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Copyright © 2015 Copyright Clearance Center, Inc. All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#). Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Appendix A (Continued)



RightsLink®

Home

Create Account

Help



Title: A crowd sensing incentive algorithm for data collection for consecutive time slot problems
Conference Proceedings: Communications (LATINCOM), 2014 IEEE Latin-America Conference on
Author: Jaimes, L.G.; Vergara-Laurens, I.; Rajj, A.
Publisher: IEEE
Date: 5-7 Nov. 2014
Copyright © 2014, IEEE

LOGIN
If you're a [copyright.com user](#), you can login to RightsLink using your [copyright.com credentials](#). Already a [RightsLink user](#) or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Copyright © 2015 Copyright Clearance Center, Inc. All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#). Comments? We would like to hear from you. E-mail us at customercare@copyright.com