University of Pisa

Department of Computer Science

PhD Degree in Computer Science

Thesis:

Just-in-time Adaptive Anomaly Detection and Personalized Health Feedback

Candidate:

Parvaneh Parvin

Supervisors:

Prof.Fabio Paternò

Stefano Chessa

Academic year 2018/2019

Acknowledgment

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

Firstly, I would like to express my sincere gratitude to my advisors Prof. Fabio Paterno and Stefano Chessa for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

I greatly appreciate the support received through the collaborative work undertaken with the Prof. Kaptein at Jheronimus Academy of Data Science (JADS), University in ’s-Hertogenbosch, Netherlands, during the last year of my research. I am especially grateful to Robin and Jules for making those few months all the more interesting.

Finally, I must express my very profound gratitude to my loved ones, especially my Mum and my late Dad for always believing in me and encouraging me to follow my dreams. I will be grateful forever for your love.

Abstract

The rapid population aging and the availability of sensors and intelligent objects motivate the development of information technology-based healthcare systems that meet the needs of older adults by supporting them to continue their day-to-day activities. These systems collect information regarding the daily activities of the users that potentially helps to detect any significant changes and to provide them with relevant and tailored health-related information and quality of life-improving suggestions.

To this aim, we propose a Just-in-time adaptive intervention system that models the user daily routine using a task model specification and detects relevant contextual events that occurred in their life in order to detect anomalous behaviors and strategically generate tailored interventions to encourage behaviors conducive to a healthier lifestyle. The system uses a novel algorithm to detect anomalies in the user daily routine. In addition, by a systematic validation through a system that automatically generates wrong sequences of activities, we show that our anomaly detection algorithm is able to find behavioral deviations from the expected behavior at different times along with the category of the anomalous activity performed by the user with good accuracy.

Later, the system uses a Mamdani-type fuzzy rule-based component to predict the level of intervention needed for each detected anomaly and a sequential decision-making algorithm, Contextual Multi-armed Bandit, to generate suggestions to minimize anomalous behavior. We describe the system architecture in detail, and we provide example implementations for corresponding health feedback. To test our approach, we collected sensor data in our smart lab testbed while an actor was performing activities of daily living over a period of 2 weeks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem overview

Nowadays, the majority of people can expect to live into their 70s and beyond [3]. The world population report predicts that life expectancy at birth will rise from 71 years in 2010-2015 to 77 years in 2045-2050 [4]. The increase of the citizen's average age is currently a matter of concern, as with the growing age, a person may encounter different physical and mental impairments such as vision and cognitive problems (dementia), and may need support to continue his/her activities of daily living (ADLs).

Several controlled studies indicated that most older adults benefit from the structure in their day-to-day living and a proper daily routine can bring order and predictability to the their life and results in reducing the stress, increasing the feeling of safety and security, and helping them to sleep better [5]. Hence, a strict daily routine provides a sense of security against unknowns and also helps caregivers as they can use the routine to plan their activities. A daily routine simply sets in place the same activities at generally the same time each day. Usually, our daily activities follow a more or less stable schedule. We wake up, wash, eat breakfast, do some activities (do household, work, do some hobbies, etc.), eat lunch, do some more activities, eat dinner, relax, go to bed, sleep. When people get older and may suffer from diverse diseases, this rhythm starts to decompose. This frequently leads to changes in cognitive abilities, sleep problems, energy loss during the

day and, mood disturbances like depression.

Studies show that older people prefer to have an independent life in their own homes [6], yet, support is needed to continue their everyday living routines. So, there will be a need for providing an adequate and sustainable solution to an increasing number of elderly people that can promote healthier lifestyles. These supports ensure that important activities get done without fail, such as medication management, regular nutritious meals, daily hygiene, etc. Also, increasing the elderly autonomy and assisting them in carrying out their daily activities create the possibility to extend the time older people can live autonomously in their homes even when, busy schedules, costs, and different circumstances make it not possible for family members or caretakers to be available with elderly all the time.

On the other hand, current technologies offer plenty of interactive devices, smart objects, and sensors to detect a wide range of environmental and user-related parameters. In the field of healthcare monitoring and Ambient Assisted Living (AAL) scenarios, collecting and monitoring such parameters through available technologies is valuable to build knowledge about the context around the users, track their health in real-time, detect anomalies and significant changes in their behavior with the assumption that these anomalies may signal health-related problems. These technologies help to provide a safe and secure ambient where older adults can live independently in their homes and give them the ability to perform basic activities (e.g., bathing/showering, dressing, feeding, functional mobility, personal hygiene, and continence) of daily living, on which the ability of a person to live independently is assessed [7]. Verifying what ADLs are performed by an older adult is a decisive factor to determine what kinds and what levels of assistance are needed for an individual and whether aging in place is desirable.

Context-aware ADL assistance is a research field that specifically deals with this issue through the integration of sensing and reasoning, to deliver context-aware data that can be employed to provide personalized support in many applications[2]. As a simple example, imagine a smart home equipped with ambient sensors able to detect people's presence and the activation of household appliances. It is possible to infer the activities performed by its residents based on the sensors' signals along with other relevant aspects such as time of the day and date (e.g., a person in the kitchen during morning time while a coffee machine is on suggests that person is making breakfast). However,

it is challenging to extract relevant knowledge in real-time from information provided by sensors and devices.

A true challenge to such context-aware technology is to convert this huge amount of information into a system competent enough to detect incoming changes in users' daily routine activities and automatically trigger just in time health interventions to prevent serious health issues and to help users maintain their routine behavior. While older people generally have unique attitudes toward healthy behaviors, they often encounter difficulties when trying to maintain their day-to-day routine activities. For those motivated, but unable to adopt a healthy lifestyle, external interventions can be very effective [8]. Besides, each individual is unique and this means that some designed interventions may fail to satisfy a person's needs [9]. So, there is a need for personalized suggestions.

Some literature reviews identified that the most highly valued systems, according to the elderly themselves and the caregivers, are remote care systems and reminders [10]. To this aim, remote care applications should be available to support remote assistance and help experts and caretakers to keep track of the overall health condition of older adults and provide them with real-time feedback and remote support. Nowadays, several contributions are focusing on older adults because of the numerous health care needs and addressing the behavior deviations and health feedback. However, technologies and solutions for detecting "unusual" behavior in the user routine daily living activities based on the user's plan are still in their infancy [11, 12], as well as tools for analyzing such behavior and acting consequently when certain conditions are met. Some systems can detect the anomalies, but they need training data or are narrowed by an offline method [13] for detecting anomalous behavior or suffer from high false alert rates because they do not include the effect of other related contexts (user activity, the criticality level of that activity for each individual) and they do not consider the personalized feedback that fit well into a users' routine.

It is thus fundamental that data analysis, both for short-term (alerts generation) and long-term (behavioral analysis) purposes, can be performed taking into account the specificities of the context where the system is used to fit the particular needs, requirements and daily routines/tasks of the considered user. In the same way, the approach to behavior analysis and especially aimed at finding the unusual behaviors, should consider that different people may have different habits (e.g., the time they get up or go to

sleep, the number and time of meals between various users, their special medical care they need, etc.). Hence, the need to automatically detect behavior and deliver tailored treatment just in time is not well satisfied with existing models of health care.

There is a pressing need to develop comprehensive healthcare-related approaches to detect the behavioral anomalies and encourage the elderly to change their anomalous behavior towards a healthy lifestyle and provide support to the family, caregivers and the elderly themselves [14]. Added to this work, quality of life of the older adults in their home can also be improved by helping them in their daily life, using automation systems. This urgent need, represent the target areas of the thesis, which aimed at developing a systemic solution to prolong the functional capacity of the older adults and to support them in preserving an independent and healthy lifestyle in their homes rather than through more expensive hospitalization solutions. We address these needs by building a personalized and online system that unobtrusively monitor the older adults' daily activities, models the user routine, detects the behavior deviations in a just-in-time manner and supports implementations for acting upon these deviations in real-time.

1.2 A Motivating Scenario

To illustrate the motivation of this research, we describe a real-life daily routine activity of an older adult. Sarah, 76 years-old, is alone and suffering from Mild cognitive impairment (MCI). She usually wakes up around 7 a.m. She does her morning hygiene and takes red medicine for her Cardiovascular disease before breakfast, in no particular order. She generally has breakfast between 8 and 9 a.m. Then, she has to do the recommended 30-minute exercises around 10:30 a.m. She takes her lunch somewhere between 1:00 and 2 p.m. She should take Blue medicine immediately after lunch. After that, she relaxes by reading a newspaper or listening to the radio. Then, she keeps herself busy with some household tasks (e.g., calling relatives, working on the computer, washing dishes, etc.). She dines about 7:30 p.m. and goes to sleep at about 10 p.m. On average, she uses the bathroom 10 times a day, and she rests anytime during her daily activities.

Furthermore, she recently had problems performing her daily hygiene and forget to

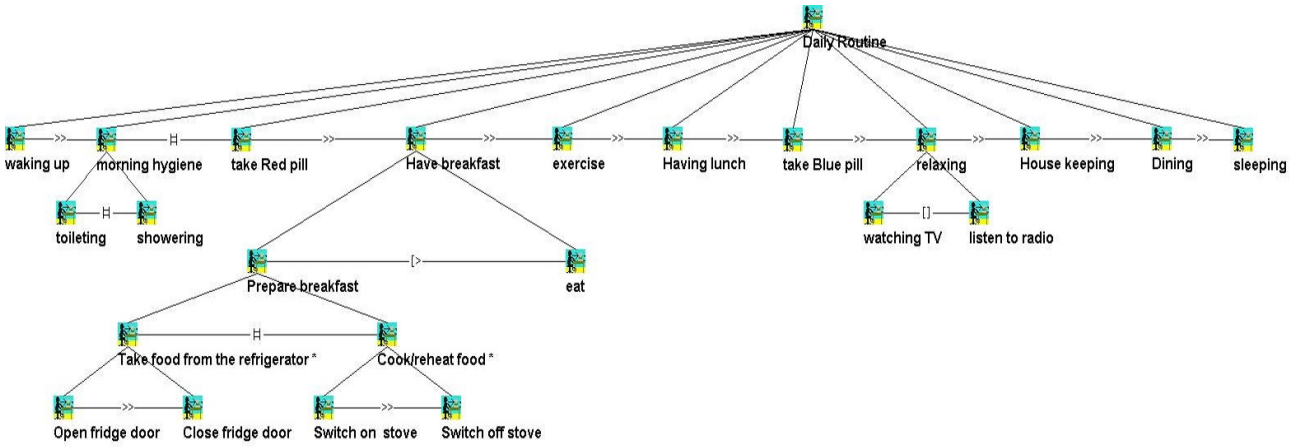


Figure 1.1: Task Model

take her pills. Her daughter Susan, who lives nearby, takes care of her mother but, due to her work and family commitments, she is not able to provide full-time assistance to Sarah. Susan is worried about Sarah's health, and she is considering whether to institutionalize Sarah for providing her long-term care. Unfortunately, the closest long-term care facility is about 50 kilometers away from the city where Susan and Sarah live. Besides, Susan has many concerns about the psychological impact on her mother of being moved from the house where Sarah has lived her whole life. Therefore, Susan is looking for a technical solution that could soften the impact of dementia symptoms on Sarah's daily life, improve her mother's quality of life and avoid the necessity to institutionalize her into a long-term care facility.

From Sarah's routine description, it is possible to derive different scenarios in which Sarah can perform her tasks differently from the expected one. In particular, since the activities that are described concern a set of task-related aspects (e.g., temporal relationships, location, order), it is possible to infer a number of corresponding types of task-related anomalies affecting elderly people such as i) unusually frequent activities (e.g., too many visits to the toilet); ii) violations of task order relationship (taking medicine before having a meal); iii) missing task (the user forgot to turn off the gas); iv) violation of the task time (e.g., take a pill in a time different from the one Prescribed). Thus, the framework for elderly monitoring is based on a task model that describes the expected daily activities from the user viewpoint (e.g., waking up, having breakfast, taking medicine, etc.), which can be used to compare with the actual behavior to detect possible deviations. Also, if the elderly routine changes due to some circumstances, the

task model should be updated accordingly with the same method used to build it, by involving the elderly themselves or their caregivers (or family members).

In our model, a situation is said to be anomalous when any behavioral changes (e.g., going to bed late, forgetting to take the medicine, delay in having dinner, visiting the bathroom frequently, etc.) occur. These anomalies may show early signs of health-related issues. The goal of our system is to detect a current change in users' daily activity and subsequently suggest automated health feedback by aggregating all the information about user behavior and the user context.

1.3 Main Contributions

The main contributions of this thesis are presented as follows:

- A novel architecture that integrates three main components: activity recognition, anomaly detection and persuasive intervention generation systems and increase the effectiveness of the interventions by personalizing them based on the user preferences. Although, there are different studies that have investigated different methods to develop each of these three main components (e.g., for AR [16, 17], Anomaly detection [18] and persuasion technology [9]), but there are few researches that address the integration and personalization of all these components together which is very important in remote care applications.

  In addition, the architecture also supports the implementation of the all 5 just-in-time Adaptive Intervention (JITAI) components (i.e., decision points, intervention options, tailoring variables, and decision rules, proximal outcome, presented in [203]) in a *personalized* way. The following is a brief explanation about novel components in the architecture.

  - Modeling domain knowledge (user routine) at different levels of abstraction. Common and generic activity knowledge can be modeled at the abstract level as an activity class described by a number of properties. These properties describe the types of objects that can be used to perform the activity. In this way, activity models can be created without the requirement of large amounts

of observation data and training processes. in addition, the user behavior model serves as a personalized knowledge base to identify the deviations in the user behavior.

  - Increasing the accuracy of activity recognition module by decreasing the volume of data associated with the user behavior through subscribing events related to each individual to the Context Manager (see section 3.2.1.4), a middleware software, in order to be notified when the expected events related to the user daily activity occur in the context.
  - Using a novel algorithm to detect the deviations in the user daily activity that outputs the anomalous task accompanying with the anomaly label.
  - Using a fuzzy rule-based system to integrate the deviation analysis and the persuasion module. This system is also responsible to identify the true anomalies (which in turn, reduces the false alarms) and to detect the level of intervention needed for each user.
  - A persuasion module that supports configuration of interventions targeting specific anomalous behavior in compliance with the JITAIs framework presented in [203]. Besides our design mechanism utilizes the reinforcement learning methods for optimization/personalization of intervention delivery in real time by modeling it as a contextual bandit problem. Benefiting from contextual multi-armed bandit formalization, the proposed personalization method puts the proposed approach beyond rule-based systems, as it is able to adapt itself according to continuously changing contexts and personal variables including both long-term (e.g., past performance, habit strength, preferences, etc.) and short-term parameters (e.g., anomalous behavior, location, time, etc.).

- A novel algorithm that detects real-time anomalies in the user daily routine and provides detailed information about the anomalies detected.

  Our online algorithm addresses the issue of detecting possible deviations from "expected users routines" (i.e., the activities that are important to be monitored for the user), that has been modeled at the different level of abstraction. Contrary to most of previous researches that label the deviations just as “normal or “abnormal”

[22], in our algorithm the deviations are classified in 11 categories considering the activity per se, place, order, time, the duration of the performed activity and more importantly the severity degree of the activity, just as, the consequence of not taking a pill could be more severe than just forgetting to take a shower. Thus, the output of the algorithm is the anomalous task along with the reason behind this anomalous behavior. This classification of anomalies makes our system transparent not only for the caregivers and experts but also for the elderly themselves, because the anomaly labels embedded the reason of the anomalous behavior in itself (e.g., “early-difference-time” that shows the task has been performed earlier than expected.). Also, the algorithm ability of classifying the anomalies, helps the further analysis of the decision-making process for intervention selection.

- Developing a platform that supports the implementation of "personalized just-in-time adaptive health intervention" systems. This platform utilizes the user context (sensor readings, anomalous behaviors and intervention level) to generate a set of personalized suggestions using contextual Multi-armed Bandit (cMAB) formalization which operationalizes the principles of behavior change theories. To integrate the deviation analysis module and the persuasion module, we applied a fuzzy rule-based system that identifies the true anomalies and the intervention level based on the detected deviations in user behavior. These interventions level are used to generate personalized and adaptive intervention messages.

- Set up a single experiment test-bed that employs sensors and appliances to accurately record movements and the activities of our case study for 2 weeks. Furthermore, to emphasize on the inaccuracy of simulation analysis, we compare our real-time collected data to our synthetically simulated data. The acquired results indicate that the difference between the accuracy is related to the missing data from sensors.

1.4 Content of the Thesis

In total, there are seven remaining chapters. Chapter.2 deliberates about the literature review performed concerning i) modeling and recognizing activities of daily living for

elderly people; ii) detection the abnormal behavior and iii) the persuasive technologies in use. Each part is closed by a discussion concerning the position of this thesis with respect to whatever was argued.

Chapter.3 describes the architecture of the developed system. In this chapter, we preliminary introduce the Just-In-time Adaptive Intervention systems (JITAIs), and we explain why our work goes under the classification of these systems. Afterward, we introduce the architecture of our system and discuss its different components that aimed at verifying the user daily activity, detecting anomalies in their daily behavior and adapting the just in times personalized health suggestion to the user behavior. The whole idea of the system architecture and method was presented at and published in the 14th International Conference on Intelligent Environments (IE) and was the winner of the IE'18 Best Student Paper Award.

Chapters 4 and 5, describe the main components of the system to detect the anomalies in user behavior and to generate personalized health feedback. These chapters start with a shorter introduction to justify their relevance and distinction in contrast to the previous chapter. After such an introduction, the core of the chapter will be presented, closed by a simulation to validate the discussed method and a summary concerning the main contribution and results of each chapter.

In detail, chapter.4 provides details about the novel algorithm for detecting the anomalies in the user behavior along with the evaluation result by the simulation of the short-term and long-term user daily activity. Part of this chapter was published in the proceedings of the ACM on Human-Computer Interaction (EICS), 2018 [15].

On the other hand, chapter 5 presents the persuasion technology and the personalized decision-making process used for delivering the persuasive interventions to the older adult including the validation by simulation of the user behavior. Besides, a rule-based fuzzy component has been presented that integrates the two main modules of the system (i.e., deviation analysis module and persuasion module) and identifies the intervention level needed for each individual. Part of this chapter was published in the Journal of Ambient Intelligence and Smart Environments 11.5, 2019 [23].

Chapter.6 will present a description of an on-going experiments in our lab testbed to reinforce the first results.

Chapter.7 will summarize discussions and results from all the previous chapters and put them in relation both to ongoing work and to possible future steps that might warrant further research. The added appendix sections contain a more detailed view of some aspects covered in chapters 4 and 5.

1.5 Publications

1. Parvin, Parvaneh, et al. "Personalized real-time anomaly detection and health feedback for older adults." Journal of Ambient Intelligence and Smart Environments 11.5 (2019): 453-469.

2. Parvin, Parvaneh, et al. "Real-Time Anomaly Detection in Elderly Behavior with the Support of Task Models." Proceedings of the ACM on Human-Computer Interaction 2. EICS (2018): 15

3. Parvin, Parvaneh. "Real-Time Anomaly Detection in Elderly Behavior." Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems. ACM, 2018.

4. Parvin, Parvaneh, Fabio Paternò, and Stefano Chessa. "Anomaly Detection in the Elderly Daily Behavior." 2018 14th International Conference on Intelligent Environments (IE). IEEE, 2018. (Best student paper award)

5. Manca, Marco, Parvaneh Parvin, Fabio Paternò, and Carmen Santoro. "Detecting anomalous elderly behavior in ambient assisted living." In Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 63-68. ACM, 2017.

Chapter 2

State of the Art

Several research contributions have been put forward that address the urgent need for remote healthcare applications for the elderly. These studies concern systems that provide support in three main areas: *modeling and understanding human routine behavior*, *detecting deviations in human behavior*, and *personalizing persuasive interventions addressing detected issues*. The first category focuses on collecting user data through advanced sensor technologies and modeling user behavior by defining the relations between situations and the actions that describe the user's routine. The second one focuses on detecting any significant changes in the users' routine and their health condition by considering current contextual events. Finally, the last topic focuses on intervening—using persuasive systems—to change the users' anomalous behavior. In this chapter, we discuss the research achievements in all these three categories.

2.1 Modeling and Recognizing Activities of Daily Living

For healthcare professionals, it is significant to determine the accurate health status of a remotely located patient or an aged person, so that, when there is a deviation, appropriate treatment is vetted in a timely manner. Addressing the deviations in the user behavior includes modeling the "normal" user behavior based on their daily life

routines, recognize the occupancy's behavior in the context and subsequently detect changes from the daily routine behavior.

2.1.1 Activities of daily Living

One important concept in remote healthcare applications is the monitoring of Activities of Daily Living (ADLs). Many of the studies focused on indoor ADLs in these applications[24, 25]. These activities have been classified into three major groups based on their complexity, duration, and activity type: short events, basic activities, and complex activities [26].

The former group is consists of brief-duration activities (on the order of seconds) such as postural transitions (e.g., stand-to-sit, sit-to-lie and so on) [27] which commonly have been used in rehabilitation-related applications [4]. Basic activities are the activities that have a variable duration but longer than transition actions (in the order of minutes), they are repetitive in nature and can be easily recognized using an accelerometer at the pocket and wrist position such as walking, running, sitting, etc. [28]. In the area of remote monitoring, there are some works focused on the basic activities [29, 30] but these systems detect a limited number of activities. Finally, the third group is one of the complex activities which is generally a collection of temporally related short events and basic activities. Complex activities are not as repetitive as basic activities [31]; For example, relaxing may contain four atomic activities, i.e., walk, reach the chair, sit, and drink from a cup.

Since human activities are complex and highly diverse, the goal of activity recognition is to recognize common activities in daily life [32]. However, recognizing complex human activities is still a challenging area, especially when dealing with concurrent or interleaved activities.

2.1.2 Modeling Human Behavior for Activity Recognition

Despite being a challenging problem in recent years, there has been a rapid growth of interest in modeling and recognizing complex activities. In previous research, the methods for building the activity models have been classified into data-driven and knowledge-

driven based on whether the Activity recognition model is built given historical database or rich prior knowledge [2, 33]. Data-driven activity recognition models users activity by using existing large datasets of user behaviors. They use data mining and machine learning techniques (unsupervised [34, 35] and supervised [36] techniques) to model the user activity and then uses the learned activity models (i.e., probabilistic classification, similarity or rule-based reasoning) to infer activities [16, 17]. Examples include Hidden Markov Models (HMMs) [37], dynamic and naive Bayes networks [38], support vector machines (SVMs) [39], graphical models [40] and multiple eigenspaces [41]. On the other hand, in knowledge-driven activity recognition approaches activity models to exploit rich prior knowledge to construct activity models directly using knowledge engineering and management technologies. This usually involves knowledge acquisition, formal modeling, and representation [2]. These activity recognition methods can be divided into a Description-based and Formalism-based representation method. The first category uses semantic and context reasoning to describe concepts and relationships in high-level and formal expressiveness [2, 42]. Activity models generated in Formalism-based representation method (e.g., event calculus [43] , lattice theory [44] and event trees[15]) are normally used for activity recognition or prediction through formal logical reasoning, e.g., deduction, induction or abduction [45]. As such, this method is referred to as a knowledge-driven or top-down approach.

2.1.2.1 Data Driven (Bottom-Up) Approaches

Data-driven approaches are generally robust to noisy, uncertain and incomplete data but suffer from several drawbacks [46]. This method requires large datasets for training and learning. The availability of large real-world datasets is a major challenge in the field of ambient assisted living. It also suffers from the data scarcity or the “cold start” problem.

It is difficult to apply learned activity models from one person to another since each individual performs activities in a variety of ways, which may differ noticeably from the others, an activity model may be inapplicable to other users. As such, this method suffers from the problems of scalability and reusability. As data-driven approaches generate statistical or probabilistic models, they cannot be easily understood by humans. If activity model information is needed for further applications (e.g., anomaly detection, assistance

applications) human-understandable activity models are more appropriate.

Another problem is that if annotated databases are needed (supervised learning approaches), scalability becomes a real problem since the effort required for annotation is huge and annotation methods are prone to errors. If annotated datasets are not used (unsupervised approach), activity granularity and activity semantics are lost and there is no guarantee that the resulting clusters represent routines.

Rashidi and Cook try to overcome the labeled activity problem by extracting activity clusters using unsupervised learning techniques (Continuous Varied Order Multi Threshold Method (COM)) [47, 48] but they still suffer from the cold-start problem, because data has to be collected in order to obtain activity models and train the recognizer [49, 50], and activity model generalization, since the learned activity models are personal and there are no mechanisms to generalize the extracted knowledge to other users. There are other data-driven methods on recognizing actions based on sensors, and most of them achieve much higher average accuracy for simple activities. Tapia et al.[40] use environmental state-change sensors to collect information about interaction with objects and recognize activities that are of interest to medical professionals such as toileting, bathing, and grooming.

The work of Ravi et al. [51] by using five different base-level classifiers (decision tables, decision trees, k-NNs, SVM, and Naïve Bayes) classified eight activities, including standing, walking, running, going up/downstairs, vacuuming, and teeth brushing, using the data collected from a single triaxial accelerometer. Wang et al. [52] utilized a Naïve Bayes model to recognize six daily actions (working, studying, running, sleeping, walking and shopping) for music recommendation. However, as the nature of the human activity is complex, people often perform not just a single action in isolation, but several actions simultaneously. Within an activity, the actions can manifest in a sequential, interleaved or concurrent way. The primary limitation of the above methods is that they have not seriously considered the relatedness among actions and the high-level semantics over actions.

Another drawback of data-driven approaches is that they can assign labels to the sensor observations but by virtue of lacking the underlying semantic structure, they are not able to reason about these labels. This is, however, an important requirement for assistive

systems, as they need to reason about the user actions as well as their goals and context [53, 54]. To address this problem, knowledge-driven approaches rely on rich prior knowledge from real-world observations which means follow a description-based approach to model the relationships between sensor data and activities. This allows them to reason about the user actions and context [53, 55].

2.1.2.2 Knowledge Driven (Top-Down) Approaches

The first step for knowledge-driven systems is to acquire the needed contextual knowledge. This is usually achieved using standard knowledge engineering approaches. Afterward, knowledge structures will be computationally modeled using a formal knowledge representation formalism [2, 56]. Depending on the nature of the acquired knowledge, different approaches can be distinguished. Knowledge engineering and knowledge management techniques are used to carry out the classification process [33, 57–59].

knowledge-driven systems have the advantages of being semantically clear and logically elegant. Some researchers use logic-based approaches for activity recognition. There exist several logical modeling methods and reasoning algorithms in terms of logical theories and representation formalisms. One thread of work is to map activity recognition to the plan recognition problem in the well studied artificial intelligence field [45, 60].

Bouchard et al. in [45] proposed a formal framework for the recognition process based on lattice theory and action Description Logic. This framework minimizes the uncertainty about observed users' activity by bounding the plausible plans set.

One type of knowledge-driven approach is symbolic models for plan synthesis [61, 62]. Such models describe user behavior in terms of preconditions and effects. These rules are later used to generate all valid execution sequences of human behavior. Such approaches have the advantage of generating execution sequences that do not appear often in the training data. Another advantage of generating models from symbolic descriptions is in situations where it is difficult to come up with sufficient amounts of training data, such as when modeling the behavior of cognitively impaired patients [53].

Semantic-based models [63, 64] have gained attention in recent years for addressing complex activity recognition problems. On the other hand, the most popular modeling

paradigm might be that of the graphical models, which include techniques such as hidden Markov models, Bayesian networks, and conditional random fields [16, 65]. While these graphical model-based approaches are capable of managing uncertainties, they are unfortunately rather limited in characterizing rich temporal relationships among activities.

Behavior recognition can be greatly levitated by a model of the task with which the behavior is associated. Task analytic methods can be used to describe normative human behavior [66]. The resulting models represent the mental and physical activities human operators performed to reach their goals. In most approaches to activity recognition, the task model is first defined according to the particular monitoring objectives, and an inference technique is then developed to map interactions with the task model. Such models (e.g., GOMS [67], AMBOSS [68] , EOFM [69], HAMSTERS [70], ConcurTaskTrees [71]) can represent the various activities which users perform to reach their goals. These models are often hierarchical: tasks are decomposed into more detailed tasks until they reach elementary tasks.

Task models have also proven to be of great help for designing and assessing the training program [72], modeling erroneous behavior to improve the safety of human-interactive systems based on where and how human behavior diverges from a task model [20], generation of scenarios to be tested over the applications [73], describing human behavior abstractly and mapped this abstract expression to humanoid robots [74], identifying possible usability issues [75], find and correct human factors issues in automated systems and generate erroneous human behavior that is a factor in the failure of complex, safety-critical systems [76], addressing human intent inference by employing a hierarchical task model [77].

According to task analysis humans, tend to decompose complex tasks into more simple ones until an atomic unit, the action, has been reached. Moreover, tasks are performed not only sequential but also decision making and interleaving task performance are common to humans. Therefore the basic concepts incorporated by most modeling languages are hierarchical decomposition and temporal operators restricting the potential execution order of tasks (e.g., sequential, concurrent or disjunct execution).

An appropriate modeling notation, accompanied by a tool, supports the user in several

ways. First, task-based specifications have the asset of being understandable to stakeholders. Thus modeling results can be discussed and revised based on feedback. Second, task models can be animated which even fosters understandability and offers the opportunity to generate early prototypes. Last but not least, task-based specifications can also be used in design stages to derive lower-level models such as HMMs [78]. For activity recognition, task modeling can be employed to specify knowledge about the behavior of users in an understandable manner which may be inspected at a very early stage.

Burghardt et al. propose the synthesis of HMMs from different symbolic descriptions of the user's activity. They investigate different kinds of description formalisms such as task models[79] and STRIPS [80]. They demonstrate that it is possible to create the state space for probabilistic models from such descriptions [81]. This model allows the generation from probabilistic models of reusable descriptions of human behavior. However, the usage of HMMs for exact inference requires the state space to be expanded prior to the inference. This limits the state space size and prevents the state space from being infinite and they did not discuss the action duration as well. While the approach would, in general, allow the use of the geometrical distribution to model action durations, the authors omit the discussion of action durations.

Yi et al. used the Markovian Task Model that tries to model the relationship and dependencies between several separate actions that build a complex activity. Markov model is a stochastic model that assumes the Markov property. The task execution is just navigation through a series of stages that lead to the end state. For this task, the graphical model is in the form of a Bayes Net. Such a network is a suitable tool for this class of problems because it uses easily observable evidence to update or infer the probabilistic distribution of the underlying random variables. A Bayesian net represents the causalities with a directed acyclic graph, its nodes denoting variables, and edges denoting causal relations. In their work, they evaluate their approach for human task recognition by performing a peanut butter and jelly sandwich experiment. In this experiment, Yi and Ballard track the eye movement to discover the object on which the eye is fixated and from that to infer the performed subtask. The results from this experiment showed that the system is able to recognize human behavior even if the sensor data is noisy [82].

In, Giersich et. al. an extension of ConcurTaskTree-s notation (CTT-notation) [79], a hierarchical task model was presented to specify the probabilities of the next possible

action during the execution that allows a transformation into Markov models, which are widely used in Ambient Intelligence. The system, having recognized the current user state, can then make use of the model in order to adjust the probabilities of the next possible state. They have the disadvantage of very large specifications. Task models can specify the same information more compact. The transformation algorithm allows us to generate an initial version of a Markov model from a task model that can be adapted in further development [78].

There are some hybrid approaches in the literature that aims at taking advantage of the features of both data and knowledge-driven modeling processes, fusing them in a single modeling approach. Ontology is often integrated into knowledge-driven methods [83]. They adopt ontology-based approaches that allow a commonly agreed explicit representation of activity definitions independent of algorithmic choices, thus facilitating portability, interoperability, and reusability. In this way, they address model incompleteness and multiple representations of terms. However, most of the studies use ontologies as mapping and/or categorization mechanisms [84, 85].

Chen et al. present an ontology-based hybrid approach to activity modeling, where learning techniques are developed to learn specific user profiles [86]. The presented research is the first step to implement dynamic knowledge-driven activity models, but the approach is limited to learn descriptive properties such as the time and duration activity is performed or the concrete order of its constituent actions. Salguero et al. propose that the ontology automatically generates the features of the ADL classifier for behavior recognition [87] however, the number of features to consider grows exponentially with every expansion process which degrades the performance of the methodology.

2.1.3 Position of This Thesis with Respect to the Activity Recognition Literature

Nowadays, In most of the researches, activity recognition is often based on machine learning over data, map a user's features into a class or a score without exposing the reasons why. This is problematic not only for lack of transparency but also for possible biases inherited by the algorithms from human prejudices and collection artifacts hidden in the training data, which may lead to unfair or wrong decisions [88]. In addition, ma-

Table 2.1: Activity classification (modeling) approaches [2]

	knowledge driven [16, 53, 61–65, 78, 78, 81–87]			**data driven** [34–41, 51, 52]	
	mining-based	logic-based	ontologie-based	generative	discriminative
Modeling type	HMM, DBN, SVM, CRF, NN.	Logical formula:, e.g., plans,lattices, calculus event trees/task models	HMM , DBN, SVM, CRF, NN	Naïve Bayes, HMM, LDS, DBNs	NN, SVM, CRF, Decision Tree
Modeling mechanism	Information retrieval and analysis	Formal knowledge modeling	Un/Supervised learning from datasets	Un/Supervised learning from datasets	Un/Supervised learning from datasets
Activity recognition method	Generative or discriminative methods	Logical inference e.g., deduction, induction	Generative or discriminative methods	Probabilistic classification	Similarity or rule-based reasoning

chine learning approaches often require lengthy and expensive data labeling by domain experts (i.e., supervised techniques) or they require long-term context history to learn the behavioral patterns (i.e., unsupervised techniques).

Instead, our approach is based on the formalism-based representation method [15], which does not need a timely and costly context history. Our approach is under the logical approach category in nature, in that it uses a task model description notation (i.e., ConcurTaskTrees (CTT) language [79]) for specifying structures and relationships among the user activities that the user has to perform to achieve a goal. It gives us the possibility to obtain a task model starting from an informal description of the system. The hierarchical structure of this specification has two advantages: it provides a wide range of granularity allowing large and small task structures to be reused, it enables reusable task structures to be defined at both low and high semantic levels.

In fact, the compelling feature of our approach is that it can model domain knowledge at different levels of abstraction. Common and generic activity knowledge can be modeled at the abstract level as an activity class described by a number of properties. These properties describe the types of objects that can be used to perform the activity. In this way, activity models can be created without the requirement of large amounts of observation data and training processes.

On the other hand, the specific way of a user performing an activity can be modeled at the lower levels of a corresponding abstract activity. It will consist of the specific information related to how the activity is performed, e.g., objects used and the sequential order they are used in. This provides a mechanism for accommodating the personal nature of an individual user's ADLs. The generated user-specific activity models also referred to as user profiles, can later be used for the realization of personalized ADL assistance. Given that most ADLs are daily routines with abundant prior knowledge of their likely patterns stemming from medical observations and psychological behavioral studies [89], it is possible to manually construct conceptual activity models, The creation of user activity profiles is then equivalent to creating activity instances in terms of a user's preference and their manner of performing ADLs. Hence it is straightforward to undertake and also scalable to a larger number of users and activities. Subsequently, tasks in the task model associated with one or more events gathered in the context. This association allows for using the semantic information contained in task models to analyze

the sequence of corresponding associated events: when the event(s) associated with a task occurs, then the corresponding task is considered completed. Another advantage of our approach is

Most of the proposed technique for activity recognition is more accurate in the presence of a lesser amount of information about the subject, as opposed to other techniques such as statistical analysis. The large volume of data associated with user behavior is one of the obstacles to achieving activity recognition in real-time. We overcome this obstacle by subscribing interesting events related to each individual to the Context Manager (see section3.2.1.4), a middleware software, in order to be notified when the planned events related to the user daily activity occur in the context [15].

2.2 Abnormal Behavior Detection

Besides behavior modeling, detecting behavior changes (anomaly detection) is another crucial and challenging task. The general definition of an anomaly is a deviation from usual behavior. The concept of "behavior" differs from one application to another. For example, for daily in-home activities, behavior is defined as the user's usual activities in his/her daily routine [90]. In this case, anomaly detection is the process of detecting behavioral "changes" of occupancy's usual lifestyle pattern [18].

In some healthcare applications [91, 92], an anomaly can be considered any abnormal reaction of a patient to the treatment. So, based on the anomaly some need rapid intervention from doctors and others only require careful follow up of the patient. Hence, we see that the decisions too are different in each case and depend on the application and the degree of the anomaly.

A survey [93] about anomaly detection approaches in smart homes basically identifies two ways to detect behavioral changes: profiling and discriminating. The former is modeling normal behavior and considering values that do not comply with the model as anomalies while the latter learn anomaly data from historical data and search for similar patterns from incoming data to find anomalies. Profiling strategy is more realistic as anomaly data is rarely seen (scarcity) in real life, to provide learning sample instances for the classifier [94].

2.2.1 Nature of input data

A key aspect of any anomaly detection technique is the nature of the input data. Input is generally a collection of data instances (e.g., events) that can be described using one or multiple attributes and each attribute can have a different type such as binary, categorical or continuous. The nature of the data attribute determines the applicability of anomaly detection techniques. For example, the nature of attributes would determine the distance measure to be used in the nearest neighbor based techniques. Input data can also be categorized based on the relationship present among data instances [95]. Most of the existing anomaly detection techniques deal with record data (or point data) [96], in which no relationship is assumed among the data instances. In general, data instances can be related to each other. Some examples are sequence data, spatial data, and graph data. In sequence data, the data instances are linearly ordered, e.g., time-series data, activity sequences, genome sequences, protein sequences. In spatial data, each data instance is related to its neighboring instances, e.g., vehicular traffic data, ecological data. When the spatial data has a temporal (sequential) component it is referred to as spatio-temporal data, e.g., climate data. In graph data, data instances are represented as vertices in a graph and are connected to other vertices with edges.

2.2.2 Nature of Output Data

An important aspect of any anomaly detection technique is the manner in which the anomalies are reported. Typically, the output is either in a scoring unit (threshold's numerical value) or a label as a categorical unit, e.g., normal or abnormal [97]. the former, assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an anomaly. Thus the output of such techniques is a ranked list of anomalies. An analyst may choose to either analyze the top few anomalies or use a cut-off threshold to select the anomalies. The later, assign a label (normal or anomalous) to each test instance. In this case, the analysts do not directly allow to make a choice about the anomaly, though this can be controlled indirectly through parameter choices within each technique.

2.2.3 Anomaly Detection Techniques

There are a number of techniques that have been used for anomaly detection, these include $statistical$: Histogram [98], Gaussian mixture model (GMM) [99], $probabilistic$: Hidden Markov Model (HMM) [100], Bayasian network (BN) [101], $MachineLearning$: Recurrent Neural Network, Fuzzy System [102–104], Support Vector Data Description (SVDD), Support Vector Machine (SVM), One Class Support Vector Machine (OCSVM), $Datamining$: clustering [35, 105–108], sequential pattern mining [109] , mining sppaning tree[22], agglomerative hierarchical clustering [11].

Statistical techniques are the most commonly used techniques for modeling human behavior in the field of healthcare. Hidden Markov Model and Bayesian networks are probabilistic reasoning concepts driven from statistical inference processes [110]. HMM is a statistical technique where the system uses a Markov process with hidden parameters. It defines a number of hidden states and observations used to model a given behavior. The hidden states represent the activities and the sensor data represents the observable output. An HMM is defined by matrices that encode possible states and probabilities of observations. State transition matrices describe the likelihood of the process of moving into a new state.

The HMM and its extensions were used for several processes including abnormality detection, and behavior prediction. Kang et al. [100] considered activities as a hierarchical structure, where main actions (preparing breakfast, preparing dinner) are composed of sub-actions (sensor firings). They applied the hierarchical Hidden Markov Model (HHMM) to find exceptional behavior patterns. A hierarchical topology of HHMM is mapped to the hierarchy of actions. Anomaly detection is based on time interval coverage of main actions to sub-actions. They assume sub-actions should last shorter than a covering main action. An important deficiency of this work is the manual classification of sub-actions (sensor firings) into main actions.

Monekosso and Remagnino [107] described a model-based behavior analysis system for assisted living in which the behavior is defined as any pattern in a sequence of observations. They use Hidden Markov Models (HMM) to model user behavior from sensed data. A model-based approach is employed for the detection of deviation from the planned behavior: given a model of the system, the predicted output generated by the model is

compared to the actual output, and any difference is a potential failure. While in our research, the identified activities (i.e. cooking, eating, etc.) corresponding to the events gathered by sensors are contributed by the caregivers and thus susceptible to errors. Meanwhile, to precisely indicate the nature of the detected anomalies, they used a further examination of a domain expert, unlike, in our work the anomaly detection algorithm detects the anomalous event along with its anomaly classification.

The main drawback of basic HMM is the lack of hierarchy in representing human behavior [106]. This problem can be solved using the Hierarchical Hidden Markov Model (HHMM) [100]. HMM, also has difficulties experienced in processing large low-level sensory data (i.e. temporal data coming from different time scales). Moreover, using HMMs as a time series prediction model needs a large and growing number of time sequences.

Fuzzy set theory is usually used for controlling automated systems in combination with other techniques such as rule-based [111] or ontology rules [106] in order to optimize contextual reasoning. One example is the context-aware model for change detection, using machine learning and statistical methods that are proposed in [112]. They have developed a system using machine learning (1-class HMM and 2-HMM) and statistical models for the inference of the anomalies in the daily activities and future behaviors. Their mathematical models for detecting anomalies are based on long-term context histories. After the abnormalities are detected, they propose a method using the fuzzy rule-based system which combines the anomalies from different domains to describe actions to be taken by the expert.

Another example is the system called Context-Aware Real-time Assistant (CARA) presented in [111]. CARA has a context-aware hybrid reasoning framework that integrates "fuzzy rule-based" reasoning with the "case-based reasoning" for pervasive healthcare. The framework was designed for home automation, continuous health monitoring and the prediction of possible risky situations. However, in spite of the context-aware reasoning, the results have been confined to limited activities in order to evaluate the system's effectiveness and efficiency in assessing the elderly.

Last but not least, Candas et al., [113] propose and validate an automatic data mining method based on physical activity measurements. The proposed method uses a fuzzy valuation function to detect abnormal human behavior in real-time conditions giving a

value (from -1 to 1) related to the abnormality identified. Abnormal human behavior is detected as an increase or decrease in physical activity according to the historical data. Historical data is used to model human behavior without assuming theoretical models, but rather information about the last physical activity levels of the user.

Meng et al., [114] by using the activation status of the sensor and by applying the probabilistic model on the two-layer hierarchy with daily activities modeled the detailed activity of the user and send it to the dynamic daily habit modeling and also to the anomaly detection module. If the similarity between the detected activity and the most similar period in the hierarchy does not reach a specific threshold, the detected activity will be defined as an anomaly and an alarm could be sent to the user. To evaluate the performance of their proposed method, they used two public datasets of fall detection and Opportunity activity recognition. Although, they are not able yet to precisely detect the complex activities which trigger the activation of the same set of sensors and they do not distinguish between the different degree of anomalies.

On the other hand, different data mining methods are used to detect anomalies in user behavior. Clustering [102] is used to group similar data instances into clusters. Clustering is primarily an unsupervised technique through semi-supervised clustering [103] has also been explored lately. Even though clustering and anomaly detection appear to be fundamentally different from each other, several clustering-based anomaly detection techniques have been developed.

Aran et al., [35] proposed a method to detect the anomalies in elderly daily behavior by monitoring the location and the outing of the user. By defining an abstract layer they create a common ground for different sensor configurations. Their approach uses cross-entropy to measure the accuracy of the predicted behavioral changes. Basically, by using k-mean clustering they discover the common behavior pattern and by comparing the sequence of the event gathered through the sensors to the user's past behavior they define the anomalies. For the evaluation of their proposed method, they used one user-written diary as ground truth and compared it with the sequences of events estimated by the abstraction layer. However, they did not investigate the anomaly type, while, in our work, we indicate different types of anomalies at the time of detection. In [104], abnormal situations like falls and fainting of the elderly were identified using K-means clustering.

Fine [22] proposes an efficient clustering technique for making a decision on the health status of a remote subject. The proposed technique uses the minimum spanning trees as part of a clustering algorithm to differentiate between normal and anomalous readings from the subject, such as, is person A in room B? If yes, for how long? The resulting information, when compared locally with similar readings from the subject during the day, can help to determine the health status of the person. The proposed technique is more accurate in the presence of a lesser amount of information about the subject, as opposed to other techniques such as statistical analysis. The large volume of data associated with user behavior is one of the obstacles to achieving anomaly detection in real-time. We overcome this obstacle by subscribing interested events related to each individual to the Context Manager, a middleware software, in order to be notified when the events related to the user daily activity occur in the context [15].

Jakkula et al. [115] describe a method to determine if anomalies can be effectively detected in smart home data using temporal data mining. They refine Allen's temporal predicates [108, 116] to specify relationships between time intervals for use in analyzing smart environment data, and apply it to the task of anomaly detection. They believe anomaly detection is most accurate when it is based on behaviors that are frequent and predictable. While they use machine learning to recognize frequent patterns, in our work we exploit task models specified with the help of relevant stakeholders to describe the planned behavior because it is not guaranteed that frequent user patterns represent the correct behavior of elder people.

The fact, that it is possible to identify human daily patterns, is an important assumption that was first introduced by [117]. They have proposed a machine learning approach to model human sequential patterns based on sequential pattern mining [109].

Weisenberg et al. [118] created an inactivity threshold function. The main idea is to mark an event as anomalous when a monitored individual is inactive for an unusually long period based on historical data for a given time of the day. The threshold is a composition of maximum inactivity data point for a given time interval and two buffers to allow slight shifts in schedule. If detected inactivity exceeds the corresponding alert threshold, an alert is issued. A disadvantage of this approach is rather a long period of anomaly detection at night, so when the user loses consciousness middle of the night, the system will recognize it only after 7 hours.

Another similar work is of Mahmoud et al. [119], who compared the binary similarity and dissimilarity measures for different days. They applied similarity measures as Jaccard Needham, Dice, Roger Tanmoto and Kulzinsky on data gathered from occupancy sensors including door and motion sensors. As the main drawback of this approach, we consider the inability to get more descriptive feedback about the anomaly. The result of their method is just binary, whether a particular day is significantly different (dissimilar) to any previous day, or not. It is impossible to acquire information about which activity at which time was anomalous.

2.2.4 Anomaly Detection for Discrete Sequence

An important aspect of an anomaly detection technique is the nature of the desired anomaly. As the nature of the data and the nature of the anomalies can differ depending on the domain, one anomaly detection technique might be efficient for one domain but not efficient for another one. Therefore many different anomaly detection techniques exist depending on the problem that needs to be solved. Even if the algorithms are different, the principle of the anomaly detection techniques is often the same. Since the outcome of the activity recognition system is a list of activities ordered according to their timestamps, the aim of the anomaly detection algorithms described in this section is to identify anomalous activities based on the obtained discrete sequences.

There is extensive work on anomaly detection techniques but most of them do not take the sequence structure of the data into consideration and not much work has been done on complex activities or activity sequences. A sequence is an ordered series of events. Sequences can be of different types, such as binary, discrete, and continuous, depending on the type of events that form the sequences. Discrete and continuous sequences (or time series) are the two most important forms of sequences encountered in real-life applications [120]. Discrete or symbolic sequences are ordered sets of events such that the events are symbols belonging to a finite alphabet. A discrete sequence consists of an ordered set of elements or events, recorded with or without a concrete notion of time.

For example, detecting anomalies in sequential data can be used for biology in order to detect anomalies in DNA sequences [121, 122]. The alphabet corresponds to the nucleic acid bases or amino acid bases. In that case, the sequences are long and the

normal structures are known. The idea is to spot the anomalous sequences within a long sequence knowing the normal structures in order to detect mutations of DNA sequences or diseases.

Another example is detecting anomalies in system calls [123, 124]. The alphabet is made with all the possible system calls or user commands. The anomalies are anomalous program behaviors that can be caused by a virus or an unauthorized user that wants to enter the computer system. For example, studying the sequences of system calls you can detect an attack on a user session. Anomalous banking transactions, purchases can also be identified using sequence anomaly detection techniques [125]. The sequences are the transactions. The alphabet corresponds to all the actions possible for the users. The anomalies are irregular or abnormal behaviors of customers.

There are several techniques to determine an anomaly in the data presented as a discrete sequence. Chandola et al. group such techniques into the following three categories that do not depend on the application domain: sequence-based, contiguous subsequence-based and pattern-based [120].In the following subsections, we describe several techniques that are instantiations of these three categories of anomaly detection techniques.

2.2.4.1 The Sequence-based Anomaly Detection Approaches

In this category a reference model or training database is assumed, containing only normal sequences, and a test sequence is tested against the normal reference to detect the anomalies. Several techniques are part of this group:

a. ***Kernel Based Techniques*** which use a similarity measure to compute similarity between sequences (e.g., distance function) together with existing proximity-based point-based anomaly detection techniques. The work of [120] also presented two main distance calculation techniques for discrete sequences: the Simple Matching Coefficient [126], and the length of the longest common subsequence[127].

   Two point-based anomaly detection algorithms that have been used for discrete sequences are k-nearest neighbor (KNN) based [128] and clustering based (e.g., k-Means [129], k-medoids [130], SVM [131]).

   Clustering methods use the distance defined in space to separate the data into homogeneous and dense groups (clusters). If we see that the point is not included in

large clusters, it is classified as an anomaly. K-nearest neighbor method is based on the concept of proximity. K nearest points have been considered based on certain rules, that decide whether the object is abnormal or not. A simple example of such rule is the distance between objects, i.e., the farthest object from its neighbors the more likely it is abnormal.

In [130, 132], Budalakoti et al. use the length of the longest common subsequence (LCS) as a similarity measure since it can compute the similarity between two sequences of unequal lengths. Thus, given two sequences X and Y, of lengths m and n respectively, they calculated the normalized LCS by the formula:
$nLCS = \frac{lenght(LCS)}{\sqrt{m.n}}$
They proposed a clustering-based technique in which the training sequences are first clustered into a fixed number of clusters using the k-medoid algorithm. The anomaly score for a test sequence is then computed as equal to the inverse of its similarity to its closest medoid.

Chandola et al. [127] proposed a KNN-based technique in which the anomaly score of a test sequence is equal to the dissimilarity to its kth nearest neighbor in the training data set.

b. ***Window Based Techniques*** [133], extract fixed length windows from a sequence and assign an anomaly score to each window. Finally, the anomaly score of all the windows is combined to obtain an anomaly score for the test sequence [134]. The anomaly score of the test sequence is the length of anomalous subsequences divided by the length of the whole sequence. By analyzing a short window at the time, window-based techniques try to localize the cause of the anomaly within one or a few windows. This technique is used for example for detecting intrusion in operating system call [123, 124, 135] although the way to assign the anomaly score (i.e., lookahead paris [133], comparing against a normal dictionary like using *Hamming distance*[124, 135], using a classifier such as HMM-based [136], neural networks [137, 138], SVM [139] and rule-based classifier [140]) to windows and how they combine the scores to obtain a global anomaly score for test sequence are different. one of the biggest drawbacks of the window-based techniques is that they are highly dependent on the length of the window (k) and finding an optimal value for

the k is challenging. This technique also requires a large amount of memory to store all unique windows and their frequencies.

c. ***Markovian Techniques:***

Markovian techniques assign a probabilistic anomaly score to each event based on the previous observations in the sequence. Such techniques exploit the Markovian dependencies in the sequence. It exists several variants of the Markovian techniques. For example, the fixed Markovian techniques (e.g., Finite State Automata Based Techniques) use a fixed memory k to estimate the conditional probability of observing a symbol of the test sequence [141]. The fixed memory k is the length of the subsequence that is used to calculate the probability of a symbol. The log average of the score vector is equal to the anomaly score of the given test sequence.

Two models Probabilistic Suffix Trees [142] and Interpolated Markov Models can be used to compute the variable-length conditional probability of a symbol. For example, Sun et al. [121] proposed a technique using the Probabilistic Suffix Trees ($PTSs$). A PTS is a tree that represents the Markov chain using suffix trees as an index structure. Each edge of the tree is a symbol. A subsequence is obtained in each node using the path from the root of the tree to the node. Each node has the frequency and conditional probability information. The PTS is created with the training sequences and contains only the subsequences that have a frequency or a conditional probability above a specified threshold. A likelihood measure is then calculated for a test sequence using the conditional probabilities.

Another method is the Sparse Markovian Techniques. Eskin et al. [143] used Sparse Markov Transducers (SMTs) which is similar to probabilistic suffix trees, to estimate a probability distribution conditioned on an input sequence, in which the utilization of wild cards ensures the sparsity of the input sequences.

Lee et al. [144] proposed a different sparse Markovian technique that uses a classification algorithm (RIPPER) [145] which is used to learn rules that can predict the kth symbol given the first $k-1$ symbols. During testing, the confidence score associated with a prediction is used to obtain an anomaly score for every symbol of the test sequence.

d. ***Hidden Markov model (HMM)-based techniques:***

The HMM technique are probabilistic models that transform the given input sequence from the symbol space to the hidden state space. The key assumption for HMM is that the normal sequences can be effectively represented in the hidden state space, while anomalous sequences cannot be. The HMM technique performs very poorly on public data sets. The key reason for the poor performance of HMM is that it assumes that the normal sequences can be represented with σ hidden states. Often, this assumption does not hold, and hence the HMM model learned from the training sequences cannot emit the normal sequences with high confidence. Thus all test sequences (normal and anomalous) are assigned a low probability score. On the other hand, HMMs are interpretable and theoretically well-motivated. Approaches that use HMMs for anomaly detection include [127, 146]. Florez et al. [146] calculated the best-hidden state sequence of an HMM of a normal training sequence by using the Viterbi algorithm [147], and utilized a threshold method to distinguish the normal and abnormal state transitions. The anomalous score of a test sequence is the average anomaly score of the state transition probabilities.

2.2.4.2 Contiguous Subsequence-based Anomaly Detection Approaches

Identifying anomalous contiguous subsequences in a long sequence is another important problem in the area of the anomaly detection of sequences where an activity is being monitored over a long period. A basic anomaly detection technique can be described as follows: A long sequence is divided into subsequences of fixed k-length windows and an anomaly score is calculating for each subsequence by comparing the subsequence to the rest of the windows. The subsequences with an anomaly score above a threshold given by the user are considered as anomalous subsequences. The length of k is an important aspect of these techniques. Since the length of the "true" anomaly is not known a priori, we do not know the length of k. If the length k is too small, the subsequences might have high probabilities and some anomalies will not be spotted as anomalies. On the contrary, if k is very large, the subsequences might have low probabilities and it will result in a high number of false anomalies.

There are two major techniques developed for solving this problem. The first category of techniques scores the windows differently, while the second category of techniques

addresses the time complexity of the basic technique.

a. ***Window Scoring Techniques:***

Several techniques can be used to calculate the anomaly scores of the subsequences. One possible technique for scoring the subsequences is to find how many times the subsequence is present in all the $k-$length sequences (which means how many times the subsequence is present in the long sequence). The anomaly score of the subsequence is the inverse of this number. As said previously, the length k is really important. For example, if k is too high, it will be difficult to find exact matches in the sequence. Keogh et al. [148] proposed a Window Comparison Anomaly Detection method that utilized the Compression-Based Dissimilarity (CDM) technique to assign an anomaly score to each window. If the compression of a subsequence and the long sequence has a low value, it means that the subsequence is normal. It will indicate that the subsequence matches the rest of the long sequence. On the other hand, if the compression between the subsequence and the long sequence has a high value, the subsequence will be spotted as anomalous. The subsequence will then not be similar to the rest of the sequence.

Another method was proposed by Keogh et al., called HOT SAX [149] in which subsequences extract out of the given sequence using the sliding window, and then the anomaly score of a window is calculated as equal to its distance to its k_th nearest neighbor. Distance between two sequences is measured using the Euclidean measure. One drawback of the nearest neighbor-based technique is that they involve an additional parameter, k, which needs to be set carefully, though approaches such as using the weighted sum of the distance to the k nearest neighbors to compute the anomaly score can be used to reduce the sensitivity on$k$. Other techniques prune the subsequences to reduce the execution time as the standard techniques need $O(l^2)$comparisons of subsequences, where l is the length of the long sequence. Considering that most of the subsequences tend to be normal, the subsequences that do not have an anomaly score high enough will be pruned [150]. It should be noted that this pruning method guarantees the same result as the basic technique, but can result in lower execution time. A similar approach is also applied to the domain of medical data by Lin et al. [151]. The same authors propose the use of Haar Wavelet-based transformation to make the previous technique more efficient

[152, 153].

b. ***Segmentation Based Techniques:***

Anomaly detection for contiguous subsequences based on fixed-length (e.g., length k) windows has difficulties in choosing k, especially in the case that there are different-length abnormal subsequences. If k is set to be very small, all k-length windows might appear highly probable, resulting in high false-negative rates. If k is set to be very large, all $k-$length windows might have a low occurrence probability, resulting in high false positive rates. To resolve this problem, Chakrabarti et al. [154] proposed a technique to segment a sequence into different-length subsequences based on *Shannons* Source Coding Theorem. The subsequences that need the highest number of bits are defined as anomalous. Gwadera et al. [155] designed a variable Markov chain method for sequence segmentation which can also be employed similarly to identify anomalies.

2.2.4.3 Pattern Frequency-based Anomaly Detection Approaches

In this category, anomalies have been detected using the number of occurrences of the test patterns. A test pattern is anomalous if its frequency in a test sequence is significantly different from its frequency in a sequence known to be a normal sequence. The basic pattern frequency-based technique calculates the anomaly score of a pattern as the absolute difference between the frequency of that pattern in a test sequence and the average of the frequencies of that pattern in the training sequences. The frequency of the pattern that occurs in the test sequence is normalized with the length of the test sequence. The average frequency obtains with the training sequences is calculated as the sum of each the normalized frequency of the pattern in each training sequence divided by the number of training sequences. The normalized frequency is the frequency of a pattern divided by the length of the long sequence. The problem with this method is that only the exact matching patterns in the test sequence and in the training sequences will be taken into account while calculating the anomaly score. If we are interested in a long pattern, it might be more difficult to find exactly the same pattern in the training sequences.

A basic technique to solve the above problem assigns an anomaly score to the query pattern, p, as the difference between the frequency of occurrence of p in the sequence and the average frequency of occurrence of p in the sequences in all training sequences. If the pattern is long, it is unlikely for the entire p to occur in a training sequence. Another issue is that in many domains, it is reasonable to assume that the symbols of the query pattern can occur interspersed with other symbols, and hence only considering substring matches will miss such occurrences. To address these issues, the following three variations of the basic technique have been proposed:

a. ***Substring Matching***

Keogh et al. [156] proposed a technique using the subsequences of the pattern. They count how many times a subsequence of the pattern occurs in a sequence. They determinate the largest length l of the subsequences, such that every subsequence of length l of the pattern occurs at least once in the training sequences.

b. ***Subsequence Matching:***

An issue with the previous technique is that counting the number of times subsequence of the pattern occurs as a subsequence in a long sequence is expensive. To make the formulation more tractable, Gwadera et al. [155] counted the number of all the windows containing the queried pattern as a subsequence. They divide the sequence in different windows of a fixed length, where the length is bigger than the length of the pattern. The pattern is considered present in a sequence if there is at last one window where the pattern is found as a subsequence of one of the windows. They determine how many windows of the sequence contain the pattern as a subsequence. The anomaly score of the pattern is the absolute difference between the number of windows containing the pattern in the sequence and the average number of windows containing the pattern in the training sequences.

c. ***Permutation Matching Techniques***

This technique counts the number of times any permutation of the query pattern occurs as a noncontiguous subsequence in a sequence. Gwadera et al. [157] proposed another approach as an extension to the subsequence matching technique

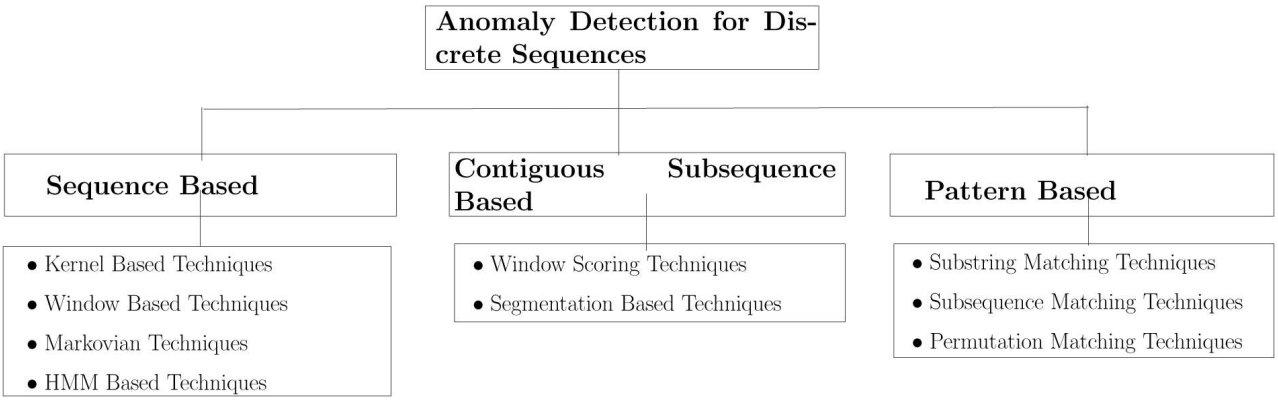


Figure 2.1: Overview of existing anomaly detection research for discrete sequences.

[155] where not only the pattern can be detected as a noncontiguous subsequence but also the order of the events (symbols) in the pattern is not important.

The biggest two issues of these techniques are *computational complexity* and *scoring of anomalies*. Computing the anomaly score depends on the length of the sequence and the length and number of sequences in a training set. So, in case of having multiple query pattern, the total time to score all the query pattern is very high.

A technique called TARZAN [158] was proposed by Keogh et al. to address this problem. in this technique, they used two suffix trees, one for the sequence, and one for each subsequence in the sequence. In this case, the complexity is linear with the length of the sequence.

Regarding the basic techniques, they assign an anomaly score to query patterns without defining which pattern is anomalous. In the case of having multiple query patterns, the top few patterns with the highest scores could be declared as anomalous. The problem is with one query pattern. Gwadera et al. [105] address this issue by scoring the query pattern using the z-score of the observed relative frequency. They assume that the relative frequency is generated from a normal distribution using sequences in the training set. A threshold on the z-score is used to determine if the occurrence of the pattern p is anomalous or not.

More recently, researchers have suggested Machine Learning approaches that most of them require all the data to have accumulated before anomalies can be identified which by modeling the expected user behavior using the task model, we have beaten this problem. Although, obstacles to achieving anomaly detection in real-time include the large

volume of data associated with user behavior and the nature of that data [159], but using our method, we overcome this obstacle by subscribing to the Context Manager (a middle-ware software) in order to be notified when the events related to the user daily activity occur in the context. This way, we just receive the relevant events from the Context Manager.

2.2.5 Position of This Thesis with Respect to the Anomaly Detection Literature

In our work, the inputs of the deviation analysis module are activity sequences and the output is a label (Normal or abnormal), which later is granulated to more specific and detailed types of anomalies. To detect anomalies in our discrete sequences (sequence of events received from Context Manager), we used the sequence-based anomaly detection techniques that compare the test data set with the task model (train set) in order to detect the anomalies. Particularly, we used a similarity measure (LCS) to compute the similarity between the receives sequences and the planned one.

Thereby, we developed an anomaly detection algorithm which compares the set of tasks enabled in the task model (according to the current execution state of the model) and the task associated with the sequence of the real events occurred in the context: if the latter one does not belong to the first one or the time and the number of occurring event is not based on the planned one, then a deviation occurs. Several types of deviations can be identified as a result of this comparison. In this work, we use a similar method [112, 113] using the fuzzy rule-based system to decide about the degree of the detected anomaly and consequently the level of intervention needed. While in [113] they just considered the user physical activity, our focus is on all the activities in the user routine behavior. In addition, our method for detecting anomalies is different from the one in [112].

2.3 Personalized Persuasive Interventions

Improving healthy lifestyle behavior is an effective strategy to increase health-related quality of life [160]. Ideally, health tools and systems for remote care should not only

help in keep tracking of one's health status but should also induce a positive change to reduce risky health-related behavior. In this regard, persuasive technology-based on pervasive computing has been proposed to improve human health and well being [161].

2.3.1 Persuasion Technology

Persuasive Technologies (PTs) are focused on influencing attitudes or behaviors. In a sense, every research or technology described has something to do with influencing attitudes or behaviors, and could, therefore, be considered persuasive technology. The concept of persuasion is based on psychological and sociological theories on behavior and technology. A combination of self-tracking, goal-setting, and feedback in automated interventions has been indicated by many to be an effective approach for increasing healthy lifestyle behavior [162]. Self-tracking is defined as the practice of systematically recording information such as behaviors, thoughts, and feelings about oneself. It encompasses collecting data and reflecting on it in order to acquire knowledge or achieve a goal, such as behavior change [163].

On the other hand, goal-setting and feedback are components that can be provided via so-called persuasive eCoaching [164] which is the use of technology during coaching to motivate and stimulate the users to change attitudes, behaviors, and rituals. In [165, 166], they describe such a persuasion technology used to design the persuasive system to positively influence health behavior change. They build their model upon earlier research by Fogg [167] which divides the persuasive components to the main 4 categories: *primary task support, dialogue support, system credibility support, and social support*. Primary task support refers to features that support the main goal that the system helps users to achieve, and this is often supported by different features from the dialogue support category, such as giving feedback, rewarding, or reminding. The most applied model is the primary task followed by dialogue support [168].

To make this model more complete for persuasive eCoaching, some coaching components that can be provided via technology can be added, namely educational coaching, goal-setting, and feedback. The integration of self-tracking and feedback technology creates a great opportunity to develop a fully automated health feedback generation. self-monitoring help us to gather and monitor the user's daily routine behavior and detect

their abnormal behavior which is more reliable than the user own memory. Moreover, by using the information gathered through smart devices, we can generate real-time personalized feedback.

2.3.2 Persuasion Strategies and Behavioral Changing Techniques Used In Interventions

The Behavioral Change Techniques(BCTs) are theory-driven methods that are used in behavior change interventions. A behavioral Change Technique is a process aiming to influence a psychological determinant (i.e., a variable in people's head). Behavior change interventions' can be defined as coordinated sets of activities designed to change specified behavior patterns [169]. There are a large number of interventions described in the literature and have been ongoing efforts to categorize interventions into taxonomies.

Michie et al. divided these interventions into 93 hierarchically clustered techniques (BCTs) into 16 groups[170]. In [171] behavioral support for smoking cessation is characterized by BCTs. Forty-three BCTs were identified that classified into four functions: 1) directly addressing motivation e.g., providing, maximizing self-regulatory capacity or skills, promoting adjuvant activities and supporting other BCTs. This taxonomy can be reliably applied to specifying BCTs in treatment protocols and reports. This paper plus another research about increasing physical activity and healthy eating [172] show that these BCTs serve five of their intervention functions: education, persuasion, incentivization, training, and enablement.

Hagger et al. refined (CALO-RE) taxonomy of Forty behavior change techniques builds on initial work on classifying psychological techniques used in intervention to change behavior, with a particular emphasis on physical activity and healthy eating [171]. The taxonomy was based on [173] initial systematic development of a taxonomy of forty-three behavior-change techniques. The CALO-RE taxonomy does not specify the theories from which the techniques are derived but define a guide to which techniques may be adopted to change physical activity behavior in interventions for the aim of identification and classification[174]. Recent research on the components of behavioral interventions has called for better intervention designs to isolate the individual components or techniques that are effective in changing the behavioral outcomes [175].

Fogg presents a new model for understanding human behavior. In the Fogg Behavior Model (FBM), behavior is a product of three factors: motivation, ability, and triggers, each of which has subcomponents. The FBM asserts that for a person to perform a target behavior, he or she must (1) be sufficiently motivated, (2) have the ability to perform the behavior, and (3) be triggered to perform the behavior. These three factors must occur at the same moment, else the behavior will not happen. The FBM is useful in the analysis and design of persuasive technologies. The FBM also helps teams work together efficiently because this model gives people a shared way of thinking about behavior change[176].

B.J. Fogg in his seminal work [167] discerns 42 strategies for persuasion and Oinas-Kukkonen [177] built on Fogg's strategies to develop 28 persuasive system design principles. The six persuasive strategies developed by Cialdini are among the oldest and most widely employed strategies [178]. These six strategies are: (1) reciprocity, the obligation you feel to repay someone when they do something for you [179]; (2) commitment and consistency, the urge you feel to be consistent with what you already agreed (or disagreed) with; (3) social proof, the safety that is in doing what others are also doing; (4) authority, the obligation we feel towards (ostensible) authority [180]; (5) liking, the tendency to be more quickly convinced or persuaded by someone we like; (6) scarcity, the desire to get things which are 'special' or limited or running out. The strategies of Fogg and Cialdini are the inspiration for many of the strategies used in persuasive technology research.

The COM-B is an abstract and simpler nature[169]. COM-B explains Behaviors based on 3 factors: capability, opportunity, and motivation. Capability is defined as the individual's psychological and physical capacity to engage in the activity concerned. It includes having the necessary knowledge and skills. Motivation is defined as all those brain processes that energize and direct behavior, not just goals and conscious decision-making. It includes habitual processes, emotional responding, as well as analytical decision-making. Opportunity is defined as all the factors that lie outside the individual that make the behavior possible or prompt it. The model not only considers that the 3 factors can individually influence the behavior, but also that there can be an influence between components of the system. For example, opportunity can influence motivation as can capability; enacting a behavior can alter capability, motivation, and opportunity.

Shelly Chaiken explains the heuristic-systematic model (HSM), which is a widely recognized communication model to show how people receive and process persuasive messages. According to these models, persuasion may occur after either intense scrutiny or extremely superficial thinking. The messages can be processed in one of two ways: heuristically or systematically. In systematic processing, the individual is careful about the content of the message. This kind of processing is more likely to be used when messages are of importance or personal significance for the recipient of the message, when the person already has strong attitudes towards the subject, and when one needs to have mental activity. The heuristic processing requires a rather modest, subjective effort. People who are not interested a lot in (or do not know much about) encouraging messages, use Heuristic (for example, statistics that do not lie, "whatever, he is an expert", or "people like me, they always have the right or saying the truth"). When a message is ambiguous, recipients usually process heuristic. This heuristic processing may influence the next systematic processing, and create attitudes consistent with the meanings and concepts derived from the initial heuristic processing. These findings support the idea that high involvement leads message recipients to employ a systematic information processing strategy in which message-based cognitions mediate persuasion, whereas low involvement leads recipients to use a heuristic processing strategy in which simple decision rules mediate persuasion.[181]

De et al, present PORTIA a computational model of persuasion that produces personalized persuasion dialogues in the well-being and healthy eating domains. Their technique is motivating and influencing the users in changing wrong behaviors or perform a given action that combines the emotional and rational techniques by considering a restricted set of goals relevant to the Healthy Eating domain. PORTIA is based on the distinction between two phases: (i) reasoning to compute the degree of importance of the various -rational and emotional- goals to the user and (ii) construct the arguments to express the strategy selected by the Reasoner [182].

2.3.3 Personalizing Persuasive Health Recommendation

Persuasive technologies play an important role in improving and effectively employing large scale, personalized interventions [176] to change behaviors. These technologies

aim to enhance the user experience by taking into account users' interests, preferences, and other relevant information and they aim to modify user attitudes, intentions, or behavior through different strategies [167, 177]. While both personalized and persuasive technologies affect user interaction and behavior, we suppose that this effect could be significantly increased if the two technologies were combined to create personalized and persuasive systems.

For example, the persuasive power of a one-size-fits-all persuasive intervention could be enhanced by considering the users being influenced and their susceptibility to the persuasion being offered [183]. Likewise, personalized technologies can be more successful in terms of user satisfaction if their services used persuasive techniques. Hence, merging personalization and persuasion has the potential to enhance the impact of both technologies.

In [184], they proposed three natural opportunities for personalization in persuasive systems: i) personalized *assistive* features that focus on monitoring and presenting information about aspects of importance to a user in an easy and simple manner. Persuasive systems are often unresponsive to the user's preferences and fail to monitor user daily activity with respect to parameters that are important for them. By incorporating personalization, which understands the desired goal and adaptively supports the user in achieving this goal, the persuasive power of the system could be leveraged. They could monitor on users' behalf, provide guidance and support, or even provide encouraging personalized feedback. An example of such tools would be a persuasive remote care application which recommends a physical activity considering the user fitness routine, exercise preferences, and location, with the aim of generating feedback that is appealing to and achievable for that user;

Health behavior interventions have, for example, been used to give step-by-step instructions to users for performing their daily activities [185, 186]. The authors in [9] proposed a COACH system for assisting individuals with dementia to wash their hands through step-by-step audiovisual prompts. For estimating the level of dementia they used a type of reinforcement learning (namely, a partially observable Markov Decision Process (POMDP)). Based on the individuals' ability to perform the task, they considered three interventions (assistance prompt with the task description, do nothing, call caregiver). The main question was the effect of the personalization for each user with

respect to the intervention choice and the step in which the user needs help to complete the task. Personalization plays an important role in designing health interventions, as the most effective persuasive and motivational strategies are likely to depend on user characteristics, behavior, and context. The POMDP, however, is likely not to work well if the dimension of the problem and the number of actions grows. Hence, the method seems infeasible for large-scale problems involving large numbers of users and intervention actions [187].

ii) personalized $messages$ adapt the content displayed to an individual based on the user's preferences, tailored to the observed contextual conditions. Health behavior interventions can be delivered at appropriate times (morning, evening), through the most appropriate medium (email, SMS), and according to the preferred frequency (daily, weekly). In this way, in the previous remote care application example, the recommendations could be delivered through contextualized *just-in-time* (JIT) messages [188], supported by appropriate language, visual style, and multimedia content;

One example of a rule-based persuasive framework applied to mobile persuasive technology is Just-in-time Adaptive Intervention (JITAI). This framework can be used to prevent negative health outcomes and promote healthy behavior by adapting the intervention to the dynamics of a user's emotional, social, and physical contexts[189]. JITAI is based on crisp decision rules that define how a given health mobile system is going to adapt just-in-time to the stored health determinants of the user.

The framework of Just-In-Time Adaptive Interventions [19] has recently been put forward to unify a number of decision making problems that arise in mobile health across a variety of behavior change domains including physical activity [190, 191], eating disorders [192], alcohol use [193, 194], mental illness [195], smoking cessation [196], obesity/weight management [197] and other chronic disorders [198, 199].

In addition, JITAI could benefit of the fuzzy rule-based system (FRBS) to model the vagueness of user behavior. In [200], authors propose the use of fuzzy rule-based systems (FRBS) to define JITAI's tailoring variables and decision rules. An FRBS is an artificial intelligence system that uses fuzzy logic (FL) to handle vagueness in a given domain knowledge through degrees of truth. Regarding behavior formalization and persuasive technology, fuzzy methods have been successfully used for persuasive online marketing

[201], as well as for the presentation personalization media adaption of online content, using, for instance, fuzzy cognitive maps [202].

iii) and personalized *strategies* focus on responding to a user's susceptibility to various persuasive techniques and methods. A core area where the type of intervention itself is adapted to a user's personality, behavior, and susceptibility to various forms of persuasion [183]. There are multiple dimensions that the remote care application could potentially personalize in this scenario, such as the credibility of the information sources, tone and style of the intervention.

Other data analytic methods emerging in computer science are designed to continually re-adapt; that is, update the JITAI decision rules to an individual over time as s/he experiences the intervention. In cases where health behavior interventions aim to encourage and support people to change their behavior toward a healthier lifestyle, exploring different strategies to find the intervention that is most effective for a single user is very important.

To this end, a more scalable approach that is currently gaining popularity is the Multi-Armed Bandit approach: the multi Armed Bandit problem provides a paradigm for sequential decision-making under uncertainty. Strategies to address this problem effectively balance *exploration* —trying out new interventions —with *exploitation* —using the intervention that we believe is best for the current user. There are five keys components of JITAIs: decision points, decision rules, tailoring variables, intervention options, and proximal outcomes. Contextual bandit algorithms can be used for personalizing JITAIs [203]. The tailoring variables, such as GPS location, calendar busyness, and heartrate, form the context. The intervention options are actions.

For instance, MyBehavior [204] is a personalized healthy lifestyle recommendation system to help users toward healthier lifestyles regarding physical activity and dietary behavior. Here the authors used a decision-making algorithm based on a Multi-Armed Bandit data analysis method to modify decision rules as user behavior changes in the course of the intervention. MyBehavior uses sensor data to suggest a frequent behavior (e.g., walking) when the person is in a particular location and life context (e.g., on the way home after work). However, it also occasionally prompts infrequent higher-energy-expending behaviors that the person does only rarely (e.g., running) to allow the data

analytic methods to learn whether the person would repeat these behaviors. If the person repeats these behaviors, the decision rule is modified so that the new, higher-energy expending behavior is recommended (instead of walking) when the person is in a particular context (e.g., after work hours). Although their approach results in maximizing the calorie loss in individuals, it still required manual entry of food photos.

Tewari and Murphy [205], introduce an approach for sequential decision making in mobile health using contextual bandit algorithms. Bandit algorithms are used to learn policies that aim to maximize an immediate response to an action. This research focuses on the speed with which an online learning method can learn optimal actions. The speed of learning the optimal actions is important in mobile health since the aim is to provide the most effective intervention support to the user as quickly as possible; they aimed to minimize user aggravation and disruption due to inappropriately-timed delivery of actions. At each time point, the state/context is observed, then the online learning algorithm selects the action, and then subsequently the reward is observed. This occurs repeatedly and the critical question is, "After T such interactions, how close are the accumulated rewards to the accumulated rewards in a setting in which we knew a priori, the optimal actions?" The authors survey a variety of algorithms and the speed at which they learn in different settings.

The MAB formalization has been used for personalized recommendation in other domains too. For example, they have been used for stress reduction [206], learning action selection for the student [187], modern service economy [207], suggesting personalized news articles on Yahoo [208] and serving advertisement in Google [209].

2.3.4 Position of This Thesis with Respect to the Persuasive Intervention Literature

In this area, the challenge is to develop intelligent applications that leverage the user's contextual information to deliver notifications without causing unpleasant interruptions to recipients. The methodologies we utilized to develop the persuasion process comprises understanding current behaviors, detecting when the older adult is deviating from the intended target, and, if so, deciding the best course of action to nudge the older adult towards the target behavior. For this to be possible, the Persuasion Module must apply

one of the 6 Cialdini's' principle persuasion techniques to promote changes.

Thus, the persuasion process derived from the JITAIs architecture, Fuzzy rule-based system and contextual multi armed-bandit, has a fundamental pattern that repeats over time is the following.

1. smart devices and sensors collect tailoring variables (i.e., the context, behavior deviations/anomalies),
2. at a given decision point (i.e., when a true anomaly is detected) do,
3. a decision rule (or policy) maps the tailoring variables into an intervention option (i.e., the persuasive intervention messages),
4. Context manager through the sensors records the proximal outcome (i.e., user feedback, interpreted as a reward. So higher is better),
5. done!

As a decision point, we use the time that a true anomaly has been detected. For intervention options, we implemented an array of possible intervention messages and as tailoring variables (i.e., information concerning the individual that is used to decide when (i.e., under what conditions) to provide an intervention (e.g., intervention level, in our case.) and which intervention to provide) we used fuzzy rule-based system to draw out the anomaly degree along with Intervention level need for that anomaly. We considered three intervention levels: No intervention and, Mild intervention (i.e., sending a notification) and High Intervention (i.e., sending alarms to the caregivers or family members). decision rules link the intervention options and tailoring variables in a systematic way. There is a decision rule for each decision point. In our case, the decision rules are our policy implemented based on the contextual multi-armed bandit formalization that decide which intervention is the best for a specific user in a specific time. Once an intervention option is chosen, a proximal outcome (i.e., reward) is obtained. our proximal outcome might be the number of steps the person walked in the 1 h following the decision point.

Chapter 3

System Architecture

In this chapter, after a brief overview of the architecture, we introduce the Just-In-time Adaptive Intervention systems (JITAIs) framework, and, we explain why our work goes under the classification of these systems. Afterward, we introduce the architecture of our system in detail and discuss its different parts that aimed at adapting the just-in-time personalized health suggestions to the user preferences. Our system is designed to close the loop between the information flow from data acquisition to personalized health intervention suggestions. We aim at verifying the users activity, detecting the user anomalous behaviors and providing the right type of support, at the right time, adapted to the dynamics of an individual's context.

The 4-layer architecture proposed gathers first sensor readings from the environment to model the user behavior using a task model specification and detects any significant changes in the users' routine in real-time. Then, it filters the result of the anomaly detection by passing the detected anomaly, the type of anomaly and the criticality level of the anomalous task trough a fuzzy logic system to find the true anomalies. Later, it gathers user feedback and knowledge from previous interactions to choose the best action knowing the user current state. The sensor readings can be used by both the activity verification system to create higher level information and by the adaptive decision process updating its reward function. In addition, the user daily behavior model serves as a personalized knowledge base for detecting abnormal behavior. All this information has been used to build a representative knowledge that will allow the system to adapt to its

users' preferences.

To address the decision-making process, we propose to implement a contextual Multi-armed Bandit (cMAB) formalization in which the "context" represents: low level information such as sensor readings, high level information such as detected currently performed activities, and the users' anomalous behavior. The chosen MAB policy will be responsible of identifying the best action knowing the user current state. Detecting the relevance of each context attribute (environment and user anomalous behavior) for each possible intervention helps in generalizing the reward function and allows the system to adapt to each individual. This architecture also enables the system to reduce false alarms by combining the user context and detected anomalies, using a Mamdani-type fuzzy rule-based system.

3.1 Just-in-time Adaptive Intervention Framework

The motivation behind using the just-in-time approach is grounded in the idea that timing plays an important role in older people's life to determine whether support provision will be beneficial. Here, the concept of timing is largely event-based, in that the answer to the question "when is the right time?" is defined by events or conditions (e.g., when the individual forgets to take medicine) rather than by clock time (e.g., at 4 p.m.). Such events/conditions are unexpected — they repeat irregularly, in a manner that cannot be fully predicted. On the other hand, JITAIs operationalize the individualization of the selection and delivery of intervention options based on ongoing assessments of the individual's internal state and context. Thus, the JITAIs system, addresses the unique and changing needs of individuals, with the goal of achieving the best outcome for each individual. Existing frameworks for the design of adaptive interventions [189, 203, 210] highlight five components that play an important role in designing these interventions:

- decision points : A decision point is a time at which an intervention decision is taken. In general, the decision points in a JITAIs might occur (a) at a pre-specified time intervals (e.g., FOCUS [195]), (b) at specific times of day/week [211, 212]; or (c) following random prompts [213]. However, in our method the decision point is a time at which a true anomaly has been detected.

- intervention options: An array of possible treatments or actions that might be employed at any given decision point and the type/source/amount of supports and the media used for delivering these supports. Intervention options are designed to impact mechanisms that underlie the health condition. In remote-care applications, different types of interventions can be triggered depending on the deviation detected. For example, the modification of the user interface (presentation/content/-navigation) or the state of an appliance (light, fridge, ...); The activation of some functionality; The generation of alarms (to highlight some potentially dangerous situations), reminders (to indicate an activity that should have been accomplished), prompt messages, explanation messages ; Providing persuasive suggestions (to encourage users to change their behavior), etc.

- tailoring variables: A tailoring variable is information concerning the individual that is used to decide when (i.e., under what conditions which is intervention_level, in our case.) and which intervention to provide.

- decision rules: The decision rules specify which intervention option to offer, for whom, and when. In other words, the decision rules link the intervention options and tailoring variables in a systematic way. There is a decision rule for each decision points. In our case, the decision rule is a cMAB algorithm that decides which intervention is the best option for a specific user in a specific time.

- Proximal outcome: The short-term goals the intervention is intended to achieve (i.e., minimize user anomalous behavior). In our case, these proximal outcomes will be save as reward (user feedback).

As we mentioned before, decision points occur when a true anomaly is detected. For intervention options, we consider an array of possible intervention messages and as tailoring variables (i.e., user context that is used to decide when to provide intervention and which intervention to provide), we use a fuzzy rule-based system to draw out the anomaly degree along with intervention level need for that anomaly. We considered three intervention levels: No intervention, Mild intervention (i.e., sending a notification) and High Intervention (i.e., sending alarms to the caregivers or family members). There is a decision rule for each decision points. Once an intervention option is chosen, a

proximal outcome (i.e., reward) is obtained. For example, our proximal outcome might be the number of steps the person walked in the one-hour following the decision point.

3.2 Just-in-time Adaptive Anomaly Detection and Personalized Health Feedback System Architecture

To describe the main components of our system and their inner connections, a four-layer architecture has been used (see Figure 3.1). The first layer includes the elements that concern modeling user context, user behavior, and sensing. In this layer, the system builds the user's daily routine model and maps each basic activity to the events in the user context (which is constructed based on the available sensors at the user place) for further analysis. The second layer comprises the recognizer (Activity Recognition). This layer uses the mapped events in the first layer along with specific models to verify human activities (e.g., sleeping, eating). Layer three takes as input the output of the recognizer. In this layer, the system detects anomalies in user behavior. Finally, the fourth layer is based on the feedback system, and it is responsible to make a decision about when to intervene, how to select and deliver the persuasive personalized interventions. In this layer, the detected anomalies along with the other components in the user context are fused to define the degree of the detected anomaly and identify the level of intervention needed for each specific user. Subsequently, the personalized intervention suggestion engine using contextual multi-armed bandit formalization generates personalized messages to help the user increase their life quality.

3.2.1 Modeling Human Behavior for Activity Recognition

Modeling complex activities requires the characterization of the temporal dependencies among their atomic activities. It is well-known that each individual often has a unique style of performing the same complex activity, which may differ noticeably from the others. To further complicate the matter, the same person might perform differently at a different time or location. One such example is provided in Figure 1.1, where "preparing breakfast" can also be performed in an alternative manner with only two atomic activities

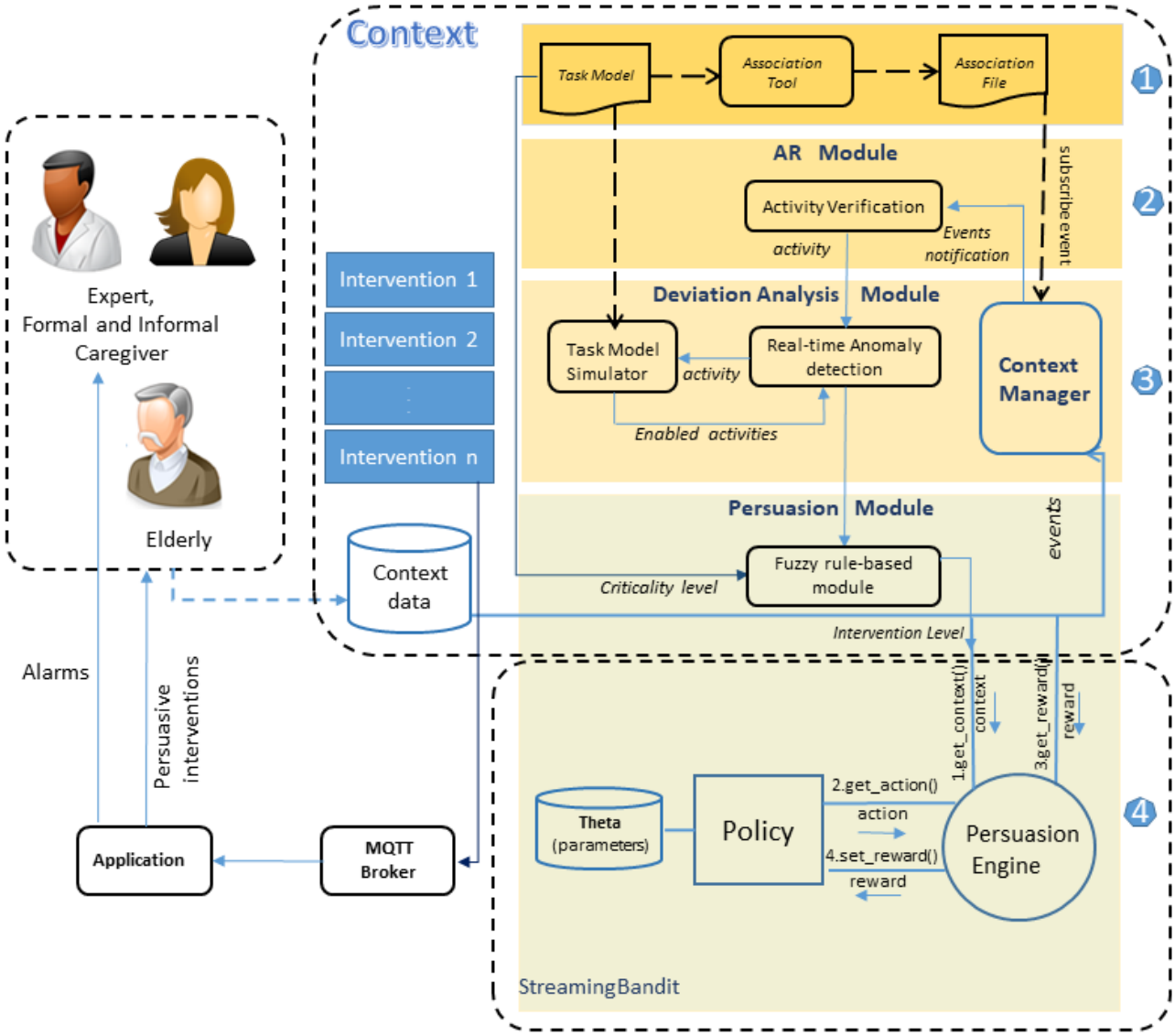


Figure 3.1: The Architecture of the System

involved (e.g., just by turning on and turning off the oven). This kind of inherent variability of complex activities usually manifests itself in terms of the types of the underlying atomic activities and their temporal relationships [214].

In this work, we define the task model, which represents the elderly routine daily activities with the previously existing ConcurTaskTrees (CTTE) tool [79]. For simplicity, seven activities of daily living (ADLs) have been considered. These activities include sleeping, waking up, eating, toileting/showering, walking, cooking, and resting. Each activity is associated with a criticality level (low, medium, high) personalized for each user. The criticality level shows how vital/important is performing this activity for the user. Activities in the CTT task model have a hierarchical structure and each activity can be divided into one or more sub-activities. Although each task model can include a

short term or a long term user activity (i.e., hourly, daily, weekly, etc.), users also can have multiple task models to cover their routine activity. Meanwhile, task models can be created by involving the relevant stakeholders (i.e., formal or informal caregivers and technical developers) and the elderly themselves.

Later, through the "Association Tool", the activities (without any further sub-activities) are mapped to the events in the user context. To this aim, we use a Context Model containing the context events. These events are configurable based on the available sensors in the elderly users' homes. In case of having an activity associated with composite events, these events are associated through the logical operators AND or OR. Therefore, an activity will be considered performed only if all the associated events occur. This association enables our system to recognize complex activities that should be detected using multiple sensors, such as waking up (e.g., pressure sensors on bed AND wearable sensors for detecting heart rate). So, effective processing and selection of meaningful mapping between the sensors (events) and activities in the task model are necessary to make them in a proper format for later use in the Deviation Analysis module. The output association file is an XML list of mappings between activities and events in the user context. Each event is associated with the unique $event\ \ id$ and the $source$ where the event comes from. Further, we subscribe these events to the Context Manager (CM) [215] in order to receive the notification each time an event occurs. Below, we explain in detail all the components of the first layer.

3.2.1.1 Task Model

To develop a task model from an informal textual description, stakeholders first have to start to analyze the description of the scenario, trying to identify the main tasks that occur in the description and refer each task to a particular role. It is possible to specify the category of the task in terms of performance allocation. In addition, a description of the task can be specified, along with the logical objects used and handled. Reviewing the scenario description, the stakeholders can identify the different tasks and add them to the task list. When each user's main tasks in the scenario have been identified, it might be necessary to make some slight modifications to the newly defined task list. The stakeholders can thus avoid task repetition, refine the task names to make them more

meaningful, and so on. Once the list of activities to consider is ready, we can start to create a hierarchical structure that describes the various levels of abstraction among tasks. The final hierarchical structure obtained will be the input for the main editor, allowing the specification of the temporal relationships and the tasks' attributes and objects.

The Task Models considered in this approach are stated according to the ConcurTaskTrees (CTT) language [79] and are defined in terms of a hierarchical composition of tasks connected by various temporal operators that describe the temporal relationships among tasks (see figure 3.2). The followings are the list of the available temporal operators used in CTT.

- enabling ($>>$): Specifies second task cannot begin until first task performed.
- disabling ($[>$): The first task (usually an iterative task) is completely interrupted by the second task.
- interruption ($| >$): First task can be interrupted by the second one. When the second task terminates then the first one can be reactivated from the state reached before.
- choice ($[\ ]$): Specifies two tasks enabled, then once one has started the other one is no longer enabled.
- concurrency ($|||$): Tasks can be performed in any order, or at same time, including the possibility of starting a task before the other one has been completed.
- order independency ($| = |$)):Tasks can be performed in any order, but when one starts then it has to finish before the other one can start.
- iteration ($\overset{*}{t}$): Define a task which can be repeated in time.
- optionality ($[t]$): Tasks that can be optional, which means the user or the system can perform them or omit them.

A task T is an activity that should be performed in order to reach a goal and can range from a high abstraction level (such as deciding a way to prepare food) to a concrete,

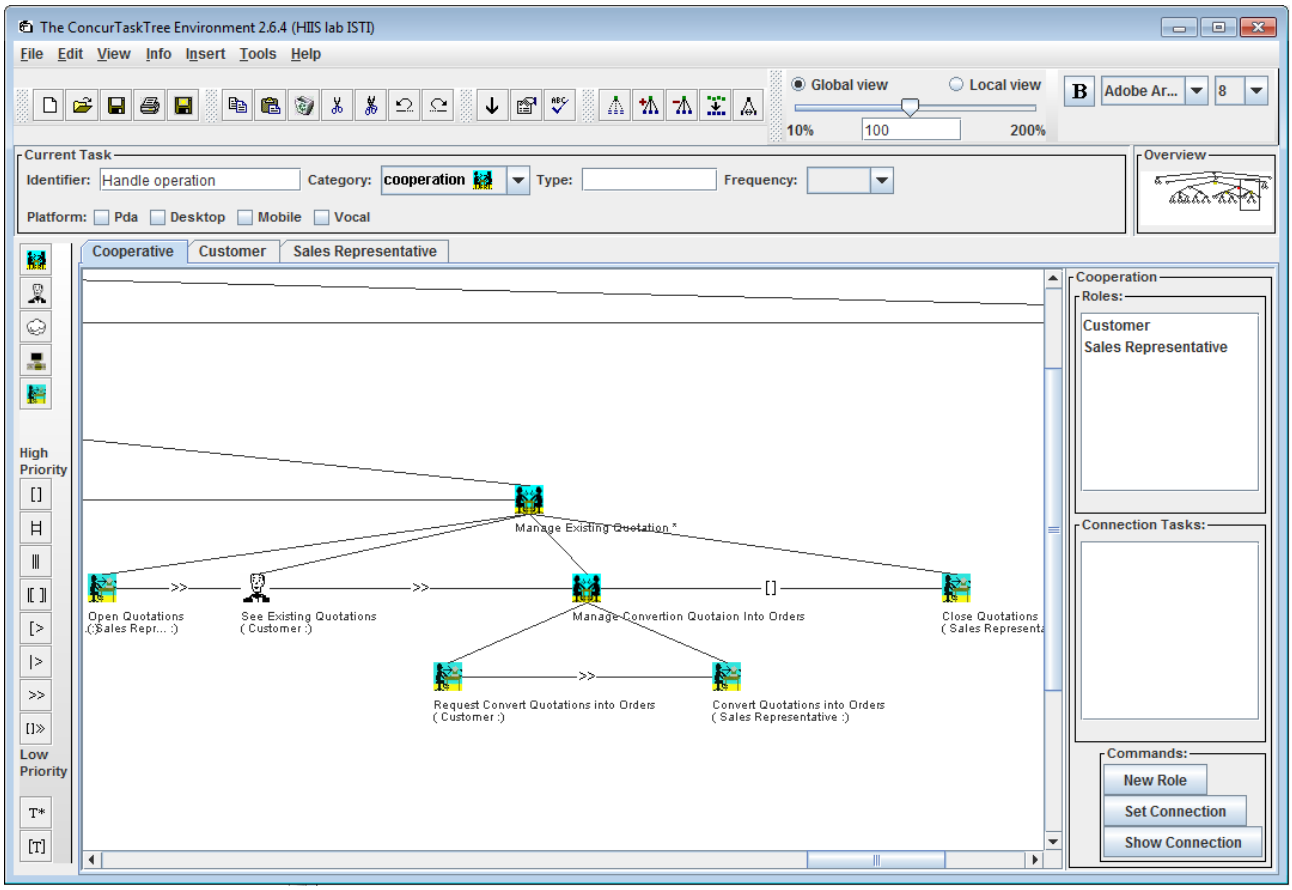


Figure 3.2: ConcurTaskTrees Environment

action-oriented level (such as "open the fridge door"). Basic tasks are *elementary tasks* that cannot be further decomposed because they do not contain any control elements. Each task in the task model is associated with some attributes. In particular, for our analysis, we exploit a time attribute that indicates the time (or the time interval) when the task should be performed. In addition, CTT provides the designer with the ability to define how the task should be allocated (i.e., to the user, to the system, to their interaction, or a combination of different types of allocation). In our case, since we consider tasks whose performance is detectable through sensors and user interactions, we limit our study to interaction and application tasks. In particular, the former refers to the tasks involving interactions with any physical or digital appliance, while the latter indicates the feedback from an application or action that happens without user interaction.

We consider a task model as composed of n tasks such as: $\Gamma = \{T_1, T_2, ..., T_n\}$ which defines the expected user behavior. In our method, a task can be represented formally by a 6-tuple, $T = <N, \iota, \theta, \xi, \tau_s, \tau_f>$, where: N is a task name, ι defines the maximum number of iteration of the task, θ is a boolean show the task optionality, ξ defines the criticality

level of the task consists of {low, medium, high} and [τ_s, τ_f] indicate the time interval in which the task should occur. Given a set of tasks in a task model, the elementary tasks may occur in many different orders according to the temporal operations between them.

Let Σ be the set of all possible sequence of elementary tasks permissible by the CTT task model. Given an $L \in \Sigma$, $L =< T_1, ..., T_m > where \ \forall \ i \in [1, m], T_i \ \widehat{=} \ T_j$ for some $j \in [1, n]$.

Definition 3.2.1. *Given a general sequence L taken from the task model as $L =< T_1, ..., T_m >$ and let $1 \leq x < y \leq m$, we can define S and F as finite, non-empty sets of start and final elementary task(s), as $S =$*{$T_1, ..., T_x$}* and $F =$*{$T_y, ..., T_m$}*. This means, users may have a choice of multiple tasks at the beginning and the end of the period of time considered in the task model. For example, for the task model in Figure 1.1, S=*{watching TV , walking}* and F=*{read newspaper, listen to radio}*.*

Between each pair of tasks in both start (S) and the final (F) sets, there is a choice [ ] temporal operation which specifies two enabled tasks such that, once one has started the other one is no longer enabled. Refer to the Figure 1.1, if Sara starts her day with the "watching TV", the task "Walking" is not enable anymore and vice versa.

3.2.1.2 Association Tool

The association tool contains two parts, namely the list of elementary tasks and the context model. This tool receives as an input the elderly task model and as an output provides an association file containing the mappings between the elementary tasks (which are the tasks in the task model with any control element, see the left panel in Figure 3.3) and the events in the context model (see the right panel in Figure 3.3). This is a solution for connecting the expected behavior with the actual one.

The context model [215] contains the context aspects that can be detected in the user context through the available sensors. It is worth pointing out that some elementary tasks correspond to a single event (e.g., the "take the pill" task corresponds to the event: "open pill" dispenser). However, there are cases in which the occurrence of one task is detected through the occurrence of multiple events (e.g., the "showering" task, depending on the available sensors, could correspond to multiple events: user presence = bathroom AND shower sensor = on). Therefore, an elementary task is considered performed only if, all the associated events occur. So, effective processing and selection of

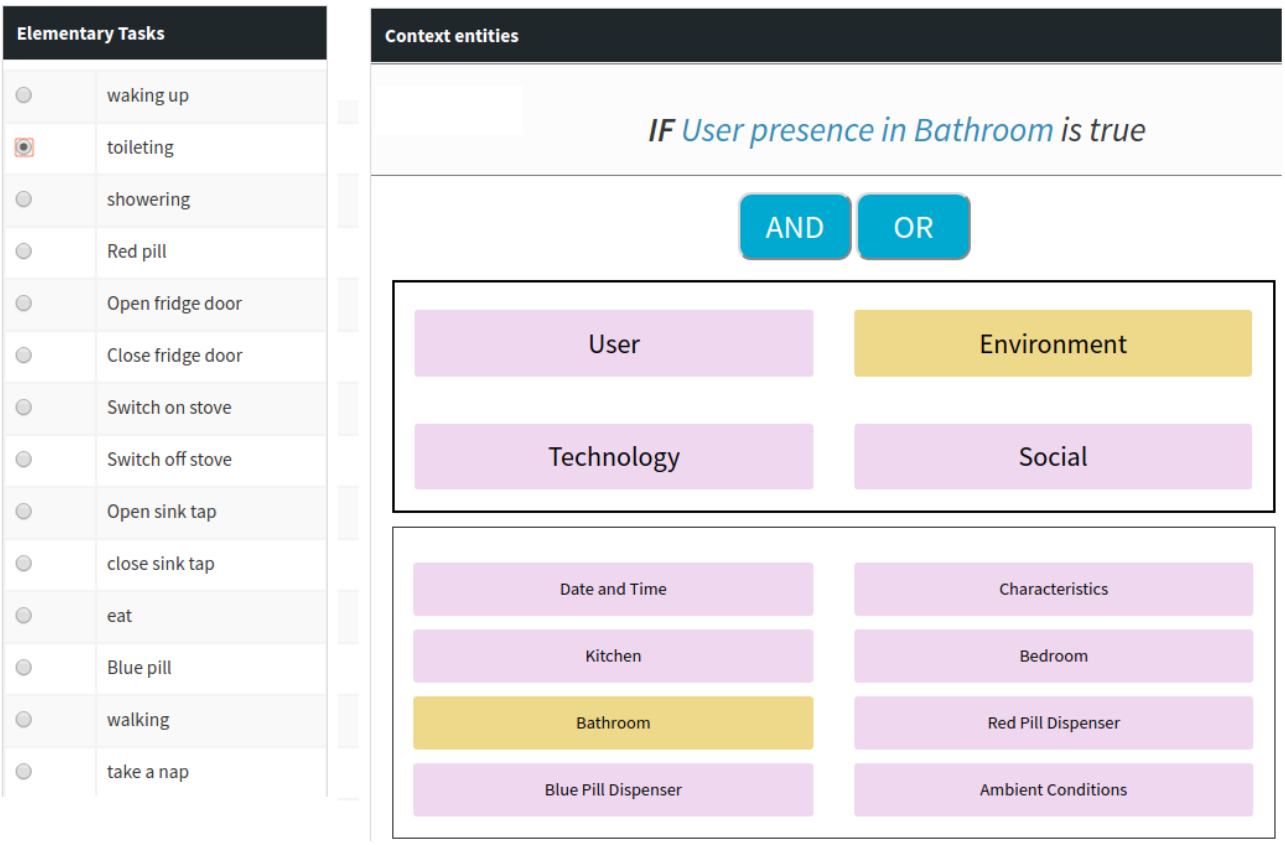


Figure 3.3: The association tool

meaningful events in the context model are necessary to have a proper format for later input to the Deviation Analysis Module. The identification of the relevant events indicating task accomplishment should be done by involving the relevant stakeholders and the elderly themselves. Stakeholders include formal or informal caregivers who know the relevant tasks and technical developers who know the possibilities and features of the available sensors. Thus, together they can identify the most appropriate associations.

At the user's login in the Association Tool, a requests is made to know the context in which the entire system is located: a request is sent to the RESTful API of the Context Server to obtain the context model on which a user can create its own rules (e.g., "If the user is in the bathroom = toileting"). These rules then will be subscribed to the Context Manager containing the list of events extrapolated from the rules and a REST endpoint so that it is informed when an event is verified.

3.2.1.2.1 Context-Dependent Behavior

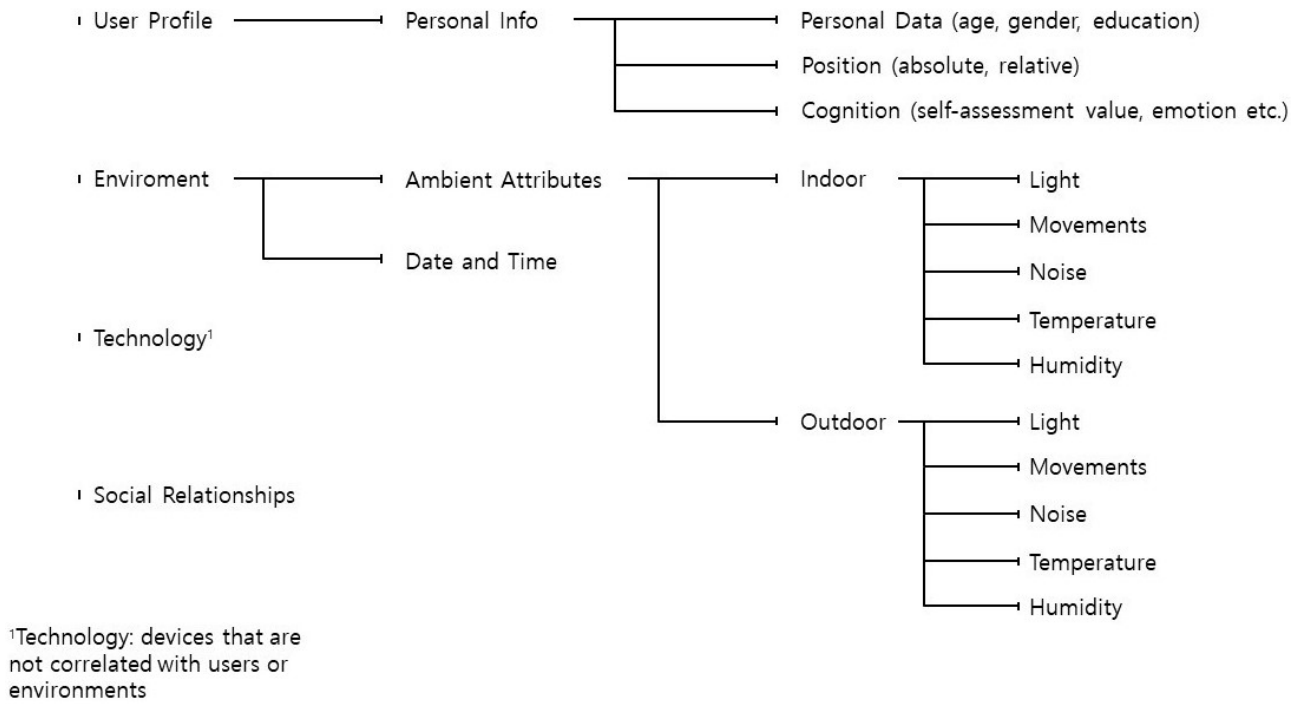


Figure 3.4: Initial context entities

The context-dependent behavior of the association tool is organized according to the context model's four main dimensions: user, environment, technology and social relationship. Each dimension is composed of various entities. The user is described in terms of personal data (e.g., age, gender, education), position (i.e., relative or absolute) and cognition (e.g., self-assessment value, emotional state, cognitive state, training results, training time,etc.). The environment is defined in terms of date, time and ambient attributes (e.g., light level, noise level, temperature, humidity, gas level, etc.). All dimensions will be refined in the future depending on the actual features of the target contexts of use (see Figure 3.4).

In order to properly support the specific, actual environment at hand, the context model needs to be 'populated' in terms of the actual triggers available in the setting considered. To this aim, at initialization time, the association tool asks the Context Server for the list of actual objects, sensors, and devices available in the current context. According to the information received from the Context server (there is a rest service in the Context server for this purpose), the association tool is able to populate the panel dedicated to the user context (i.e., the right panel in figure 3.3).

3.2.1.3 Event Subscription

Formal and informal caregivers can describe the association between events and user context entities through the association tool by mapping the events and the context entities which compose a personalized activity (e.g., IF the user opens the pill dispenser THEN she/he is taking the medicine). Those association rules are composed of Events and Conditions that take into account the state of the Context Entities. For this reason, the Association Tool communicates with the Context Manager (CM) by subscribing to the defined activities and receives the notification each time an event occurs or a condition is verified (i.e. in order to be informed when a rule is triggered). In Listing 3.1, you can see an example of a message that Association Tool subscribes to the Context Manager for a specific event, for example, user presence inside the kitchen. In the entityReference element specifies the Path of the context entity linked to the event (there is a REST service to get all the Path available in context). In line 32, we have to specify for each rule the endpoint where we want to be notified when the event occurs (which in our case is the Deviation Analysis module).

In this message, events are associated to a change in the state of a contextual entity. The “event” part of a rule can be both an elementary event or a complex event (e.g., Boolean, comparison or sequential operators). Instead, conditions refers to a persistent state of a contextual entity. The "condition" in a rule can be either elementary (e.g., Boolean predicates) or complex (composition of elementary conditions). If the subscription performs correctly the CM send a verification message with the unique RuleId (see Listing 3.2, line 4). It is important to save the ruleId returned by the context server, since it will assign an id that can be different from the one inserted by you during the subscription.

3.2.1.4 Context Manager

Context-awareness plays a crucial role in smart home applications, it improves the productivity of smart home as well as pervasive and mobile application [216]. The Context Manager manages the context-awareness of the platform by collecting and analyzing the contextual information coming from the environment/user/external sensors. It is a middleware Web-based application (Java Servlets, RESTful Web services) able to detect and communicate the data in a logical structure to the other modules (deviation analysis,

```
"ruleTypeList":[{
  "event":{
   "simpleEvent":{
         "entityReference":{
                 "xpath":"position/relativePosition/@typeOfProximity",
                 "dimensionId":1
                                        },
         "constant":{
                "value":"INSIDE",
                "type":"STRING"
                                  },
         "operator":"EQ",
         "eventId" :"1",
         "eventName":"SimpleEvent"
                         }
                   },
  "condition":{
        "entityReference": {
           "xpath": "user/position/relativePosition/@pointOfInterest",
           "dimensionId": 1
                   },
                "constant":{
                "value": "KITCHEN",
                "type": "STRING"
                   },
                   "operator": "GT"
                   },
  "priority":1,
  "id":1,
  "name":"simple event presence"
                }],
"adaptationEngineEndpoint":"http://xxx:8880/NewAdaptationEngine/rest/
   DeviationModule/"
```

Listing 3.1: Example of Subscription to the Context Manager.

```
1
2        "acceptedRules": [{
3                "ruleName": "simple event presence",
4                "ruleId": 2,
5                "ruleDbId": 103
6        }]
```

Listing 3.2: subscription verification from Context Manager.

persuasion module). The Context Manager (CM) [215] has a client-server architecture. It is a software module composed of a number of context delegates (running on one or more devices located in the environment where the user lives) and a context server. The Context Delegates are software connected to the appliances and devices that update the content (data, information) of the Context Server, which organizes the data into a common vocabulary used for communication with other platform components.

The next phase begins when the Context Delegates send the new values collected by the sensors to which they are connected to the Context Server (to dedicated REST endpoints). The Context Server proceeds to check all the subscribed triggers corresponding to the entity of the context that has been updated: in the event that a trigger (in the form of an event and/or condition) is satisfied, the Context Server will proceed to send a notification to the AR module to analyze the performed activity.

Therefore, synchronous notifications are automatically sent by the context server to modules that have previously subscribed (in our case, the Activity Recognition) for a particular state or for changing one or more parameters (i.e. when an event occurs or a condition is fulfilled). The advantage of the asynchronous approach is that the subscriber module is not required to continuously query the context server for the current values of the involved parameters. Thus, as soon as an event (specified in the association file, here "user presence inside the kitchen") occurs, the Context Manager sends a notification (see Listing 3.3) to the AR. The ruleId in Listing 3.2 will be used in the notification that will be received from the Context Manager, to recognize the performed activity (as you can see in Listing 3.3).

let ε be an event received from the CM. Each event ε is expressed as 5-tuple, $< \mathcal{R}, \mathcal{X}, \gamma, \upsilon, \tau >$ where in Listing 3.3, $\mathcal{R}$ (ruleId), is a rule number associated to the subscribed rule in Association file. Next element, $\mathcal{X}$ (xpath), indicates the event path in the context model,

```
{
        "ruleId": 2,
        "xpath": "environment/ambientConditions/@presence",
        "value": "true",
        "dimensionId": 1,
        "verified": true
}
```

Listing 3.3: Notification received from Context Manager.

γ (value) is a value of a state attribute associated with event, υ (verified) is a boolean indicating whether the value is true or false and finally, τ defines the event time.

3.2.2 Online Activity Recognition

Our Activity recognition (AR) verifies the activities of daily living of older adults at their homes. We verify activities, instead of inferring them, because our monitoring approach is driven by routines, initially sketched by users in their environment. Monitoring is supported by a lightweight sensor infrastructure, comprising non-intrusive, low-cost, wireless devices. Specifically, our approach relies on the following key observation: as people age their daily activities are increasingly organized according to a routine to optimize their daily functioning. As a result, their activities do not need to be recognized but should rather be verified. Deviations are a warning sign of degradation. Verification is performed by applying a simple formula to sensor data (events) received from the CM, for each activity of interest. The result value determines whether an activity has been performed. Thenceforth, the AR sends the verified activity to the deviation Analysis module.

Thus, we present the rule-based approach which extracts the activity from the simple events. This model sets the conditions that lead to the composite activity, selecting the primitive events and combining them according to a well-precise relationship. Most ADLs are composed of a succession of simpler events. A composite activity is a high-level activity composed of zero or more atomic events. The task of identifying so-called composite activities from basic events relies on a set of rules that analyze and correlate other events, considering the logical operators (And or Or) between these events. For instance, "sleeping" may consist of "opening the bedroom door" AND "going to bed"

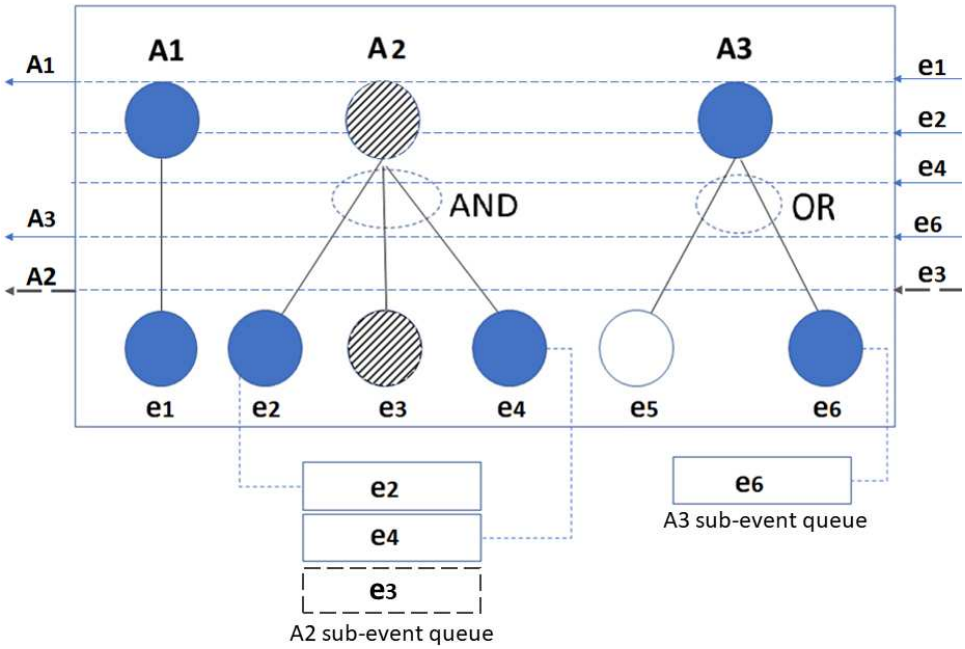


Figure 3.5: The sketch of the Activity Recognition taken from IBM [1].

AND "turning off the light". When all these events associated with the activity "sleeping" occur, regarding temporal relations, a sleeping activity considered to be complete.

A composite activity that uses the OR Boolean operator fires when any of its sub-events fires. A composite activity that uses the AND operator fires when all of its sub-events have fired. Each composite activity has a sub-event queue associated with it. The sub-event queue might be empty or contain only the names of those sub-events that have received and not been retrieved. So, each time the CM sends an event notification, the AR, from the ruleId, retrieves the related user association file and the related activity associated with the received event. Then, it checks the sub-event queue and controls the logical expression between the sub-events in the queue. If all the necessary sub-events (based on the logical expression AND and OR between them) have arrived, it means that the activity has been completed. Figure 3.5 shows an example of all the activities that are recognized by a particular set of events. Note that, two of them are composite activities (namely A_2 and A_3). The sub-events' queue for A_3 contains the event e_6, that triggers A_3. The sub-events' queue for composite activity A_2 contains the events e_2 and e_4 and it will be fired when e_3 arrives. Let us assume, for instance, that A_2 is defined as "sleeping: <enter the bedroom $(e_2)\wedge$ laying down on the bed $(e_3)\wedge$ turn off the light (e_4) >", and, A_3 as "taking the medicine: $<$ taking syrup $(e_5)\vee$ taking the pill (e_6) $>$". If the user behaves in such a way to produce the sequence $< e_2, e_4, e_6 >$ (which corresponds to a situation

in which the user goes to bed and only then remembers to take the medicine), the AR module will recognize "taking medication" activity (A_3) and will send it to the Anomaly Detection module. Later, if the user goes back and lays down on the bed (e_3), the AR module recognizes the "sleeping" activity and only then triggers A_2.

3.2.3 Deviation Analysis

The Deviation Analysis module receives two elements as input: i) the events occurred in the current context (received from AR) and ii) the enabled task sets (produced by the Task Model Simulator). The Deviation Analysis, through the real-time anomaly detection algorithm, controls if any deviation has occurred. The algorithm compares the temporal order and the time of the received activity with the expected activity produced by the task model simulator. When anomalies have been detected, the deviation analysis module sends the output to the Persuasion module.

3.2.3.1 Task Model Simulator

We implemented a Task Model Simulator (TMS), which receives as input the elderly task model, starts from the first elementary task enabled and communicates the list of tasks currently enabled (based on the temporal operators between the tasks). This process continues until it arrives at the final task in the task model. Indeed, the Task Model Simulator acts as a function $f : T \mapsto EN_T$, which is a map where T is the currently executed elementary task and EN_T is the array of enabled tasks after the executed task. For example, refer to the scenario in Figure 1.1, if the TMS gets as input the task "walking", then, based on the temporal operation between tasks, it communicates the list of enabled task: " toileting, showering, Red pill", because after the user goes for a walk, she can do the "morning hygiene" or "taking the Red pill" tasks in any order. Actually, the main goal of the TMS module is to detect if the task corresponding to the event(s) detected by the Activity Recognition module is one of those enabled according to the temporal relationships between tasks in the task model.

3.2.3.2 Anomaly Detection

The Anomaly Detection Algorithm, which is the core of the Deviation Analysis module checks whether the order (by interacting with the Task Model Simulator) and the time performance (as compared to the task model information) of the identified elementary task comply with the expected behavior. If not, the algorithm specifies the time and the type of anomaly encountered. A more detailed description of the algorithm mechanism for detecting anomalies will be found in chapter 4.

The anomaly detection algorithm can confirm the presence of anomalies, but depending on the user context the final situation may not be abnormal. Consider our example, Sarah is supposed to take her medicine (e.g., red medicine for Cardiovascular disease) at 8 a.m. and this medicine is highly critical for her. We assume that it is 9:01 and, she forgot to take her medicine. We assume that the "emergency treatment" for the elderly with Cardiovascular disease in a Hospital is legally defined as forgetting to take their medicine for 60 minutes or more. In our example, Sarah, who forgot to take her medicine for 61 minutes, gets the emergency treatment while another patient who was 56 minutes late gets virtually no treatment. The fuzzy theory could smooth out such inequities by offering a sliding scale which matches the degree of sickness (which in our work is the elderly health state: $\delta_{anomaly}$ and criticality level: means how vital is that activity) to the degree of treatment (level of considered intervention). Two similar patients with Cardiovascular disease would then experience similar "realistic" treatment regardless of social, economic, or any other status.

3.2.4 Persuasive Health Feedback

Our persuasion module consists of a fuzzy rule-based that implements a decision-making model system (see Figure 5.2) to recognize the true anomalies and an intelligent suggestion engine, StreamingBandit (see [217]), the design of which is inspired by the contextual multi-armed bandit (cMAB) problem [218].

We use a fuzzy rule-based system to decide about the degree of the abnormality detected which results in identifying the level of intervention needed for the specific user. In our scenario, if Sarah goes to bed at 10:30 and the "sleeping" activity is not very critical for

her, a fuzzy rule can conclude that the patient state is normal; this is not a truly abnormal situation, and she needs no intervention for now. These rules should be elaborated based on expert opinions. So, in this phase based on the intervention level detected by the Fuzzy system, the system decides whether to intervene or not (i.e., decision point).

The level of intervention can be categorized as "No intervention", "Mild Intervention" and "Strong Intervention". After the true anomalies and the level of intervention have been detected, the suggestion engine in the persuasion module by using the contextual multi-armed bandit formalization tries to find the best intervention message (intervention options in our case are messages) which is personalized based on the user context (tailoring variables). This module dynamically learns what is the best action for each individual based on the individuals' context and their response to each action (decision rule).

3.3 Communication Across the Platform Components and User Application

To better understanding how the different components of the architecture work together, we extend our "motivation scenario" in section 1.2 to the following one. Sarah is suffering from Mild cognitive impairment (MCI), her daughter Susan, was looking for a technical solution that could soften the impact of dementia symptoms on Sarah's daily life and improve her mother's quality of life. We offered our solution to Susan. First, our technical team allocated different sensors like, motion, temperature and humidity detection in every room in the user house, a sensor on a pill dispenser to control her daily medicines and a sensor under her bed to monitor her sleeping routine. To understand the Sarah's location, they gave her a smartwatch and they connected this smartwatch with the different beacons in different room to estimate her relative position and also to measure her heart beat and her daily steps. In addition, they installed an application on Sarah's smartphone and tablet that enable her to receive the interventions via different channel based on her preferences.

Susan has located her mother pills in the kitchen cabinet and she mentioned that it is very vital for Sarah to take her medicines on time. When we ask Sarah's doctor

to give a number between 1 and 10 to the vitality (the criticality degree of the task) of her medicines and the bed time, he relatively gave us 10 and 6. Based on these information and with the collaboration of Sarah and her daughter, we created a task model that describe Sarah's daily routine. We did this task via CTT software. Each task is associated with time interval, criticality level, optionality and etc. After that, we mapped each task in her daily routine (e.g., taking medicine) to the sensors available in the house and subscribe all the association rules to the Context Manager. For example, for the medication the association rule is as follow.

Example 1 (association rule for medicine)**.** *If the user is INSIDE the KITCHEN and the pill dispenser is OPEN = Taking medicine.*

Now, it passes 1 week. In the morning she should take her medicine somewhere between 8 and 9 a.m after having breakfast. Instead, this morning she wakes up at 7:00, goes to the kitchen, open the cabinet and open the pill dispenser to take her medicine. So, as soon as she walks into the kitchen the context manager sends a JSON notification (see Listing 3.1) to the AR module. The context manager is a middleware Web-based application (Java Servlets, RESTful Web services), that notifies the activity verification module (AR) via HTTP(S). It is worth notice that, in the real-time analysis, communication among the various components is performed through REST services. The activity verification module receives the notifications in JSON format (see Listing 3.3), retrieves the user activity from the association file and in this case, it waits until the activity gets complete. The AR finds 2 files (cooking and taking medicine) that both includes "user position = kitchen". As soon as Sarah takes the second step and opens the pill dispenser, CM sends another notification to the AR. At this point, AR identifies the "taking medicine" task as "completed" and sends the activity to the Deviation analysis module.

The Deviation Analysis module back-end is implemented in PHP (i.e., the anomaly detection algorithm implementation) and its front-end part in HTML/JavaScript. This module verifies if any deviation happens. In case the anomaly detection algorithm finds any anomaly, it sends the detected anomaly to the persuasion module.

When the anomaly detection algorithm receives "taking medicine" as an input, it sends the "taking medicine" task to the task model simulator. The TMS, refers to the task model and checks the activity's order and time. Based on Sarah's daily routine, she should first

have her breakfast and then take the medicine and it should happen between 8 and 9 a.m. The algorithm checks also the optionality and the criticality level of the task. So, first the algorithm detect the anomaly and mark it as *Order* and then as *Difference-Early-time*. Finally, it combines both anomalies and outputs the final anomaly as *Difference-Order-Time* (equation 4.11, that means user perform an activity in a wrong temporal order and in a wrong time).

The algorithm sends the result to the persuasion module that is responsible for the decision-making process. But, not all anomalies detected are true anomalies. The system filters the anomalies through the fuzzy rule-based system and identifies the level of intervention needed for that anomalous task. The inference system in fuzzy logic, maps the input (task = taking medicine, criticality = 10, anomaly type = Difference-Order-Time) space to the output space.

According to the rules in our system (which has been elaborated by experts) the level of intervention needed for Sarah in this specific situation is "Medium" and it means that we need to send an intervention message to Sarah. Now, the persuasion module sends (via restful services) all information along with the user context to the intervention generation engine (i.e., persuasion engine).

The persuasion engine, by using one of the well-known sequential decision-making formalization, contextual multi-armed bandit, chooses the best intervention (between all the persuasive interventions created using Cialdini's persuasive principle) for Sarah's particular situation which is calculated base on her preferences and her historical context (i.e., based on her response to the former messages). This personalized intervention message will be send to Sarah via the application former installed on her devices. This application has been subscribed to the persuasion module to be informed about relevant interventions (in this work, the application for receiving the notifications/alarms/reminders has been developed in PETAL European project). The communication between the persuasion module and the Application is through the MQ Telemetry Transport (MQTT) protocol. The persuasion engine sends the intervention to the broker by publishing them using a specific topic and the application will subscribe to the same topic in order to receive them.

In listing 3.4, the $\$notificationMsg$ is the intervention message, the $\$notificationType$

```
msg ='{
      "action":[
        {
         "invokeFunction":{
          "input":[
           {"value":{"constant":{"value":"'.$notificationMsg.'","type":"
              STRING"}},
           "name":"'.$paramName.'"},
           {"value":{"constant":{"value":"NOTIFICATION","type":"STRING"}},
           "name":"notificationMode"}],

         "name":"'.$notificationType.'"

        }
         }]
 }';
 $client = new MQTTClient("persuasion.notification.service","giove.isti.
    cnr.it",1883);
 $client->connect();
 $client->publish($topic,0,"y",$msg);
```

Listing 3.4: Publishing the intervention messages via MQTT

could be a NOTIFICATION or an ALARM and $\$topic$ is the user's application which receives the intervention. Thus, back to our scenario, as soon as Sarah opens the pill dispenser, she receives a message as, *Keep a "medicine calendar" that your doctor gave you with your pill dispenser, and note each time you take a dose.*

We chose MQTT because it is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery.

These principles also turn out to make the protocol ideal for the emerging machine-to-machine (M2M) or Internet of Things (IoT) world of connected devices, and for mobile applications where bandwidth and battery power are limited. Since it is the application that subscribes for a topic to the MQTT broker, there is no need to open and route requests into the router. Thus we implemented the possibility to send the actions using MQTT protocol by installing the MQTT broker called Mosquitto.

3.4 Chapter Summary

The present chapter provides a 4 layers "just-in-time adaptive intervention" architecture to close the loop between user event log data and personalized health intervention recommendation. This architecture supports the unique and changing needs of individuals to achieve the best outcome for each. Our architecture has three main features: behavioral support that directly corresponds to a need in real-time; content or timing of support is adapted or tailored according to input collected and detected by the system; support is system-triggered. To this aim, the first and the second layers include the elements that concern modeling user behavior, sensing and verifying their daily activity. On the other hand, the third and fourth layers are responsible for detecting any significant changes in their daily routine and issue personalized health feedback.

This architecture is capable not only to support the implementations of the remote care applications in smart homes but has the potential to support various fields, such as modeling erroneous behavior to improve the safety of human-interactive systems based on where and how human behavior diverges from a task model [20], designing and assessing interactive systems [21],(in first 2 layers) and also the implementations of personalized recommendation systems that analyze the user behavior [187, 206–209].

Chapter 4

Real-time Anomaly Detection

For healthcare professionals, it is significant to determine the accurate health status of a remotely located patient or an aged person, so that when there is a need, appropriate treatment is vetted in a timely manner. Though, there are many studies addressing anomaly detection in user behavior, there have been few contributions address the issue of detecting possible deviations from expected users' routines at the abstraction level considered in our work. Furthermore, deviations could become manifest in the different ways based on the activity type, place, time and the duration compared to the expected one.

In addition, time plays a serious role, because some anomalies need longer time-span to consider as a serious, risky situation and some are detectable immediately. For example, going to the bathroom is not a risky situation by itself but if the older people repeat this activity too frequently in a day, it can be an initial symptom of health problem. In this study, by using the profiling strategy, we focused on detecting behavioral "changes" of occupancy's usual lifestyle pattern which could involve the changes in different contextual aspects, such as spatial (location), temporal (time and duration), time's order, activity's order, health's status and more.

4.1 Anomaly Classification

It is an unequivocal fact that whenever a human is involved in an activity, human error will occur at some point as the Roman philosopher Cicero cautioned “It is the nature of man to error”. Errors are the result of actions that fail to generate the intended outcomes. They are categorized according to the cognitive processes involved towards the goal of the action and according to whether they are related to planning or execution of the activity.

The two leading general taxonomies for classifying, and modeling human behavior are Generic Error Modeling System (GEMS) [219] and phenotypes of erroneous action [220]. Hollnagel describes how erroneous behaviors observably manifest as deviations from a plan of action. He proposed a clear distinction between the phenotypes (manifestations) and the genotypes (causes) of erroneous actions. In his opinion an actions by human operators can fail to achieve their goal in two different ways: The actions can go as planned, but the plan can be inadequate, or the plan can be satisfactory, but the performance can still be deficient.

GEMS categorized the possible human behavior mistakes in 3 types (see Figure 4.1): rule-based, knowledge-based and slip based mistakes. If the user is aware of the problem, we can bring in the play the rule-based and knowledge-based performance but, if the user has the knowledge about how to perform the activities, slips are supposed to occur, and such analysis can be supported by a "task model". In a familiar and anticipated situation, people perform a skill-based behavior. At this level, they can commit skill-based errors (slips or lapses). In the case of slips and lapses, the person’s intentions were correct, but the execution of the action was flawed - done incorrectly, or not done at all. This distinction, between being done incorrectly or not at all, is another important discriminator.

When the appropriate action is carried out incorrectly, the error is classified as a slip. When the action is simply omitted or not carried out, the error is termed a lapse. “Slips and lapses are errors which result from some failure in the execution and/or storage stage of an action sequence.” Reason refers to these errors as failures in the modality of action control: at this level, errors happen because we do not perform the appropriate

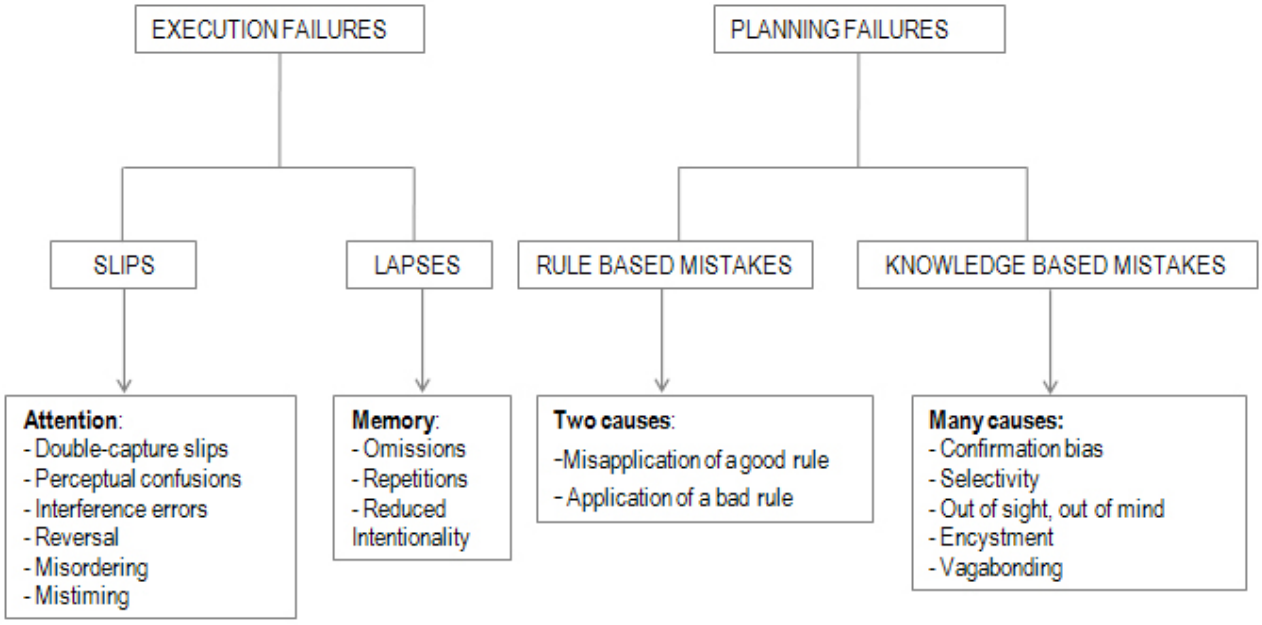


Figure 4.1: Execution and planning failures adapted from Rasmussen

attentional control over the action and therefore a wrong routine is activated.

4.1.1 Possible Task-related Anomalies in Elderly Behavior

As already mentioned, an anomalous situation is characterized by a dissimilarity between the observed situation (derived from sensors in the current user context) and the expected one ($\sigma \in \Sigma$) described in a task model. The identified deviation could be characterized by different degrees of severity and could be the sign of various situations such as the beginning of an unhealthy habit, the insurgence of an illness or even a serious physical/mental decline. Thus, analyzing the behavior of the elderly can be useful to highlight trends as well as giving recommendations and health-related reminders to them.

In Reason's taxonomy [219], slips (i.e., the user has the knowledge about how to perform the activities) occur due to attention failures and can manifest as *omissions, repetitions, and commissions*. However, based on the analysis of the example scenarios (the possible anomalies in the elderly routine such as missing or repeating some critical tasks), and considering that the time and the order of task performance in elderly routines play an important role, our method models these slips in a more systematic categorization of the

various types of deviations.

In Reason's taxonomy, slips can manifest as omissions, repetitions, and commissions. However, based on the analysis of the example scenarios and considering that the time and the order of task performance in elderly routines play an important role, our method models slips in a more systematic categorization of the various types of deviations. In addition, our detailed classification of different anomaly types also helps to converge more rapidly to the right anomaly identification and also demonstrate the reason behind each deviation.

We considered two types of classifications for the different grades of anomalies: i) anomaly in complete sequences and ii) in partial sequences (prefixes) of elementary tasks ($P = <T_1, ..., T_p>$, and, $P \in \sigma$). In the former, anomalies are defined based on the whole sequence (for example at the end of the day) and in the latter, the different degree of anomalies are considered for the partial sequences (from the first activity received till the current performed activity. Figure 4.2 shows an example of prefix sequences.). Using the general anomaly classification for the partial sequences can cause many errors before converging on the right classification, the more detailed classification also helps to converge more rapidly on the right classification and in addition demonstrate the reason behind each deviation. In formalizing such classification, we used the binary symbol ($\circ$) to combine two strings. For example, if $P =< A, B >$ and $p' =< C, D >$, $P \circ P' =< A, B, C, D >$.

The general classification of the anomalies considering the whole sequence (sequence P) is as follows:

Less: A task that was expected has not been performed. Thus, P is Less and t is a missing task if:

$$P \notin \Sigma \quad \wedge \quad P =< P_1 \circ P_2 > \wedge \ \exists \sigma \in \Sigma : \sigma =< P_1 \circ T \circ P_2 > \tag{4.1}$$

More: A task has been performed more than expected time. Hence, P is More and T is an extra task if:

$$P \notin \Sigma \quad \wedge \quad P =< P_1 \circ T \circ P_2 > \wedge \ \exists \sigma \in \Sigma : \sigma =< P_1 \circ P_2 > \tag{4.2}$$

Difference: The tasks considered have been performed differently from what the designer intended in the task model. So, P is Difference and t is a task out of its order if:

$$\begin{aligned} &P \notin \Sigma \quad \wedge \quad P =< P_1 \circ T_1 \circ T_2 \circ P_2 > \wedge T_2 \notin EN_{T_1} \\ &\exists \sigma \in \Sigma : \sigma =< P_1 \circ T_1 \circ T_3 \circ P_2 > \wedge T_3 \neq T_2 \end{aligned} \tag{4.3}$$

No-Anomaly: A task has none of the above conditions.

$$P \in \Sigma \tag{4.4}$$

As we defined in section 3.2.1.1, $\sigma =< T_1, ..., T_m >$ is a sequence of tasks planned by the user and given $h \leq m$, σ'= $< T_1, ..., T_h >$ is a subsequence from σ. For finding the anomalies in the partial sequences, each time the Deviation analysis send an elementary task to the algorithm, it compares the partial sequence P with the σ'. It is worth to mention that, the tasks time are embedded in them and the ascending relationship between tasks time has been preserved means: $T_i \circ T_j$ = $T_i^{t_1} \circ T_j^{t_2} \Rightarrow t_2 > t_1$.

Given $P \notin \Sigma$, the anomalies in the partial sequences are categorized as:

- ***Less_Partial:*** An event that was expected to be carried out, has not been performed in the partial sequence.

$$\begin{aligned} &p =< P_1 \circ T_i > \wedge \exists \sigma \in \Sigma : \sigma =< P_1 \circ T_{i-1} \circ T_i \circ P_2 > \wedge \\ &\nexists \sigma' \in \Sigma : \sigma' =< P_1 \circ T_i \circ P_2 > \text{for any } \ P_2 \wedge \quad current - time > \tau_f(T_{i-1}) \end{aligned} \tag{4.5}$$

- ***Difference-Early-time:*** An event that was expected to be carried out in certain interval of time, has been performed before its starting time.

$$p =< P_1 \circ T^t \circ P_2 > \wedge \exists \sigma' \in \Sigma : \sigma' =< P_1 \circ T \circ P_2 > \wedge \ \ t < \tau_s \tag{4.6}$$

- ***Difference-Later-Time:*** An event that was expected to be carried out in certain interval of time, has been performed after its completing time is finished.

$$p =< P_1 \circ T^t \circ P_2 > \wedge \exists \sigma' \in \Sigma : \sigma' =< P_1 \circ T \circ P_2 > \wedge t > \tau_f \tag{4.7}$$

- ***More-Order:*** An event is performed more times than expected one and out of its order.

$$\begin{aligned} &P =< P_1 \circ T_i \circ P_2 \circ T_j >: T_i = T_j \ \ ,T_j \notin EN_{T_{j-1}} \ \ \wedge \\ &\exists \sigma' \in \Sigma : \sigma' =< P_1 \circ T_i \circ P_2 \circ T_k > \wedge T_k \neq T_j \end{aligned} \tag{4.8}$$

- ***More-Number:*** An event is performed more times than expected one, in its time interval and order.

$$P =< P_1 \circ T_i \circ T_j \circ P_2 >: T_i = T_j \ \ ,\exists \sigma' \in \Sigma : \sigma' =< P_1 \circ T_i \circ P_2 > \tag{4.9}$$

- ***Difference-Order:*** An event has been performed in a wrong temporal order.

$$P =< P_1 \circ T_1 \circ T_2 \circ P_2 > \wedge \ \ T_2 \notin EN_{T_1}, \exists \sigma' \in \Sigma : \sigma' =< P_1 \circ T_1 \circ T_3 \circ P_2 > \wedge T_3 \neq \varepsilon_2 \tag{4.10}$$

- ***Difference-Order-Time:*** An event considered has been performed in a wrong temporal order and at a wrong time.

$$P =< P_1 \circ T^t \circ P_2 \circ P_3 > \wedge \ \ \exists \sigma' \in \Sigma : \sigma' =< P_1 \circ P_2 \circ T \circ P_3 > \wedge \ \ \wedge t \notin [\tau_s, \tau_f] \tag{4.11}$$

As we referred before, as the partial sequence evolves (with the occurrence of each new observation), the output of the anomaly detection (and thus the potential anomaly already identified) may change or it may remain the same. For example let $\sigma =< A^{(t_1-t_2)}, B^{(t_3-t_4)}, C^{(t_5-t_6)}, D^{(t_7-t_8)} >$ be a sequence of activities which describe the user expected routine, where $[t_1, t_8]$ is the time interval of the activities performance. Thus, given $P =< A^{(t_0)}, B^{(t_3)} >$ as a partial sequence, we have *Difference-Early-time* anomaly on task A. Assume that at t_4, user performs activity A again. So, the partial sequence becomes $P =< A^{(t_0)}, B^{(t_3)}, A^{(t_4)} >$ and subsequently the anomaly type of A changes to *More-Order* based on Equation (4.7). As you see in Figure 4.2, the activities' anomaly type (e.g., A) may change over time as the partial sequence get completed.

Disjoint anomalies cannot happen at the same time. In other words, they are mutually exclusive. For example, we suppose that Sara should take her Red pill at 8:00 AM before breakfast. If she takes her pill at 9:00 AM but still before the breakfast, we encounter

Time	Expected Activity Sequence	Received Prefix Sequence	Anomaly Type
t_0	--	A	Difference-Early-Time (**A**)
t_2	A	A	Difference-Early-Time (**A**)
t_3	AB	AB	Difference-Early-Time (**A**)
t_4	AB	ABA	More-Order (**A**)
T_5	ABC	ABAC	More-Order (**A**)
t_8	ABCD	ABAC	More-Order (**A**) Less (**D**)

Figure 4.2: The example of task-related anomalies

the "Different-Later-Time" anomaly, which at the same time it cannot be a "Different-Early-Time" (a task cannot be performed before and after its expected time at the same moment), "Less" (because it requires that task has not been performed at all), "More" (requires task has been performed in its time interval but repeatedly) and "Different-Order" and "Different-Order-Time" which require the violation of the task order.

lemma 4.1.1. *Anomalies in Equation 4.4- 4.10 are pairwise disjoint. Given P as a partial sequence, P cannot have two or more classification of the anomaly on the same Task at the same time.*

Proof. We prove our statement by contradiction. Let $\sigma =< A^{(t_1-t_2)}, B^{(t_3-t_4)}, C^{(t_5-t_6)} >$ be a sequence of elementary tasks. We assume, at t_2 we have a partial sequence P as $< A^{(t_1)}, B^{(t_2)} >$ and task B has two kind of anomalies *Difference-Order* (Equation 4.9) and *Difference-Early-time* (Equation 4.5) at time t_2. So, based on Equation (4.5) : $B \in EN_A$, $t_2 < t_3$ and based on Equation (4.9) : $B \notin EN_A$, $t_2 \in [t_3, t_4]$, which obviously, $t_2 < t_3$ but $t_2 \notin [t_3, t_4]$ and also, task B cannot be enabled by task A and at the same time not be enabled by task A. Thus, task B at t_2 has just *Difference-Early-time* anomaly. The other cases given by all the possible combination of these 7 cases (Equation 4.4- 4.10) have a similar proof. Since these proofs are straightforward they are omitted here.

□

4.2 Online and Personalized Anomaly Detection

Trough the events received via the Context Manager, the online AR module recognizes the activity associated to the elementary task in the task model and sends it to the Deviation Analysis module. The Deviation Analysis, in turn, through the real-time anomaly detection algorithm controls if any deviation has occurred. The algorithm by using the information in the user task model and the task model simulator (TMS), extracts the personalized context of each user activity in one hand and the temporal relationship between the received activity (associated with task in the task model) and the enabled task before and after the received activity on the other hand, to control if any deviation is happening in terms of time, order and the number of times an activity is repeated.

4.2.1 Real-time Anomaly Detection Algorithm

As each activity (associated with elementary task in task model and received via AR module) occurs over time, the Anomaly Detection Algorithm (see Algorithm 1), starts to construct a sequence (σ) permissible by CTT user task model (which is the expected user behavior), using the currently received activity and the information generated by the Task Model Simulator (TMS). Sequence σ is one of the possible sequences of tasks with respect to the temporal operators between tasks in the user task model and serves as ground truth to be compared with the current received partial sequence (P) to find the anomalies.

As soon as the algorithm receives as input an activity, it starts to check the time interval of the received activity (defined in the task model). Line 2 and 3 check if the Equation (4.5) holds - means that the task is happening before its planned time interval- marks it as *Difference-Early-Time* and if the Equation (4.6) holds - means that the task is happening later than its planned time interval- marks the task as *Difference-Later-Time*. The next step, it checks if the received activity has happened before. So, line 4 controls if in the partial sequence P, there is any task equal to the currently received elementary task, if yes, controls the maximum number of iterations allowed for that task, if the threshold has been violated marks the task as *More* and deletes it from the sequence P.

As we have already mentioned, our algorithm makes the user planned sequence σ step

ALGORITHM 1: Anomaly detection

Data: Detected activity (associated to elemetary task) : T_i
Result: Anomalous task T

1 **while** $i < n$ **do**
2 **if** $T_i.t > \tau_f$ **then** marks T_i as *Difference-Later-Time*
3 **if** $T_i.t < \tau_s$ **then** marks T_i as *Difference-Early-Time*
4 $Dup \leftarrow$ *Delete_Duplication*(P) and marks Dup as $More$
5 constract σ by using (P, S, F) and marks *Dif* and *Les* as *Difference* and *temporary-Less*
6 **while** $flag == true$ **do**
7 **for** $\forall i \in [0, n-1]$ **do**
8 $swapcount \leftarrow 0$ `// number of time we swap the tasks`
9 $F\_error \leftarrow 0$ `// number of times we encounter nonswappble tasks`
10 **if** $T_i \notin EN_{T_{i-1}} \wedge T_{i-1} \in EN_{T_i}$ **then** Swap (T_i, T_{i-1}); $swapcount++$
11 **else** $F - error++$
12 **end**
13 **end**
14 **if** $swapcount > 0$ **then** $Flag = true$ **else** $Flag = false$
15 **if** $F\_error > 0$ **then** `// there exist 2 non-swappable tasks`
16 $S_p \leftarrow$ *FindShortestPath* $(T_{i-1}, T_i) \rightarrow \sigma =< P, S_p, T_i >$
17 *Delete_Duplication* (σ)
18 $Temporary - Less \leftarrow |\sigma - P|$
19 *difference1* $\leftarrow |LCS(P, \sigma) - P|$
20 *difference2* $\leftarrow |LCS(\sigma, P) - P|$
21 *Difference* $\leftarrow$ *difference1* $\cup$ *difference2*
22 **end**
23 **if** $\exists T \in$ Less $\wedge\, current - time > T.\tau_f$ **then** $T.anomaly \rightarrow$ *Less-Partial*
24 **if** $\exists T \in$ Difference-Later-Time $\cup$ Difference-Early-Time $\wedge\, \varepsilon \in$ Difference **then** $T.anomaly \rightarrow$ *Difference-Order-Time*
25 **if** $\exists T_i \in$ Difference, $T_i \notin EN_{T_{i-1}}$ **then** $T_i.anomaly \rightarrow$ *Difference-Order*
26 **if** $\exists T \in$ More $\wedge\, T_i \notin EN_{T_{i-1}}$ **then** $T.anomaly \rightarrow$ *More-Order* **else** $T.anomaly \rightarrow$ *More-Number*
27 **end**

by step. As the first step, at line 5 the algorithm finds the start and the final tasks in the task model and inserts/shifts them at the start and the end position in the sequence σ as $\sigma =< S, P, F >$. Therefore, the algorithm controls whether the sequence P includes any member of the start (S) and the final (F) tasks in the task model. If any member in S and F exists in P but not at the start and the end of the P (see d^s and d^f in Listing 4.1), it shifts them to the start and the final position in the σ and marks these tasks (see *Dif* in Listing 4.1) as *Difference*. While, if any member of S or F does not exist in P (see l^s and l^f in Listing 4.1), it inserts the start and the final task(s) at the start and the end of the sequence σ and marks these tasks (see Les equation in Listing 4.1) as *Temporary-Less*.

Further, the algorithm (line 8-11) checks if the currently received activity corresponds

$$
\begin{aligned}
d^s &= \{T_i \in P : T_i \in S \wedge \exists \quad T_j \notin S, T_j \in P \quad j < i\} \\
d^f &= \{T_i \in P : T_i \in F \wedge \exists \quad T_j \notin F, T_j \in P \quad j > i\} \\
Dif &= d^s \cup d^f \\
\\
l^s &= \{T_i \in S : T_i \notin P\} \\
l^f &= \{T_i \in F : T_i \notin P\} \\
Les &= l^s \cup l^f
\end{aligned}
$$

Listing 4.1: Set the start and the final task

to one of the enabled task/s of the last task in the sequence P. If so, the received activity was the expected one (according to the task model), then the process iterates in a similar manner for the next activity receives from the AR. Otherwise (i.e., Equation 4.3), it controls the opposite order to see if the previously performed activity corresponds to an enabled task of the currently received activity. If yes, it swaps two consecutive activities (which are not in the right order) in order to make the user a planned sequence (σ). While, if none of the above conditions were applicable, it means there are missing task(s)/activities between the currently received activity and the last activity in the partial sequence P.

For finding these missing tasks (line 16), the function *FindShortestPath()* takes in input these two successive activities (that are not swappable) and finds the shortest paths between them. Then, it fills up the distance between these two activities by the missing tasks and deletes the repetitive tasks (line 17) in the new sequence if there is any. Later, by comparing the sequence P and the planned sequence σ, it finds any missing tasks and marks them as *Less* (line 18).

Now, we have a complete planned sequence σ. Next, the Longest Common subsequence *(LCS)* function finds the longest subsequence common between two sequences (σ, P). It compares the result of the LCS with P and saves the output (if it is any) as *difference1*. Later, it repeats this process by switching the parameters as $LCS(P, \sigma)$ and compares the result with the partial sequence P and saves the output as *difference2* (line 19-20). Later,at line 21, it merges the result of *difference1* and *difference2* and mark them as *Difference*.

The last step is announcing the final decision about the anomaly type of the received activity (lines 23- 27). In this step, the algorithm checks first the tasks which are marked

as *Temporary-Less*, if the current time is greater than the end time in their predefined interval ($t > \tau_f$), the algorithm saves these tasks as a *Less-Partial* anomaly. Otherwise, they remain marked as *Temporary-Less* for future analysis. If the tasks which have been marked as *Difference-Early-Time* or *Difference-Later-Time* are even marked as *Difference*, their anomaly becomes *Difference-Order-Time*, otherwise, their anomalies remain the same.

For the tasks that are marked as *Difference*, the algorithm checks again the Equation (4.9) and if it is true, it announces these tasks as *Difference-Order*. And finally, if tasks which are marked as *More* are not enabled by their previous task (Equation 4.9), announces them as *More-Order*, otherwise, they are *More-Number*.

4.2.2 Simulation Studies

In this section, we empirically demonstrate the performance of the algorithm. For monitoring the result of the algorithm, we developed a tool using JavaScript, JQuery and, PHP. The algorithm is implemented in PHP 7.0 language and evaluated on a PC with Intel Core i5 CPU.

It is worth noting that, two different rounds of simulations have been carried out in this study. In the first round, we considered a short-term user activity (a day) and we created a task model containing the one-day daily routine program of the elderly. In the second round, we created a complex task model containing a whole week routine of the elderly. It means that the user has a different program during the week and over the weekend. We considered these two different simulations to show that the algorithm is not limited to the short and fix time frame, as it can detect anomalies concerning the task model. If the task model is extended to cover a longer period, say a week, the algorithm will work accordingly. We put the result of these simulations separately in section 4.2.2.1 and section 4.2.2.2.

In order to have a systematic analysis of the algorithm capability to detect the anomalies, we carried out a laboratory evaluation by simulating the partial task sequences composed of activities received from the AR module. We adopted the following simulation methodology:

1. **Preparation of the ground truth**. To this aim, we created two task models, one considering one day and the other one spans a week (i.e., the user follows the same routine during the weekday and a different routine on the weekend. For example, the user goes to the church on Sunday). In the second task model, the activities of a weekday have different order and time intervals $[T_s, T_e]$ from those on the weekend. In these example task models, the routine includes activities such as taking medicine, showering, cooking, sleeping, outdoor activity. Moreover, multiple activities can be freely chosen or performed concurrently.

2. **Simulation of the user normal behavior**.

   By using the task model simulator (which is a component enable in CTT that simulates all the possible user behavior based on the temporal operations among activities in the task model), we obtain 10000 normal sequences for the first round and 100 sequences for the second round of simulation. Afterward, the simulator allocates a timestamp ($t \in [T_s, T_e]$) to each activity in each sequence.

   The normal sequences Σ serve as a ground truth for the system performance validation. Thus, an activity sequence σ on Σ can be formally express as a triple (σ, T_s, T_e), where: $T_s <= T_e$ and T_s is the starting time, and T_e is the ending time, and $\sigma =< (A_1, t_1), (A_2, t_2), ..., (A_n, t_n) >$ is an ordered sequence of activities such that $A_i \in \Sigma$ for all $i = 1, ..., n$, and $t_i <= t_{i+1}$ for all $i = 1, ..., n-1$, and $T_s <= t_i <= T_e$ for all $i = 1$ to n.

3. **Simulation of the user anomalous behavior.** We generated the anomalous sequences (Σ_1) obtained by manipulating the Σ and applying one of the anomalies described in section 4.1.1. To this aim, the anomalous sequences simulator takes as input all normal sequences in Σ and from each sequence, $\sigma \in \Sigma$ selects randomly an activity and applies one of the anomalies. Meanwhile, the duplicated sequences are deleted.

   So, given $\sigma \in \Sigma$ and $\sigma_1 \in \Sigma_1$, the generator produces the anomalous sequences as follow:

   (a) *Less*: Randomly omits an elementary task/event from each sequence in Σ.

   Given $\sigma \in \Sigma : \sigma =< P_1 \circ T_1 \circ P_2 >\Rightarrow \sigma_1 \in \Sigma_1 : \sigma_1 =< P_1 \circ P_2 >$.

(b) *Difference-Early-time*: Randomly chooses an elementary task/event from each sequence in Σ and anticipates task time one minute before its starting time (τ_s). The change is made in such a way to still respect the order of the tasks.

Given $\sigma \in \Sigma : \sigma =< P_1 \circ T_1^{t_1} \circ T_2^{t_2} \circ P_2 >\Rightarrow \sigma_1 \in \Sigma_1 : \sigma_1 =< P_1 \circ T_1^{t_1} \circ T_2^{t_{2-1}} \circ P_2 >$.

(c) *Difference-Later-time*: Randomly chooses an elementary task/event from each sequence in Σ and brings the task time one minute after its final time (τ_f) with respect to the order of the tasks.

Given $\sigma \in \Sigma : \sigma =< P_1 \circ T_1^{t_1} \circ T_2^{t_2} \circ P_2 >\Rightarrow \sigma_1 \in \Sigma_1 : \sigma_1 =< P_1 \circ T_1^{t_1} \circ T_2^{t_{2+1}} \circ P_2 >$.

(d) *Difference-Time-order*: Randomly chooses an elementary task/event from each sequence in Σ and changes the order and the time of the task.

Given $\sigma \in \Sigma : \sigma =< P_1 \circ T^{t_2} \circ P_2 >\Rightarrow \sigma_1 \in \Sigma_1 : \sigma_1 =< P_1 \circ P_2 \circ T^{t_3} >, t_3 \notin [\tau_s, \tau_f]$.

(e) *Difference-Order*: Randomly chooses an elementary task/event from each sequence in Σ and changes its order with respect to the sequence Σ in a way that the task time remains in its predefined time interval.

Given $\sigma \in \Sigma : \sigma =< P_1 \circ T_i^{t_1} \circ T_j^{t_2} \circ P_2 >\Rightarrow \sigma_1 \in \Sigma_1 : \sigma_1 =< P_1 \circ T_i^{t_1} \circ P_2 \circ T_j^{t_2} >$

(f) *More-number*: Chooses randomly one elementary task/event from each sequence in the Σ and creates another instance of the same elementary task that is located in a position allowed by the task model.

Given $\sigma \in \Sigma : \sigma =< P_1 \circ T_i^{t} \circ P_2 >\Rightarrow \sigma_1 \in \Sigma_1 : \sigma_1 =< P_1 \circ T_i^{t_1} \circ T_i^{t_2} \circ P_2 >, t \in [t_1, t_3]$.

(g) *More-order*: Choose randomly one elementary task/event from each sequence in Σ and creates another instance of the same elementary task that is located in a position not allowed by the task model.

Given $\sigma \in \Sigma : \sigma =< P_1 \circ T_i \circ P_2 >\Rightarrow \sigma_1 \in \Sigma_1 : \sigma_1 =< P_1 \circ T_i \circ P_2 \circ T_i >$.

Later, to show that the algorithm is capable of detecting different types of anomalies that occur at different times, the generator introduces another type of anomaly, thus obtaining sequences containing two or more types of anomalies (Σ_2). For this aim, the generator uses the generated anomalous sequences in Σ_1, chooses the anomalous task from each sequence and duplicates, omits or changes the event order and time to generate another type of anomalous sequences Σ_2.

4. **Execution of the Anomaly Detection Algorithm over the simulated data.** For each generated sequence of activities (σ, σ_1 and σ_2), the simulator feeds the anomaly detection algorithm with one activity per time. The output of the anomaly detection is thus a sequence of responses, one per each input activity.

5. **Validation.** The core of the Deviation Analysis module is the anomaly detection algorithm that analyses one by one of the incoming activities to detect any deviation from the user behavior as defined in the CTT task model. Thus, the anomaly detection algorithm operates in real-time and indicates any potential deviation at the time of occurrence, based on the activities already received. In other words, the anomaly detection operates on the prefixes of the entire sequence of activity caused, along with a time frame (e.g., one full day or one week, as defined in the CTT graph model), by the user.

   For this reason, the dynamic behavior of the algorithm on a prefix can be different from that one on the entire sequence because an anomaly that can be correctly classified by the analysis of an entire sequence may be temporarily misclassified based on a prefix of the sequence. In order to assess the extent of such temporary misclassification, we simulate the execution of the algorithm over the simulated sequences (both correct and with anomalies), as defined at point 4 of Section 4.2.2.

   The resulting output sequences are used to construct confusion matrices and calculate the algorithm performance measures. We measured sensitivity and specificity to evaluate algorithm performance. Sensitivity is defined as the capacity of the system for correctly identifying true abnormal tasks. While specificity is defined as the capacity of the system for not generating false positives. These values were calculated as in Equation (4.12) and (4.13).

$$Sensitivity = \frac{TP}{FN + TP} \tag{4.12}$$

$$Specificity = \frac{TN}{FP + TN} \tag{4.13}$$

   Actually, Positive/Negative means that the model predicts that the sequence corresponds to abnormal/normal and True/False means that the prediction is right/wrong.

True positives (TP) are the number of case where abnormal sequences are correctly detected and classified by the system. False negatives (FN) are the number of cases where normal sequences classified as abnormal. True negatives (TN) are the number of cases where normal sequences are correctly detected by the algorithm. False positives (FP) are the number of cases where abnormal sequences are falsely detected as normal.

Another important measure is False Positive Rate (FPR), which is defined as the number of false positive instances (FP) divided by the number of False Positives (FP) and True Negatives (TN). A hight FPR can significantly decrease the utility of the anomaly detection system. Finally the accuracy of the algorithm is calculated as in (4.14).

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.14}$$

4.2.2.1 Simulation Results for Short-term Analysis

The performance of the proposed system is shown in Table 4.1 and the experimental result of our Anomaly Detection Algorithm based on a one-day daily routine is shown in Table 4.2. The experimental results showed that our online Anomaly detection Algorithm has an accuracy of 95% and a false positive rate of 2%.

Table 4.1: Experimental results with the simulation round one

System detection / Simulated	Abnormal	No Anomaly
Abnormal	$TruePositive : 18850$	$FalsePositive : 150$
No Anomaly	$FalseNegative : 100$	$TrueNegative : 9900$

The algorithm detects the tasks that triggered the anomaly and indicates the anomaly type (Equation 4.1 - 4.11) and the time of the occurrence.

Table 4.3: Experimental results with the simulation round two

Simulated \ System detection	Abnormal	No Anomaly	Another Anomaly
Abnormal	689	6	29
No Anomaly	3	100	

Table 4.4: Deviation analysis performance summary

Measure	Rate
Sensitivity	95%
Specificity	97%
FPR	3%
Accuracy	95%

Table 4.2: Algorithm performance summary

Measure	Rate
Sensitivity	99%
Specificity	98%
FPR	2%
Accuracy	95%

4.2.2.2 Simulation Results and Discussion for Long-term Analysis

The results of the simulation are presented in Table 4.3. The first line of the table shows the behavior of the algorithm when a given anomaly A is generated in a sequence. It reports the number of cases in which anomaly A is correctly classified (true positives), the number of cases in which the anomaly is not detected (false negatives), and the number of cases in which the anomaly A is misclassified as another anomaly A' (false negatives). Similarly, the second line shows the behavior of the algorithm for the sequences that do not contain an anomaly. In this case, it reports the number of cases in which an anomaly is detected (false positives) and the number of cases in which no anomalies are detected (true negatives).

The performance of the simulated Deviation Analysis module is shown in Table 4.4. The FPR (false positive rate) shows the proportion of all the cases in which abnormal

activities have been identified as a normal one. Since each anomaly may have, in general, different interpretations, the accuracy of the anomaly detection algorithm can not reach 100%. The accuracy of the algorithm has been calculated as 95%.

As mentioned earlier, the classification of an anomaly may change during the time (as the prefix sequence evolves with the occurrence of each new observation), as the algorithm progressively converges towards the proper classification. For example, assume that a user forgets to perform an activity A. The algorithm correctly detects that A has been omitted (thus it outputs "Less"), but later, after the occurrence of some other activities, the user performs A. In this case, the proper classification would be *Order* (i.e., the activity happens in the wrong order), but this output can be reached by the algorithm only when A actually occurs. Hence, the correct classification is output by the algorithm with a delay (latency), during which the anomaly had been misclassified. Note that, latency is particularly interesting because it indicates the time necessary for the algorithm to converge towards the correct classification.

For this reason, we evaluate the latency of the anomaly detection algorithm. Specifically, latency is defined as the time elapsed between the time of occurrence of an anomaly and the time in which the anomaly detection algorithm outputs the correct classification. In the simulation, we calculate the average latency in the cases in which there is a temporary misclassification, along with the confidence interval to show the range of its variation. The results show with 95% confidence that the average latency in the cases where there is a misclassification is between 34.9 and 57.9.

On the one hand, when we simulate the user behavior, the anomaly detection is not based on the initial set of sequences from the real world, but we are adding potential failures with synthetic generation. So, an elderly person that misbehaves may acts differently from how the misbehaviors are simulated. On the other hand, having a larger scale validation with the real data from the older adults is expensive since requires a lot of effort in the deployment and maintenance of the sensor's infrastructure.

4.3 Chapter Summary

The present chapter provides a novel framework for a wellness determination process as it verifies the behavior of elderly people at different stages of daily living in a smart home monitoring environment. The anomalous behavior is determined using the values for the behavioral attributes within a specific context. A data instance might be a contextual anomaly in a given context, but an identical data instance (in terms of behavioral attributes) could be considered normal in a different context. This property is key in identifying contextual and behavioral attributes for a contextual personalized anomaly detection technique.

This thesis applies the idea of user profiling analysis using the task model to the anomaly detection systems. We developed a discriminative algorithm that compares the expected behavior expressed in such models with the sequence of events generated by the real user. For achieving this result, we had to simulate the event sequences received from the Context Manager (which represents the user's daily activities). Then, we validated the algorithm performance by sending the simulated normal and anomalous sequences to the Deviation Analysis module. The experiment results are promising.

Chapter 5

Personalized Persuasive Interventions

Traditionally, persuasion has meant “human communication designed to influence the autonomous judgments and actions of others” [221]. New technologies create opportunities for persuasive interaction because we can reach the users easily. Since persuasion is described as changing the attitudes and/or behavior of others, the persuader is often trying to convince the persuadee of something which relies primarily on symbolic strategies that trigger the emotions. Persuasive technology is a great help to motivate people toward healthy behavior, and thereby possibly delay or even prevent health problems.

Our persuasion module consists of a fuzzy rule-based system that implements a decision-making model to recognize the true anomalies and an intelligent suggestion engine, StreamingBandit (see [217]), the design of which was inspired by the contextual multi-armed bandit (cMAB) problem [218].

We introduce an approach that enables configuration of interventions targeting specific anomalous behavior in compliance with the JITAI framework presented in [19]. Besides, our design mechanism utilizes the reinforcement learning methods for optimization/personalization of intervention delivery in real time by modeling it as a contextual bandit problem.

5.1 Background

In this section, we define some concepts that are required to better understand the rest of the chapter. Section 5.1.1 introduces some basic concepts and structure of fuzzy logic systems and Section 5.1.2 presents some background concept on (contextual) multi armed-bandit formalization.

5.1.1 Fuzzy set theory and logic

Fuzzy Logic resembles the human decision-making methodology and deals with vague and imprecise information. It is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic. To better understand how fuzzy expert systems work, the main concepts of fuzzy sets and logic are briefly described in the following paragraphs.

5.1.1.1 Fuzzy Set Membership Functions

Membership functions were first introduced in 1965 by Lofti A. Zadeh in his first research paper “fuzzy sets” [222]. It can be defined as a technique to solve practical problems by experience rather than knowledge and are represented by graphical forms A membership function for a fuzzy set A on the universe of discourse U is defined as $\mu A : U \rightarrow [0, 1]$, where each element of U is mapped to a value between 0 and 1. This value, called membership value or degree of membership, quantifies the grade of membership of the element in U to the fuzzy set A.

Membership functions allow us to graphically represent a fuzzy set. The x axis represents the universe of discourse, whereas the y axis represents the degrees of membership in the $[0, 1]$ interval. They are divided into 3 categories as follows.

Below is a list of the membership functions we will use in the section 5.4 of this thesis.

- Triangular function: defined by a lower limit a, an upper limit b, and a value m, where $a < m < b$.

- Trapezoidal function: defined by a lower limit a, an upper limit d, a lower support limit b, and an upper support limit c, where $a < b < c < d$.

  There are two special cases of a trapezoidal function, which are called R-functions and L-functions:

  - R-functions: with parameters $a = b = -\infty$
  - L-Functions: with parameters $c = d = +\infty$

5.1.1.2 Fuzzy Operators

Similarly to classical set theory, fuzzy sets can be combined in different ways through operations (i.e., intersection, union, complement, etc.) to produce another set. Having two fuzzy sets $\widetilde{A}$ and $\widetilde{B}$, the universe of information U and an element y of the universe, the following relations express the union, intersection, and complement operation on fuzzy sets.

- Union/Fuzzy 'OR': Let $\mu_{\widetilde{A}}$ and $\mu_{\widetilde{B}}$ be membership functions that define the fuzzy sets $\widetilde{A}$ and $\widetilde{B}$, respectively, on the universe U. The union of fuzzy sets $\widetilde{A}$ and $\widetilde{B}$ is a fuzzy set defined by the membership function as below.

$$\mu_{\widetilde{A}\cup\widetilde{B}}(y) = \mu_{\widetilde{A}} \vee \mu_{\widetilde{B}} \quad \forall y \in U \tag{5.1}$$

  Here $\vee$ represents the 'max' operation.

- Intersection/Fuzzy 'AND' Let $\mu_{\widetilde{A}}$ and $\mu_{\widetilde{B}}$ be membership functions that define the fuzzy sets $\widetilde{A}$ and $\widetilde{B}$, respectively, on the universe U. The intersection of fuzzy sets$\widetilde{A}$ and $\widetilde{B}$ is a fuzzy set defined by the membership function as represented below.

$$\mu_{\widetilde{A}\cap\widetilde{B}}(y) = \mu_{\widetilde{A}} \wedge \mu_{\widetilde{B}} \quad \forall y \in U \tag{5.2}$$

  Here $\wedge$ represents the 'min' operation.

- Complement/Fuzzy 'NOT' Let $\widetilde{A}$ be a membership function that defines the fuzzy set $\widetilde{A}$, on the universe U (as you see in Equation 5.3).

$$\mu_{\widetilde{A}} = 1 - \mu_{\widetilde{A}}(y) \quad y \in U \tag{5.3}$$

In the fuzzy theory, there are two basic classes of binary operators of particular interest that generalize intersection and union: triangular norms (t-norms) and conorms.

- t-norms are used to represent intersection in fuzzy set theory and conjunction in fuzzy logic, and are specified by a function $t : [0,1] \times [0,1] \rightarrow [0,1]$. Some examples of t-norms are the minimum $min(a,b)$ and the product $prod(a,b) = a \cdot b$.
- t-conorm: it is a binary operation that represent the union of two fuzzy sets ($S : [0,1] \times [0,1] \rightarrow [0,1]$). Some examples of t-conorms are the maximum $max(a,b)$, the probabilistic sum or sum-product $sum - prod(a,b) = a + b - a \cdot b$.

5.1.1.3 Mamdani's Fuzzy Inference Method

A Fuzzy Inference System (FIS) is a way of mapping an input space to an output space using fuzzy logic. A FIS tries to formalize the reasoning process of human language utilizing employing fuzzy logic (that is, by building fuzzy IF-THEN rules). For instance:

"If the $criticality$ is high, OR the $\delta_{anomaly}$ is high, Intervention level is high (the caregiver should receive an alert.)"

FIS is used to solve decision problems, i.e. to make a decision and act accordingly. In general, a fuzzy inference system consists of four modules (see section 5.4, figure 5.2), Fuzzification module (transforms the system inputs, which are crisp numbers, into fuzzy sets.), Knowledgebase (stores IF-THEN rules provided by experts), Inference engine (simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules) and Defuzzification module (transforms the fuzzy set obtained by the inference engine into a crisp value).

Mamdani's Fuzzy system was proposed in 1975 by Ebhasim Mamdani. Basically, it was anticipated to control a steam engine and boiler combination by synthesizing a set of fuzzy rules obtained from people working on the system. Mamdani's method is the most commonly used in applications, due to its simple structure of 'min-max' operations. Following steps need to be followed to compute the output from this FIS:

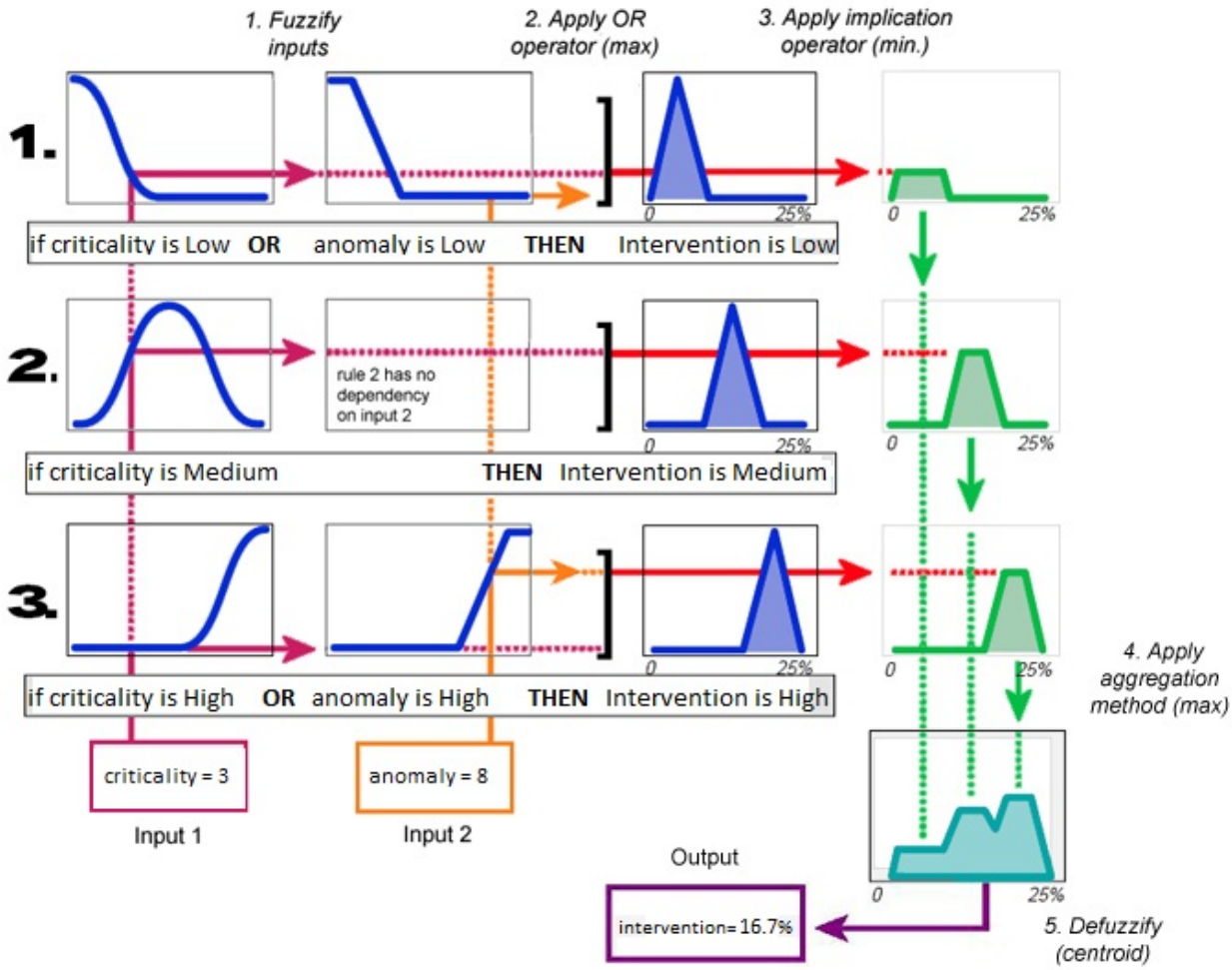


Figure 5.1: Block diagram of Mamdani Fuzzy Interface System

1. Set of fuzzy rules need to be determined in this step.

2. Evaluate the antecedent for each rule: In step 1, by using the input membership function, the input would be made fuzzy. Given the inputs (crisp values) we obtain their membership values. This process is called 'input fuzzification'. If the antecedent of the rule has more than one part, a fuzzy operator (t-norm or t-conorm) is applied to obtain a single membership value. When fuzzifying the first part of the antecedent (the $criticality$ is high) we obtain the degree to which the criticality is high if we rate it as a 3. As we can see, a 3 rating stands for a low criticality, that is why we obtain the membership value 0. When fuzzifying the second part of the antecedent ($delta - error$ is high) we obtain the degree to which the $\delta_{anomaly}$ is high if we rate it as an 8. Naturally, an 8 rating stands for a quite high $\delta_{anomaly}$, that is why we obtain the membership value 0.7.

   Lastly, as the two parts of the antecedent are joined by a disjunction ($criticality$

is high, OR the $\delta_{anomaly}$ is high), we apply an OR operation, the maximum, to both membership values to obtain the membership value 0.7. Let's suppose both parts of the antecedent were joined by a conjunction ('and'). In this case, we would have to apply an AND operation, the minimum for instance.

3. Obtain each rule's conclusion: Given the consequent of each rule (a fuzzy set) and the antecedent value obtained in the previous step, a fuzzy implication operator will be applied to obtain a new fuzzy set. Two of the most commonly used implication methods are the minimum, which truncates the consequent's membership function, and the product, which scales it. In figure 5.1, the minimum operator is used: Now establish the rule strength by combining the fuzzified inputs according to fuzzy rules.

4. Aggregate conclusions: In this step, the outputs obtained for each rule in the previous step (obtain conclusion) are combined into a single fuzzy set, using a fuzzy aggregation operator.

   Some of the most commonly used aggregation operators are the maximum, the sum and the probabilistic sum. In figure 5.1, the maximum is used: In this step, determine the consequences of rule by combining the rule strength and the output membership function.

5. Defuzzification: Finally, a defuzzified output distribution is obtained. When we try to solve a decision problem, we want the output to be a number (crisp value) and not a fuzzy set. So, we need to transform the fuzzy set we obtained in step 4 into a single numerical value. One of the most popular defuzzification methods is the centroid, which returns the center of the area under the fuzzy set obtained in step 3.

5.1.2 Multi Armed Bandit

Multi-armed bandit is a simple but very powerful framework for algorithms that make decisions over time under uncertainty. The term "multi-armed bandit" comes from a hypothetical experiment where a person must choose between multiple actions (i.e. slot

machines, the "one-armed bandits"), each with an unknown payout. The goal is to determine the best or most profitable outcome through a series of choices. At the beginning of the experiment, when odds and payouts are unknown, the gambler must determine which machine to pull, in which order and how many times. This is the multi-armed bandit problem.

5.1.2.1 Multi-Armed Bandit Algorithm

There are many different solutions that computer scientists have developed to tackle the multi-armed bandit problem. Below is a list of some of the most commonly used multi-armed bandit solutions:

- Epsilon-Greedy

  This is an algorithm for continuously balancing exploration with exploitation. (In ‘greedy’ experiments, the lever with highest known payout is always pulled except when a random action is taken). A randomly chosen arm has pulled a fraction of ϵ of the time. The other $1 - \epsilon$ of the time, the arm with highest known payout is pulled.

- Thompson Sampling (Bayesian): With this randomized probability matching strategy, the number of pulls for a given lever should match its actual probability of being the optimal lever.

- Upper Confidence Bound: This strategy is based on the Optimism in the Face of the Uncertainty principle and assumes that the unknown mean payoffs of each arm will be as high as possible, based on observable data.

5.1.3 What is a Contextual Bandit?

In a real-world scenario, we sometimes have data that can help inform decision-making when choosing between the various actions in a multi-armed bandit situation. This information is the "contextual bandit", the context and environment in which the experiment occurs.

In remote care applications, contextual bandits rely on incoming user context data as it can be used to help make better algorithmic decisions in real-time. For example, we can use a contextual bandit to select an intervention to display to the user. The context is any historical or current information about the user, such as user anomalous behavior, user location, user daily routine, etc.

The problem comes from an iterative process generating data as follows:

At each round, the world creates an observation consisting of a set of covariates X of fixed dimension and a vector of rewards r (which are stochastic but dependent on the covariates) of a length corresponding to m, the number of arms. An agent must choose an arm or label for the observation among the set of m arms. The world reveals the reward for the arm chosen by the agent, but not for the other arms. The purpose is of course to build a policy that would maximize the rewards obtained by the agent. The arms might also expire over time and new arms might appear too, leading to the same exploration-exploitation dilemma faced in multi-armed bandits.

5.2 Persuasion Strategy

Persuasive strategies are techniques that can be employed in PTs to motivate behavior and/or attitude change. Over the years, a number of strategies for persuading people to perform the desired behavior have been developed. For example, Fogg [176] developed seven persuasive tools, and Oinas-Kukkonen [177] built on Fogg's strategies to develop 28 persuasive system design principles. In addition, the six persuasive strategies developed by Cialdini – Reciprocity, Scarcity, Authority, Commitment and Consistency, Consensus and Liking – are among the oldest and most widely employed strategies[223]. The six principles are as follow:

- Reciprocity: Cialdini's rule of reciprocation is built on a sense of obligation based on a previous favor or gift. People by their nature feel obliged to return a favor and to pay back others. Thus when a persuasive request is made by a person the receiver feels indebted to, the receiver is more inclined to adhere to the request [223].

- Scarcity: People tend to place more value on things that are in short supply. This is due to the popular belief that fewer available options are of higher quality.
- Authority: People defer to experts. Therefore, individuals are more likely to comply with a request when it is made by a person or people they perceived as possessing high levels of knowledge, wisdom, or power.
- Commitment and Consistency: People by their nature strive to be consistent with previous or reported behavior to avoid cognitive dissonance. "commitments are most effective when they are active, public, effortful, and viewed as internally motivated (uncoerced)" [223]
- Liking: People can be easily influenced or persuaded by someone they like. Factors such as similarity, praise, and attractiveness can reliably increase the effectiveness of the liking strategy.
- Social Proof: Looking at those around oneself to determine what to believe or how to act in a situation. We often observe the behaviors of others to help us make decisions. Most influential: when in uncertain or ambiguous situations and when others are viewed as similar to oneself.

5.3 Persuasive Interventions

Interventions using short messages may be the most effective as a reminder system to support behavioral management [224]. We create several short messages that implement different social influence strategies as defined in [225] to be delivered to individual users [226]. We propose to use three of these six principles in our system, i) *reciprocity* that by providing something of value, we create an emotional attachment and increase the desire to cooperate to achieve mutual goals (e.g., listen to this relaxing song and go to sleep, including a link to the song); ii) *social proof* because it has the effect that they make decisions only after observing the behaviors and consequences of those around them. (e.g., 70% of the healthy Adolescent Populations go to sleep on a regular time). This principle is very similar to the relational concept of reputation [227]. It is particularly useful in the absence of the ability to make a sound decision in which individuals

look for ways to reduce cognitive search costs. iii) *authority* that expects individuals will take a shortcut to decision-making by deferring to the judgment to an authority (e.g., your doctor says that sleeping late increases your stress level.). Given these different possible intervention messages, and the anomaly detection module described before, the persuasion module can be formalized as a contextual MAB problem and addressed using StreamingBandit.

To improve the quality of the content of intervention messages, we organized a workshop in which we explained these three principles to the experts in ambient assisted living, and then we asked them to sort the messages accordingly. Later, those that were hard to sort or wrongly sorted was omitted. Appendix A contains the list of messages created in this workshop.

5.4 Intervention Level Classification Using Fuzzy Rule-Based System

As we mentioned in section 3.2.4, we have developed a Mamdani-type Fuzzy Rule-Based System that implements a decision-making model presented in Figure 5.2. The purpose of this module is to recognize the true anomalies with two inputs, a decision rule base (including all fuzzy rules for decision-making) and one output in charge of selecting the intervention level. In fuzzy systems, things are assumed to be true to some degree, and simultaneously false to some degree, where, by mutual agreement, a numerical value between (or including) 0 and 1 is arbitrarily assigned to represent that degree (i.e., degree of membership, $\mu A : X \rightarrow [0, 1]$). Using this information as a starting point, we aim to infer the degree of the abnormality detected which forms a continuous variable with values from 0 to 1 that allows us to distinguish between situations characterize as "No intervention", "Mild Intervention" and "Strong Intervention" needed. The followings are the three main components of the model for intervention level selection.

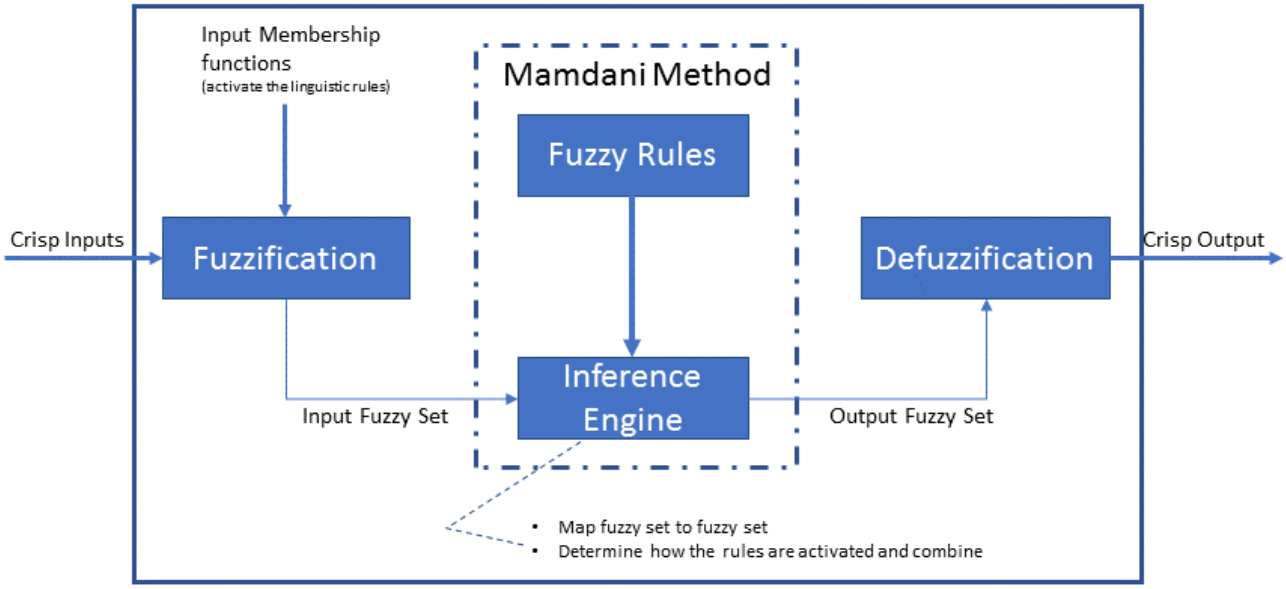


Figure 5.2: Mamdani-type Fuzzy Rule-Based System

5.4.1 Fuzzification

The variables presented in our model restricted to i) the category of the anomalous activity performed by the user (e.g., cooking, taking medicine, sleeping, toileting, etc.); ii) the criticality level of accomplished activity and iii) the error measure ($\delta_{anomaly}$).

5.4.1.1 Inputs

Amongst the possible elderly activities (for the first input), we only consider "sleeping" and "taking medicine". For the second input, which is the criticality level of the task, the information is available in the user task model. On the other hand, the third input has been calculated as explained in the following section 5.4.1.1.1.

5.4.1.1.1 Error Measurement

When anomalies have been detected, the next step is to measure the error ($\delta_{anomaly}$), which corresponds to the difference between the detected anomalous activity and the planned activity in terms of time, the number of repetitions and the order of the activity. This Delta error measure, along with the information about the user context allows our system to distinguish the level of intervention. We calculated the Delta errors as follows:

- δ_{time}: In case of having *Difference-Early-Time*, *Difference-Later-Time* (i.e. an activity has been performed before or after the planned time), and *Less* anomaly, we subtract the time in which the anomaly has been detected from the time the planned activity should have been happened and we will save it as δ_{time}.

- δ_{order}: In case of having *Difference-Order* anomaly (i.e., activity has been performed in the wrong temporal order), we construct the shortest path from the starting activity(ies) planned in the task model to the anomalous activity occurred. Then, we subtract the anomalous activity index in the current partial sequence (i.e., the sequence of activities that have taken place) from the anomalous activity index in the new sequence (i.e., constructed via shortest path permissible by the task model). We save the result as δ_{order}.

- δ_{more}: In case of having *More* anomaly (user performed the activity more times than expected one), the number of repetition has been considered δ_{more}.

In case of having *Difference-Order-Time* and *More-Order*, we consider δ_{time}, δ_{order} and δ_{more} together to calculate the degree of anomaly. Moreover, each time a new anomaly occurs, all the above Delta errors will be updated based on the new position of the anomalous activity in the currently received sequence P. Also, in case of having *Less*, *More-Order* and *Difference-Order*, based on the criticality level of the anomalous activities in P, namely low, medium and high (defined in task model) the δ_{time} measure gets updated relatively each 10, 30 and 60 minutes.

All input variables (except activity category) are called linguistic variables that later will be separated into linguistic values. To determine the fuzzy sets we used triangular and trapezoidal membership functions (MFs) [228]. To define the membership function for the Criticality level, a range from 0 to 10 has been considered that has three input sets defined as Low criticality (Lc), Medium (Mc) and High (Hc).

In case of having a $Time$ anomaly (δ_{time}), a range from 0 to 120 is used to describe the degree of the Time anomaly. Within this range, 0 represents the lowest error, while 120 defines the most undesirable situation. This input has five fuzzy sets defined as follows: very low (VLt), Low (Lt), Medium (Mt), High (Ht), and Very Hight (VHt). In case of having $Order$ anomaly, δ_{order} has a range from 0 to 10 where 0 defines the minimum

distance and 10 represents the highest distance compared to the user task model. It has three fuzzy sets defined as follows: Low (Lo), Medium (Mo) and High (Ho) order. The δ_{more} range is the same as δ_{order} with the difference that the range numbers represent the number of repetitions.

The membership functions for both inputs (i.e., criticality level and the $\delta_{anomaly}$) are shown in Figure 5.3 where every set is depicted as described previously.

5.4.2 Decision Rules Base

Returning to our scenario, let us assume that Sarah visits a physician and describes her situation as follows: *Doctor, I should have taken my medicine at 9:00 with a full stomach to keep my blood sugar in balance. As I had an appointment for doing a simple checkup this morning, I forgot to take my medicine and now it is 2 hours after my prescribed time. I am not feeling good.*

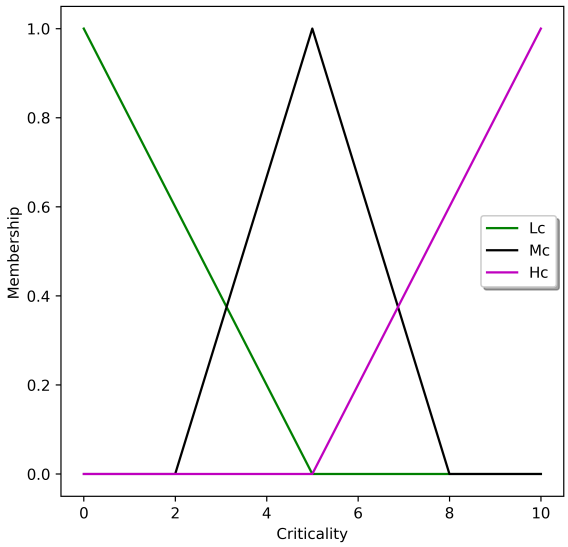
In this situation, the expert tries to guess how critical the patient's state is and which level of treatment should be chosen. He gives his opinion by considering the activity type, the amount of delayed time for performing that activity (δ_{time}) and the criticality level of this activity for this specific user. So, if taking the medicine is very critical for Sarah and, she is late for her medicine, the doctor considers a strong level of treatment for her (e.g., prescribing a new medicine or suggesting a clinical service). In our model, those opinions (i.e., expert opinion) are converted to fuzzy rules not only to choose the most proper level of intervention for the elderly but also to avoid sending false alerts to healthcare providers or the elderly themselves, if the anomalous activity is not truly abnormal.

We model our fuzzy system with conventional rules based on the structure:

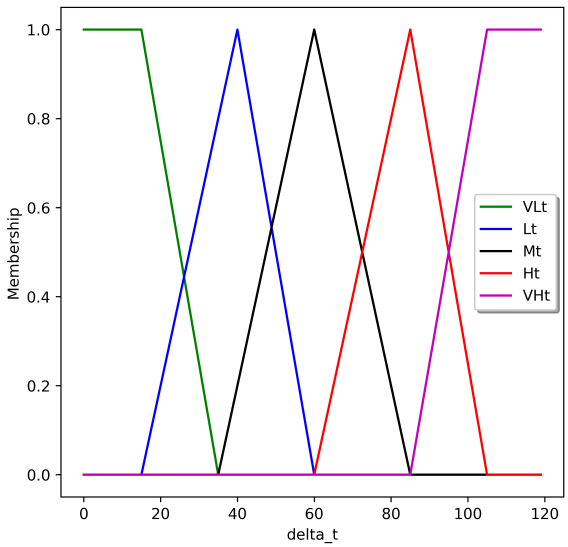
if $< antecedents >$ then $< consequent >$. The structure of Mamdani-type fuzzy logic rule is expressed as follows:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_1 \ AND....AND \ x_n \ is \ A_n \\ &THEN \ y \ is \ B. \end{aligned} \tag{5.4}$$

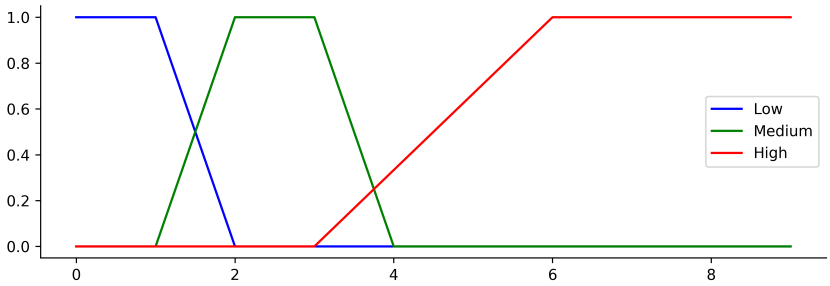
Where x_i (i=1, 2,..., n) are input variables and y is the output variable. $A1, ..., An$ and B



(a) Criticality MFs



(b) Delta Time MFs



(c) Delta Order MFs

Figure 5.3: Membership Functions of System Inputs

define the fuzzy subsets (membership function distributions, conventionally expressed in linguistic terms like Low, Medium, High, etc.) of the corresponding input and output variables, respectively.

In our rules structure shown in Equation 5.4, the aggregation of the membership values (i.e., the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set in order to make a decision) is performed by using MAX which by far is the most common implementation of the rule aggregation operation [229]. According to the *MAX* procedure, the final fuzzy output is calculated from the set of single outputs taking the maximum truth value where one or more terms overlap. Thereafter, the new combined fuzzy set representing the outcome for the output variable "Intervention level evaluation" is ready for the last defuzzification process.

5.4.3 Defuzzification

The fuzzy system output provides information about which intervention level needs to be considered when applying the personalized interventions for a user. The output is defined in a discourse Universe between zero and one, where zero indicates the lowest degree of an anomaly which requires no intervention, and one is the highest degree of the anomaly which requires the strongest intervention.

Defuzzification selects the appropriate action based on the fuzzy recommendation. The input for the defuzzifier is a fuzzy set (the aggregated output fuzzy set) and the output is a single number coming from the max-product of the output areas of each rule. However, the rule itself may be fuzzy, which means the strength of the recommendation depends on the rule strength expressed as the membership of y in B in the Equation 5.4. The strength of the rule models differences in statements like " No Intervention", "Mild Intervention" and "Strong Intervention". For the current case, the final output is the intervention level needed by the elderly. Later, the output is interpreted by the persuasion module as the different kinds of interventions required for the individual user. The centroid method (also called center of area or center of gravity), which is the most popular defuzzification method [230, 231] and which returns the center of the area under the curve, is used by the defuzzifier to estimate the final intervention level.

The fuzzy system output provides information about which intervention level needs to

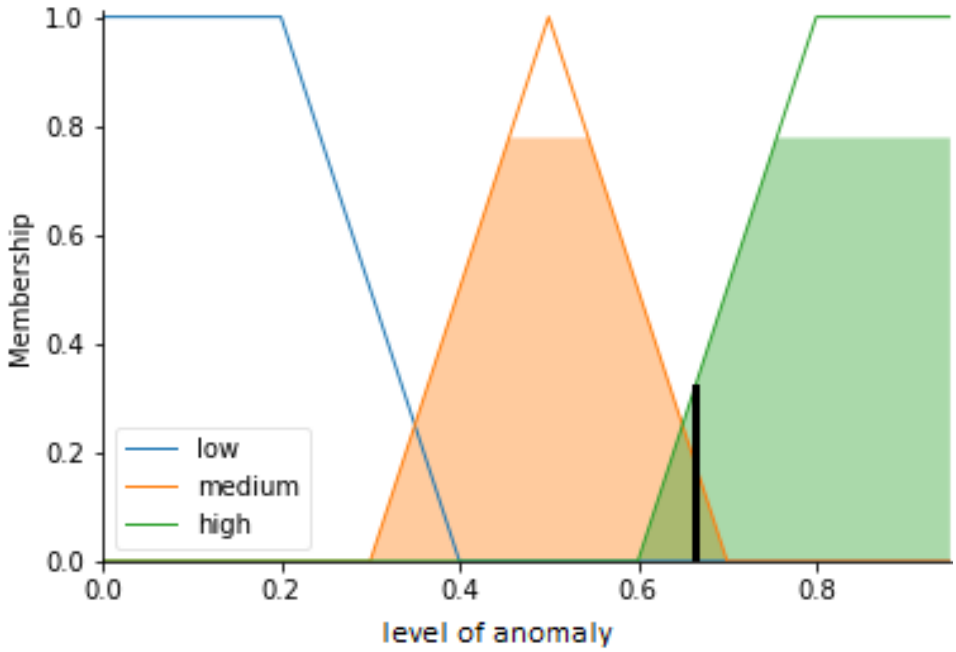


Figure 5.4: Defuzzification

be considered when applying the personalized interventions for a user. The output is defined in a discourse Universe between zero and one, where zero indicates the lowest degree of an anomaly which requires no intervention, and one is the highest degree of anomaly which requires the strongest intervention. For example, if the input of the system is $criticality$ = 8 and δ_{time} = 75, the final output of the system is a crisp value "0.66" as shown in Figure 5.4. According to the defined output membership functions, this calculated value "0.66" represents the degree of the detected anomaly which corresponds to the "Strong intervention".

Therefore, in case the result is "no intervention", the persuasion module does not issue any message. On the other hand, if the persuasion module receives "Mild Intervention" and "strong intervention", it informs the elderly and the health professionals relatively for further support. Thus, in the following section, an intervention message will only be sent to the older adults when a "Mild intervention" level is encountered.

5.5 Formalize Persuasive Interventions

After the true anomalies have been detected and the level of intervention for each anomaly has been identified, the persuasion module suggestion engine —by using the in-

tervention messages and the well-known sequential decision-making formalization, contextual multi-armed bandit (cMAB)— provides an abstraction that allows for the implementation of multiple decision strategies to select persuasive messages in our context. The module can be used to dynamically learn from user behavior by suggesting actions that maximize the chances of losing bad habits in older adult's behavior. The suggested actions in our work are text-messaging interventions.

5.5.1 Contextual Multi-Armed Bandit

The cMAB problem is a well-known sequential decision-making problem and its formalization provides an abstraction that allows for the implementation of multiple decision strategies to select persuasive messages based on the true anomalies detected in our context. The module can be used to dynamically learn from user behavior by suggesting actions that maximize the chances of losing bad habits in older adults' behavior. The suggested actions in our work are text-messaging interventions that are defined based on the six persuasive principles as described by Cialdini [225].

In abstract terms, a contextual multi-armed bandit problem concerns a sequence of interactions in which an agent chooses, based on a given context, an action from a set of possible actions such that the total reward of the chosen actions is maximized. The rewards depend on both the chosen action and the context. A basic formalization of the cMAB problem is as follows: At each interaction t, the agent observes a context x_t and consequently chooses an action (arm) a_t to receive reward r_t. The aim of the agent is to maximize the cumulative reward, $\sum_{t=1}^{T} r_t$, where T denotes the total number of interactions. To achieve this aim, the agent employs a policy π which is a function that takes the context x_t and the historical interactions (x', a', r'), and returns an action. A personalized persuasive system can easily be described using this formalization: the user's current states correspond to the "context", the personalized persuasive interventions correspond to the different "actions", and the users' performance on future activities correspond to the "rewards" [232]. The proposed approach works under the assumption that users can provide positive and negative feedback regarding the system actions.

5.5.1.1 StreamingBandit

StreamingBandit is an open-source RESTful web application [233] that allows formalizing sequential decision-making procedures as a cMAB problem and, by virtue of this formalization, makes it easy to experiment with different policies *in situ*. The implementation of a cMAB policy in StreamingBandit is based on two distinct steps that jointly comprise the policy itself:

1. The $summary$ step: In this step all the data are summarized into a set of parameters θ_t. Effectively, the previous state of the parameters θ'_t gets updated in each step by the new information (x_t, a_t, r_t). This step implements how the policy learns from observations that arrive over time. To ensure scalability, often $|\theta'_t| \ll |(x', a', r')|$.

2. The $decision$ step: In this step, by using the new context, x_t and the current state of the parameters θ_t, the next action a_t is selected. This step implements the decision that is made by the policy.

StreamingBandit facilitates the implementation of different summary and decision steps to compose policies. Subsequently, a policy can be directly employed in the field using the REST API provided by StreamingBandit. StreamingBandit implements a number of common policies by default, for example:

- ϵ-first [234] : For the first ϵN, where $0 < \epsilon < 1$, actions are selected uniformly at random from the set of possible actions. For the remaining $(1 - \epsilon)N$ interactions the action that attained the highest mean reward during the period of random selection is chosen. This policy effectively implements a randomized experiment (with $n = \epsilon N$), after which the action with the highest mean reward is selected.

- ϵ-greedy [235]: The best performing action—that with the highest mean reward—is selected for a proportion $1 - \epsilon$ of the interactions, and a random action is selected (with uniform probability) for a proportion ϵ.

- Thompson sampling [236]: Using a Bayesian model on the underlying parameters of the reward of each arm, and at each interaction, an arm is played according to its posterior probability of being the best arm.

- Bootstrap Thompson sampling [237]: Similar to Thompson sampling, however, in this policy the posterior distribution is approximated using an online Bootstrap distribution for computational ease.

5.5.2 Formalizing Personalized Persuasive Interventions Using StreamingBandit

To detail the design of the persuasion module we first give an example and subsequently introduce the formalism adopted in the system: consider the "sleeping" activity discussed in the previous scenario and assume that we have various sensors in the older adult's house which allow our system to measure this activity. Sleeping has different criticality levels for different users and each user might have planned to carry out this activity at a different time of the day, for a different duration or in a different order. Every day, our system observes the activities and detects whether the user deviates from the expected routine. Upon detection of a deviation, we initiate an interaction with StreamingBandit. The context that StreamingBandit receives consists of the current state of the user and the detected anomaly. Next, one of the possible persuasive messages (i.e., intervention messages created based on three persuasive principles, namely, reciprocity, social proof, or authority) is selected. Subsequently, the sensors are used to determine whether the message was successful: if the users' behavior is changed by the message, a reward is received. Looking at the Persuasion module Diagram shown in Figure 3, the scenario described above consists of the following elements:

1. An index of the interactions $t = 1..., t = T$ where t is every time that an anomaly is detected.

2. The $context$ $x_t \in X_t$ where X is a set of variables describing the current state. The context feature vector x consists of 3 variables that describe the current user, the activity type (e.g., sleeping) and intervention level respectively. In the following, and for the sake of clarity, we focus only on "mild" interventions.

3. The $action$ $a_t \in A_t$ where A is a set of possible interventions that our system can take. In our case, the action space consists of a list of persuasive messages, one for each of the three persuasive strategies that we have selected.

4. The $reward$ r_t is a (function of the) measured response at that point in time. In our system, a decrease in the day-to-day number of anomalies is considered a reward.

5. A $policy$ $\Pi : x_1, ..., x_{t'-1}, a_1, ..., a_{t'-1}, r_1, ..., r_{t'-1}$ $\rightarrow a_t$, which is a mapping from all possible interactions (their contexts, actions, and rewards) up to some point in time $t = t'$ to the next action $a_{t'}$ in a way that the cumulative reward is maximized.

In some respects, the integrated platform (i.e., the online anomaly detection and persuasion module) emulates the behavior of a physician who meets different elderly patients sequentially (at each interaction t). For each elderly, the physician observes the current condition (context x_t) and background (historical data θ') and consequently chooses the treatment (action a_t) such that the cumulative reward, measured in terms of the general health of the elderly is maximized. To choose the best treatment, which in our case is the best persuasive intervention message, different decision policies π can be implemented which take the current context and the historical interactions, and assign a new intervention message.

StreamingBandit provides a RESTful API facilitating the implementation of the aforementioned summary and decision steps which jointly allow one to implement a sequential decision policy. First, we create a so-called "experiment" within the platform, and subsequently, we implement a policy consisting of the decision step given θ and x_t and a summary step. To use StreamingBandit as a service, the persuasion module, sequentially retrieves a JSON object containing the current context—using a call we denote `getcontext` in Figure 3—and subsequently posts this context to StreamingBandit using the `getaction` call. After receiving the action, i.e., the message to show to the current user, the `getreward` call queries the users state to see if the number of anomalies has changed, and the `setreward` updates θ stored in StreamingBandit for use in future interactions. The code in the `getaction` and `setreward` steps implement the decision and summary steps, respectively. Once a decision policy is implemented in StreamingBandit it is easy to change the policy and experiment with alternatives. A more detailed simulation study using StreamingBandit is explained in section 5.6.1.

5.6 Personalized Intervention Simulation and Results

In the real world, evaluating of MAB policies is challenging and include expensive field trials. Thus, to validate our suggested "Persuasion Module" we implemented a simulation study based on the scenario introduced in 5.5.2. To do so, we first specify the data generating mechanism. Table 5.1 lists the possible values that the different variables can have in our simple scenario; in reality the number of contextual variables is much larger.

Table 5.1: Variables and possible values in the cMAB scenario describing our personalized persuasive intervention module.

Type	Variable	Values
Context	User	*0, 1*
Context	Activity	*"sleeping", "medication"*
Context	Intervention level	*"mild"*
Action	Message	*"Authority", "Social proof", "Reciprocity"*

In concordance with Table 5.1, we setup the following data generating process:

1. A user id $\in \{0, 1\}$ is generated. We assume user 0 has a higher probability of performing anomalous behavior and we select users with probabilities $\Pr(\text{user} = 0) = .6$ and $\Pr(\text{user} = 1) = .4$.

2. An activity $\in \{\text{sleeping}, \text{medication}\}$ is generated. We assume the activity is independent of the user, and each activity occurs equally often: $\Pr(\text{sleeping}) = \Pr(\text{medication}) = .5$.

3. A cMAB policy π selects an action. In this simulation study we implement k-arm Bernoulli Thompson sampling [237] where we define an arm for each possible context-action combination and we use a Beta$(1, 1)$ prior for each arm. Hence, we model each context-arm combination independently and we consider $2 \times 2 \times 3 = 12$ arms.

4. Subsequently, binary rewards are generated using the success probabilities (e.g., $Pr(r = 1|a, x)$ provided in Table 5.2. Note that user 0 responds well to Reciprocity

messages, irrespective of the activity type, while user 1 responds well to Social proof messages when a sleeping anomaly is detected, and she responds well to Authority messages when a medication anomaly is detected.

5. Finally, the observed reward is used to update θ (and thus in this case the Beta(α, β) posterior for each arm).

Table 5.2: Probabilities of rewards in data-generating process

User	Activity	Message	$Pr(r = 1)$
0	Sleeping	Authority	.1
0	Sleeping	Social proof	.1
0	Sleeping	Reciprocity	.7
0	Medication	Authority	.1
0	Medication	Social proof	.1
0	Medication	Reciprocity	.6
1	Sleeping	Authority	.2
1	Sleeping	Social proof	.6
1	Sleeping	Reciprocity	.2
1	Medication	Authority	.9
1	Medication	Social proof	.1
1	Medication	Reciprocity	.1

Figure 5.5 demonstrates the expected mean reward at each interaction $t \in \{1, \ldots, T = 200\}$ for our implemented policy. The expectation is computed over $m = 10000$ simulation runs. Note that choosing messages randomly would, in this case, produce an expected reward of $.31$, while an optimal policy—an oracle policy that always selects the action with the highest expected reward for each context—achieves an expected reward of $.69$: we indicate these using horizontal dashed lines in Figure 5.5. It is clear that over time Thompson sampling “learns” to match the correct action to the correct context: thus, over time, the selected messages are tailored to the user and the system indeed provides personalized persuasive messages. The detailed code of this simulation in the "R" language has shown in appendix B.

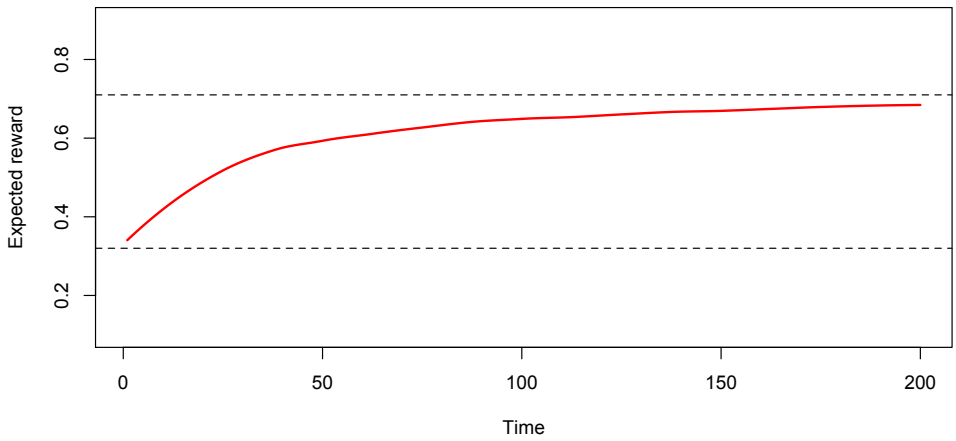


Figure 5.5: Performance of Thompson sampling for allocating personalized persuasive messages.

5.6.1 Personalized Intervention Simulation using StreamingBandit Application

A we describe in section 5.5.2, implementation of persuasive intervention in StreamingBandit consists of four steps, i) getcontext; ii) getaction; iii) getreward and, iv) setreward. Let's say we have two subjects: 0 and 1 which have been monitored for two-class activity (sleeping and taking medicine). We want to send them a persuasive message in case an anomaly has occurred. In addition, we use "Rec", "Soc" and "Auth" as a keyword for our three example intervention messages. And we assume an intervention message should be sent just in case we encounter a "mild intervention" level. Moreover, we assume a hypothesis for simulating our model such as the user "0" responds to the "authority" persuasive intervention with the probability of "0.8" and user "1" responds to the "social proof" persuasive intervention with the probability of "0.9" irrespective of the activity type. For the implementation of the decision step, we implemented a ϵ-greedy policy. Here we describe the implementation of each of the respective steps.

- `getcontext`: A JSON object describing the current context should normally be retrieved from the environment. The object, in our example scenario, looks as follows:

  ```
  "context" : {"user id" : 1}
  "context" : {"intervention level" : Mild}
  "context" : {"activity" : medication}
  ```

  To simulate the performance of StreamingBandit we generate the context with ran-

dom values for the user id and the activity using the following code:

```
# Let's say we have two subjects: 0 and 1
self.context["subject"] = random.randint(0,1)
# We have two activities: taking medicine and sleeping
self.context["activity"] = random.choice(["medicine", "sleeping"])
# We focus on "mild" interventions only.
self.context["interventon_level"] = "mild"
```

- `getaction`: For the implementation of the decision step, we detail how to use the ϵ-greedy policy. Thus, we randomly select one of the three arms (auth, soc, rec) at each decision with probability ϵ, and we selecting the hitherto best-performing arm with probability 1-ϵ. The following code implements this decision policy in StreamingBandit:

```
# Declaring epsilon:
e = 0.1

# Only if the interventon_level is mild we need to send a notification
if self.context["interventon_level"]=="mild":

# We save a Theta for each subject and for each activity separately
prop_list = base.List(self.get_theta(key=str(self.context["subject"])+
self.context["activity"]), base.Proportion, ["rec", "soc", "auth"])

# With a probability of 0.1
# we pick a random notification message
if np.random.binomial(1,e) == 1:
self.action["intervention"] = prop_list.random()

# Else choose the message with the highest probability of succeeding
else:
self.action["intervention"] = prop_list.max()
```

- `getreward`: As for the context in the `getcontext` step, the rewards should be generated by the actual user response to the message selected in the `getaction` step.

However, to demonstrate our implementation, we implement our scenario—and the respective success probabilities for the two different users—using the following code:

```
# Generate random rewards for the different notification messages
# if intervention level is mild
# Start with subject 0
if self.context["interventon_level"]
== "mild" and self.context["subject"] == 0:
# Generate random rewards for each intervention
# Let's say subject 0 is very susceptible to the persuasion by authority
if self.action["intervention"] == "auth":
self.reward["value"] = np.random.binomial(1,0.8)
else:
self.reward["value"] = np.random.binomial(1,0.3)
# Then the subject number 1
elif self.context["interventon_level"] == "mild" and self.context["subject"] == 1:
# Subject 1 responds to social proof well:
if self.action["intervention"] == "soc":
   self.reward["value"] = np.random.binomial(1,0.9)
else:
self.reward["value"] = np.random.binomial(1,0.1)
```

- `setreward`: Finally, when a reward has been generated, the summary step for the ϵ-greedy policy is implemented by retrieving the stored object θ and subsequently updating it using the observed reward. Using StreamingBandits build in functionalities this can be achieved using the following code:

```
# This is the summary step in which the succes probabilities are-
#stored for each distinct context X action combination
# First get currently stored probability:
prop = base.Proportion(self.get_theta(key=str(self.context["subject"])
+self.context["activity"], value=self.action["intervention"]))

# Compute update based on the observed reward:
prop.update(self.reward["value"])
```

```
# Store the new updated probability:
self.set_theta(prop, key=str(self.context["subject"])
+self.context["activity"], value = self.action["intervention"])
```

As stated in the previous section, the `getcontext` and `getreward` codes are not strictly necessary to use the implemented policy in the field study; these two pieces of code provide us with the opportunity to simulate the usage of StreamingBandit for our envisioned system. To demonstrate the result of our implemented code, we simulated our scenario using $N = 1000$ interactions. This yields the following JSON response:


```
"theta":{
        "0 medicine:soc": {
           "p": "0.36363636363636365",
           "n": "11"
         },
        "0 medicine:rec": {
           "p": "0.25",
           "n": "20"
        },
        0 medicine:auth: {
           "p": "0.8298755186721987",
           "n": "482"
         },
        "0 sleeping:soc": {
           "p": "0.29411764705882354",
           "n": "17"
         },
        "0 sleeping:rec": {
           "p": "0.15384615384615385",
           "n": "13"
         },
        0 sleeping:auth: {
           "p": "0.7886178861788607",
           "n": "492"
```

```
        },
        1 medicine: soc: {
          "p": "0.873661670235546",
          "n": "467"
        },
        "1 medicine:rec": {
          "p": "0.06250000000000001",
          "n": "16"
        },
        "1 sleeping:auth": {
          "p": "0.08333333333333333",
          "n": "24"
        },
        1 sleeping:soc: {
          "p": "0.8974358974358976",
          "n": "429"
        },
}
```

Listing 5.1: Number of selected intervention and their respective mean reward

This object clearly shows the number of times the persuasive interventions (i.e. Auth, Soc, and Rec) messages were selected (n) and their respective mean reward (p), for different user-activity combinations. It is clear that the "Auth" arm is preferred for subject "0", while for subject "1" the arm "soc" has the highest mean reward regardless of activity type and is played most often (see the red rows in the listing 5.1). This result shows that our decision policy indeed predominantly chooses the right message based on our hypothesis, for the right user.

5.7 Chapter Summary

This chapter introduces a system that takes an approach to generate deeply personalized health feedback. The persuasion module automatically recommends an intervention message aim at changing user anomalous behaviors to have a healthier lifestyle. The

system uses a sequential decision-making algorithm, Contextual Multi-armed Bandit, to generate suggestions that minimize the user anomalous behavior. It utilizes concepts of Cialdini's 6 Principles of Persuasion from behavior change theory literature and operationalized them in machine learning optimization functions.

We presented the results from a simulation study that shows promising results. As more and more people use automated technologies to track their health, we believe the persuasion engine's ability to auto-personalize suggestions holds great promise for providing actionable feedback at scale. Note that, our description of the use of StreamingBandit above is rudimentary; the platform itself enables much more advanced modeling of the relationship between the rewards, the context, and the actions, and the platform can be used to easily implement policies in which the action selection probabilities change over time. However, the current simulations show how activity recognition, anomaly detection, and active interventions can be used to create a persuasive system that personalizes interventions.

Chapter 6

Testbed Infrastructure for Experimental Validation

After the aforementioned validation using the simulation studies presented in chapters 4 and 5, we decided to conduct an experimental validation using real-data in our lab testbed. This experiment is employed to assess the performance of the anomaly detection algorithm and to show the functioning of the integrated system (personalized health feedback system).

As a result, this experiment determines to what extent the required functionality is achieved under realistic conditions. In the next sections, we present our smart-lab testbed architecture and review some of the technical aspects that have been used in the building of the test-bed.

6.1 The Test-bed

In order to test our system in a real-time approach with real data, we conducted a lab test-bed, simulating the user environment at home. To this aim, we have deployed sensors and appliances as they may be placed in a real home environment. Regarding the choice of sensors, appliance and an application that receives the issued health feedback, we have chosen those used and developed in a European project (PETAL) carried out in

the HIIS Lab.

The test-bed used to collect the data is the HIIS Lab (shown in Figure 1) with three rooms, a corridor, and a bathroom (see Figure 6.1). We labeled these 3 rooms as the living room, kitchen area, and bedroom.

6.2 The Experimental Procedure

Our experimental procedure is coherent with our methodology explained in this thesis. We first, allocate devices and sensors in our testbed and log the data based on the typical scenario while the user is interacting with sensors. The log data gathered through context manager will be sent to the AR and deviation analysis modules and consequently, if there is any true anomaly that needs an intervention, the persuasion engine issues a personalized intervention and delivers it to the user's application.

6.2.1 Logging Infrastructure

A hybrid (wireless/wired) communication infrastructure is used to read data gathered from the wearable and environmental sensors. Changes in sensors state may trigger personalization interventions related to their value. The sensors that have been considered for this project are standalone and wireless and are as follows:

- Environmental measures:
  - Motion (true/false)
  - Light detection (lux)
  - Temperature (C)
  - Humidity
  - Gas (true/false)
  - Windows/Doors state (Open/Close)

  Obviously, all these metrics will be associated to the different environments which compose the actual context of use. For example, during the laboratory evaluation,

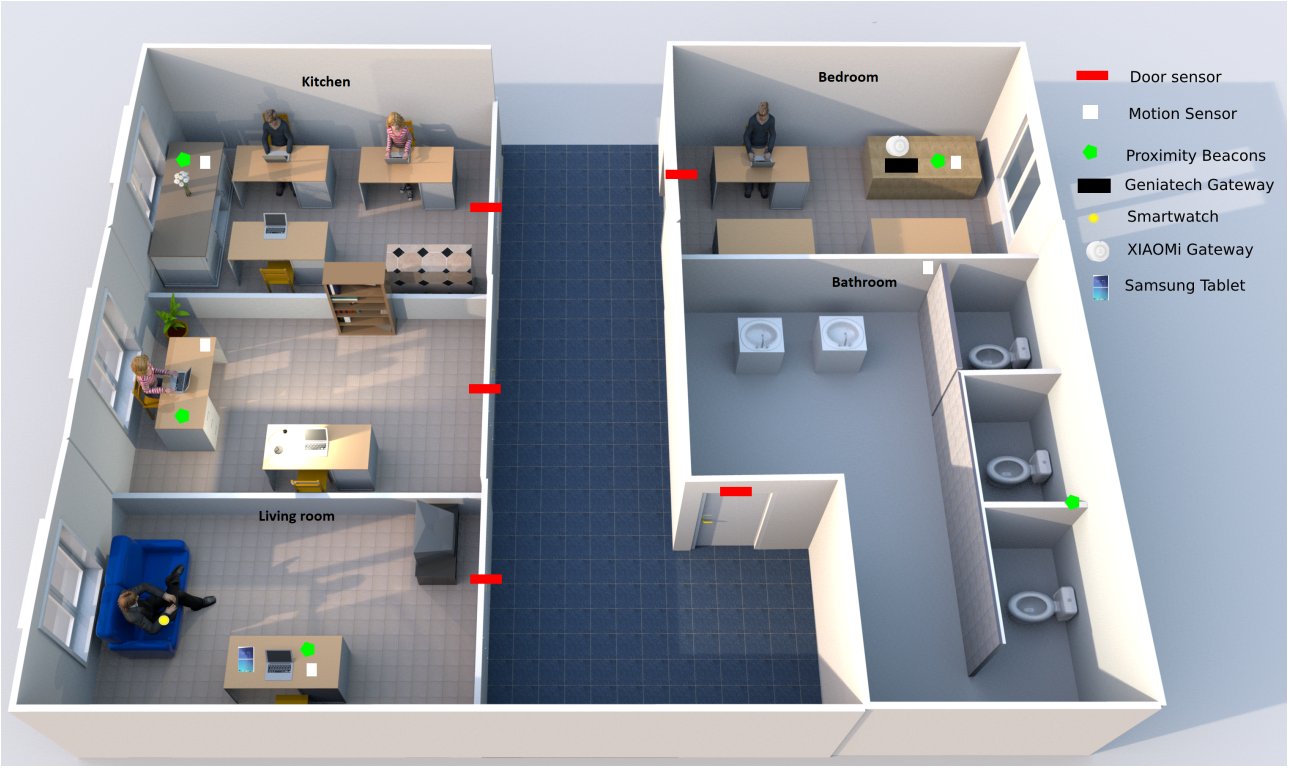


Figure 6.1: Testbed view

it will be possible to define association rules referring to the motion or the temperature of a specific room. Additionally, the following general user measures will be used, which may not be related to specific environment.

- User measures:
  - Position/Room (in which room the user is)
  - Position/Time (how long a user stays in a room)
  - Heart Rate (bpm)
  - Sleep (What time the user went to sleep? how long did the user sleep tonight?)
  - Physical Activity (steps, distance, calories)
  - Going out and re-enter home

Although most of the studies use smartphones to identify the user position, they need the user to have it always with them. This situation, especially at home for older adults, is not realistic. For this reason, the LEMFO LEM7 Android 7 Smart Watch has been used that can be worn most of the time and can support both Bluetooth and Wi-Fi connection at the same time.

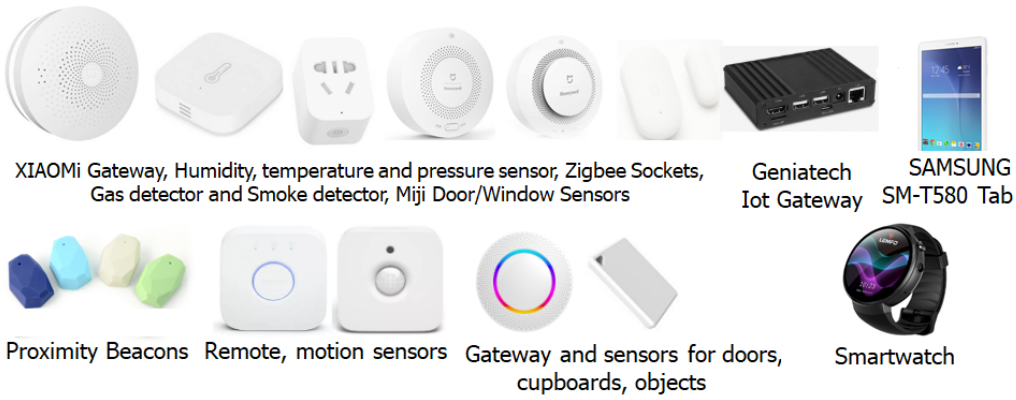


Figure 6.2: Set of sensors and appliances integrated for the trials.

The motion, light level, and temperature can be measured by HUE Motion Sensors [1]. The Hue Motion Sensor has a range of 5 meters and a detection angle of 100 degrees in both horizontal and vertical directions. These sensors are connected to the Hue Bridge through the ZigBee communication protocol, then the Hue Bridge exposes some REST services, which can be called by the Context Delegate in CM.

Also, we considered using Estimote Beacons [2] for taking data about the location and proximity. Estimote Proximity Beacons have a range of 70 meters and Estimote Location Beacons can detect the presence in a range of 200 meters. The beacons send a signal via Bluetooth with an ID that identifies them.

Moreover, the integrated smartwatch, in addition to detecting physiological parameters such as heart rate and step counter and sending them to the Context Server, is also able to connect and communicate at the same time through Bluetooth and Wi-Fi. This is exploited to obtain indoor positioning with the support of Estimote Bluetooth proximity beacons. The Indoor Proximity Delegate exploits the Estimote Proximity SDK and its cloud configuration platform in which each beacon is associated with a room. In this way, when the proximity SDK detects that the smartwatch is inside the region (room) defined by the beacon, it triggers an event in the delegate by providing the room name associated to the beacon. Afterward, the delegate sends this information to the context server.

Figure 6.2 shows various sensors, devices and appliances used in this test-bed.

[1] www2.meethue.com
[2] https://estimote.com

6.2.2 Scenario

We created a typical household scenario consisting of temporally unfolding situations made out of both complex (e.g., prepare a meal) and atomic activities (e.g., open pill dispenser). Naturalistic execution of the scenario is achieved by relying on a hierarchical task tree and leaving the free interpretation to one volunteer that was recruited to perform a set of trials in the test-bed. The scenario describes a sequence of activities (e.g., having breakfast, sleeping, resting, etc.) in terms of high-level actions to be performed by the user, leaving space for a natural execution of the task. For instance, the volunteer is allowed to variate the way the scenario is performed, such as changing, to some extent, the sequence of activities or atomic events involved. The scenario is as follows:

Marco starts his day by taking his medicine before breakfast. He usually has his breakfast between 8:30 to 9:15. He uses to relax in the afternoon around 5 o'clock by watching his favorite show on TV. Normally he goes to bed at 18:30. He usually uses the bathroom 5 times a day.

First, we created a task model using CTT, containing user daily behavior. Then, we associated the user activities with the events gathered by available sensors in the lab. Below there are some rules created in the association tool:

- toileting = typeOfProximity equal INSIDE AND pointOfInterest equal BATHROOM AND motion equal true
- relax = typeOfProximity equal INSIDE and pointOfInterest equal LIVINGROOM AND TV is on
- medication = motion (on pill dispenser) equal true
- breakfast = typeOfProximity equal INSIDE AND pointOfInterest equal KITCHEN AND (Gas Sensor is true or Fridge Door is open)
- sleeping = typeOfProximity equal INSIDE 1 AND pointOfInterest equal BEDROOM 1 AND timeInsideBedroom more 5H AND Motion is false

6.2.3 Data Annotation

During the data collection, a human operator annotates (using a piece of paper and a pen) the activities and locations performed by the user to produce the ground truth comparable with the data in the database. In this way, we can understand the amount of missing data from sensors.

6.2.4 Dataset

We gathered real-world data for the test-bed in our Lab facility by storing data from the context manager while an actor was performing a set of daily-life activities. As the user interacts with the environment, the sensor's information is sent from the Context Delegate to the Context Manager via HTTP(S) through RESTfulservices implemented on the Context Server. When one or more events are verified and the rule conditions are met, the Context Manager notifies the Activity recognition module via HTTP(S) POST to a REST service endpoint (the endpoint is specified during the subscription phase); then the AR stores the list of verified activities into the database and forwards them to the Anomaly detection algorithm. The anomaly detection algorithm identifies if there is any anomaly and updates the activity records in the database with the new information about their anomaly type, time, and date. Then, it transmits these data to the persuasion module. In figure 6.3, you can see one day example of the anomaly detection algorithm output. In figure 6.3, the "$delta$" column is the delta anomaly measure, calculated based on the explanation gave in section 5.4.1.1.1. As we mentioned before, the ($\delta_{anomaly}$) corresponds to the difference between the detected anomalous activity and the planned activity in terms of time, the number of repetitions and the order of the activity. This delta is one of the inputs of the fuzzy rule-based system.

receive_event	enabling_task	anomaly	delta	time	event_date
breakfast	lunch,relaxing,medication	Less-Partial	Delta_LP=1	09:15	2019-07-02
medication	breakfast	Difference-Later-Time	Delta_DLT=129	13:09	2019-07-02
medication	breakfast	More-Order	Delta_MN=1	13:26	2019-07-02
relaxing	sleeping	Less-Partial	Delta_LP=1	17:00	2019-07-02
sleeping	end	Less-Partial	Delta_LP=1	18:30	2019-07-02
sleeping	end	Difference-Later-Time	Delta_DLT=183	20:13	2019-07-02

Figure 6.3: The output of the anomaly detection algorithm for 1 day

Later, the fuzzy rule-based module via a query retrieves all the information needed from the anomaly detection database and calculate the level of intervention needed for each specific anomaly. If, any detected anomaly goes under the "No intervention" category means that the problem is not severe and there is no need for issuing any notification or reminders. But, if the anomaly goes under the "mild intervention" and "high intervention" category, the persuasion engine should analyze the situation and for the former, issue a personalized intervention notification and for the latter should issue an alarm to inform the caretaker or the family members. For this test, we did not consider "high intervention".

In Figure 6.4, the activity "breakfast", first classified as "low intervention" at 9:15, as one minute of delay is not a true anomaly. Each activity error measure gets updated based on the criticality level of that activity and consequently, the new anomaly degree will be calculated. As we mentioned before, activities with a high criticality level get updated every 10 minutes. Therefore, at 10:15 the error measurement $\delta_{anomaly}$ changes in a way that the new calculated anomaly is higher and the "breakfast" activity goes under the "Mild intervention" category.

In this case, the fuzzy system sends the anomalous task, anomaly type and the level of intervention to the REST service endpoint (persuasion engine), via HTTP(S) POST.

category	anomaly	delta_t	anomaly_type	Low Intervention	Mild intervention	High Intervention	TimeStamp	criticality
medication	0.800365	129	Difference-Later-Time	0	0	1	2019-07-02 13:09:57	10
breakfast	0.204167	1	Less-Partial	1	0	0	2019-07-02 09:15:58	9
breakfast	0.790726	60	Less-Partial	0	1	0	2019-07-02 10:15:58	9

Figure 6.4: The output of the fuzzy rule-based system

The Persuasion engine in the persuasion module choose the personalized intervention based on the policy applied in streamingBandit and forwards the intervention to be applied to the application. The user application will receive the intervention via the MQTT protocol. In figure 6.5, you can see the example log of interventions sent to the volunteer.

date	topic	msg	type
2019-07-02 08:30:57	petal*marco.manca	It is time to take your medicine based on your doctor prescription	NOTIFICATION
2019-07-02 13:09:58	petal*marco.manca	Science shows that 70% of healthy older adults take their medicine on time.	NOTIFICATION
2019-07-02 10:15:58	petal*marco.manca	Experts say that skipping breakfast may put you on the fast track to low memory	NOTIFICATION

Figure 6.5: Intervention messages log

For evaluating the second main part of the system, the persuasion module, we need real data from real users for a long period. Because the persuasion module is based on the contextual multi-armed bandit formalization which is able to learn the user preferences based on the user reaction to each issued intervention. It is obvious that achieving such a result in 2 weeks with emulated data is impossible. The limitations in our experiment have been addressed in section 6.4. In this experiment, we test the capacity of the anomaly detection algorithm to find the anomalies in user behavior. The next section shows the discussion and the results.

6.3 Results

We run the test for 2 weeks and during the 2-week monitoring of the user behavior, we gathered 123 activities (based on the user annotation) which 113 of them are stored in the database. This means that our dataset gathered from sensor networks suffer from a significant fraction of missing data (10 records).

This happened due to issues such as communication and sensor interference, power depletion, and hardware failure and will be explained in detail in section 6.4.

We constructed a confusion matrix to calculate the performance of the anomaly detection algorithm. To this aim, we used the measures that already defined in equation Equations (4.12) to (4.14).

The performance of the proposed deviation analysis module and the experiment result are shown respectively in table 6.1 and 6.2. The first line of the table 6.1 shows the behavior of the algorithm when a given anomaly A is generated in a sequence. It reports the number of cases in which anomaly A is correctly classified (true positives), the number of cases in which the anomaly is not detected (false negatives), and the number of cases in which the anomaly A is misclassified as another anomaly A' (false negatives). Similarly, the second line shows the behavior of the algorithm for the sequences that do not contain an anomaly. In this case, it reports the number of cases in which an anomaly is detected (false positives) and the number of cases in which no anomalies are detected (true negatives).

The results show that our system is capable of finding abnormal activities with 87%

Table 6.1: Testbed experimental results

Simulated \ System detection	Abnormal	No Anomaly	Another Anomaly
Abnormal	46	5	4
No Anomaly	6	62	

Table 6.2: Deviation analysis performance summary

Measure	Rate
Sensitivity	88%
Specificity	92%
FPR	7%
Accuracy	87%

accuracy. The difference between the result of the simulation in section 4.2.2.2 and the result of the test-bed is due to the missing data. The anomaly detection algorithm per se has an accuracy of 95%.

Moreover, the "Another Anomaly" considered in the confusion matrix is also due to the misclassification that happens naturally in real-time analysis. For example, in figure 6.3, At the end of the day, we can see that "medication" and "sleeping" activities have been misclassified at 13:09 and 18:30 respectively. But later, the medication task classified as *"More-order"* at 13:26 and "sleeping" is classified at 20.13 as *"Difference-Later-Time"* . It is worth to note that, none of the mentioned anomalies are classified wrongly, rather, they are true at that specific moment in the partial (prefix) sequence that received till then.

6.4 Limitations and Challenge Faced During the Experiments

User behavior is key to the correct detection of the system performance. Given the difficulty of recruiting elders in the remaining time to complete the thesis, we used an actor to make a preliminary tests.

In the first experimental part (simulation) in section 4, the anomaly detection process is

not based on the initial set of sequences from the real world, but we are adding potential failures with synthetic generation. In the second experiment (test-bed in section 6), we emulated an older adult behavior in a living lab, but we can not exactly duplicate it. So, an elderly person that misbehaves may act differently from the way in which the users' misbehavior is simulated or performed.

Summing up, central challenges in the test-bed were:

- Missing data was a prevalent issue. One of the disadvantages of the smartwatch used in our experiment was the battery that was not powerful enough to make it both useful and long-lasting. Due to this problem and the fact that the user location is where the smartwatch is, data may be missing at times due to sensors not being available.

- The false and the missing alarms occurring due to the previous problem and weak technical infrastructure and a long distance between the sensors and the gateway. Because of the missing sensor data, some alarms reached the mobile phones/tablet of the volunteer, and some alarms did not.

- Another limitation was the time and the number of users. The idea behind the persuasion module is that an agent will learn from the environment by interacting with it and receiving rewards for performing actions. In this approach, the system can take time to learn the user's preference. To allow adaptation over time, the user must be monitored for a long time. Hence, in our test-bed we can not validate how this module can learn from the experience to improve its performance over time.

However, we are currently running preliminary experiments in the smart home with volunteer users (which is part of the PETAL European project). The objective of these experiments is to test the capacity of our proposed architecture to use users' feedback and learn their preferences. We encountered problems during the preliminary phase of the experiment such as subjects that dropped out of the study or the users who refused to wear the smartwatch that makes rendering the resulting data incomplete. In addition, Failure in the analysis to appropriately account for subjects dropping out of the study for reasons unknown results in biased conclusions.

Chapter 7

Conclusion and Future Work

The well-being of elderly people depends greatly on proper care and treatment. Their physical status needs to be monitored regularly on a continuous basis. Normally this can be accomplished with the help of a human assistant. In modern days people are rarely available to take care of their elderly relatives. Due to that, the design and development of the remote care systems is an important issue. This is a complicated process as it involves numerous concerns. The design and development of an effective full-featured automated remotecare system is a challenging task. Taking proper care of elderly people differs from generic caregiving systems in various ways because of the diverse nature of problems experienced by people at old age.

7.1 Conclusion

The central objective of our method is to extend the time older people can live in their home environment by increasing their autonomy and assisting them in carrying out activities of daily living. In particular, our goals are to support older adults with useful and usable means to increase their awareness and control of their current lifestyle. This will be achieved through an intelligent platform able to monitor users' behavior, detect the significant changes in their behavior and provide them with relevant and tailored health-related information and quality of life-improving suggestions. Meanwhile, our method is not limited to the specific target group (i.e., older adults and e-health systems). Not only

it can be customized for care programs targeting varying health problems and populations, but also can be used for the other targets and different context of use.

We proposed a general architecture that describes the flow of information from data acquisition to decision making. Our just-in-time adaptive anomaly detection and personalized health intervention framework, is an intervention design aiming to provide the right type/amount of support, at the right time, by adapting to an individual's changing internal and contextual state. Therefore, our framework for JITAI design and personalization is an adaptive and personalized solution to support older adults by performing a daily activity verification and anomaly detection. Moreover, our solution supports implementations for an automated, personalized and persuasive health intervention generation system that issues interventions based on the detected anomalous activity and the user preferences.

The proposed solution first, models the users' daily routine activities using a task model specification and associates these activities with the events in user context. The user model serves as a personalized knowledge-base for detecting the users' abnormal behavior.

Second, it verifies the user activity using the information received from the context manager (which detects relevant contextual events occurring in the older adults' home environment) and associated data in the user model.

Third, it performs an online anomaly detection algorithm to detect any significant changes in user routine. The deviations in user behavior will be classified in 11 categories regarding the activity type, location, order, time and the duration. Later, the system filters the anomalies to reduces false alarms using a Mamdani-type fuzzy rule-based system that as input takes the user context and detected anomalies and outputs the true anomalies with the level of intervention needed.

Finally, by a systematic validation through a system that automatically generates wrong sequences of tasks, we show that our online anomaly detection algorithm is able to find behavioral deviations from the expected behavior at different times by considering the extended classification of the possible deviations with good accuracy. We also presented some preliminary results based on the 2-weeks experiment with real-data in our lab testbed.

In addition, our system supports implementations to issue personalized interventions to users aiming to minimize their anomalous behavior. The personalization part employs a reinforcement learning-based approach to optimize/personalize the intervention delivery concerning the frequency, type, and timing of interventions dynamically according to data aggregated for a person over time. To this aim, we propose the use of a sequential decision policy, implemented based on the contextual multi-armed bandit formalization to select messages adopting distinct persuasive strategies for each individual so that compliance is maximized. We validated the personalized intervention delivery mechanism through a simulation in which deviations, interventions and personas, with differentiating characteristics, are simulated. We present that the personalization algorithm is able to capture the rules associated with the simulated concepts, indicating its potential to be used in real-world settings.

We are currently running more experiments in our living lab to validate our approach and architecture in a real environment and we believe that the overall system—which combines detection and intervention in a closed feedback-loop—can provide a solution for health-care professionals and the elderly themselves. Overall, the developed system can improve the quality of support in context-aware remote healthcare systems and help users to improve significantly the quality of their lives.

7.2 Future Work

There are several possible further developments. First off, while we have proposed the design of the overall system and parts of it have already been tested in practice, evaluating the full system *in situ* poses a future research challenge. In future studies, we aim to validate our platform's mechanism, empirically throughout a filed trial that will be carried out in the scope of Petal project with 10 MCI patients in total. Such an evaluation should allow us to experiment with different sequential allocation policies and properly long-term assess the effectiveness of the proposed system with real data. Our secondary aim is to disseminate the results and tools, so that other studies could easily make use of our material.

Second of all, there are various health-behavior theories and models for developing ef-

fective persuasive intervention messages that will help improve people's health state. While in this thesis, we applied one of these theories (i.e., Cialdini's six principles of persuasion) in future research we aim to apply and evaluate different ones through our proposed sequential decision-making method and determine which aspects of these theories are most effective for older adults. Another improvement of the system which we aim to develop in future is to inject dynamically calculated information at the intervention delivery time. The placeholders in the intervention messages may differ according to the persuasion philosophy behind each intervention.

Third of all, we plan to investigate how to improve the activity recognition module by considering the complex event processing under uncertainty. Usually there exist two types of uncertainty, namely uncertainty in the data coming from sources, and uncertainty in the induction step that derives the main activity (e.g., sleeping) from composite activities. Furthermore, activities may have time-related connections to each other, to form a composite activity. In this respect, we should process the activities to control if they are happening in a meaningful time window. In our system, we assumed the complex events are received in an acceptable time window, and we did not model the uncertainty of events timestamps.

In addition, as our algorithm has the potential to be useful to detect safety implications, another possibility to improve the system is to introduce a reliability parameter for each task associated with events received from the context manager. We plan to compute a confidence probability (CP) as an estimator on the reliability that the input entry belongs to a specific task or not, that for each new event and in real-time.

Finally, ambient intelligence (AmI) and intelligent environments, in general, represent the actual trends in the development of the field of artificial intelligence (AI). Along with the incorporation of these technologies into our everyday life, there are also existing issues related to ethics and transparency, which are still not satisfactorily solved. To be ethical, AI must become explainable and the system must have a transparent design. Especially in remote care systems, the user must be able to fully understand how decisions are being made. Despite the fact that intelligent systems and agents are becoming pervasive in the daily-living of users in ambient assisted living and e-health systems, still the literature reviews on explainable agents are missing. Therefore, we plan to provide an explanation for our recommendation system in the persuasion module because the

lack of explanation for the user may increase the risk of self-deception and may degrade the quality of the interaction [238].

Appendix A

Interventions

We create several short messages that implement different social influence strategies as defined in [225] that will be delivered to individual users. Our focuses on these interventions as we mentioned in section 5.4 is on "medication intake" and "sleeping" activities and between Cialdini's 6 Principles, we chose *Social proof, Reciprocity, and Authority*. Following is the list of intervention messages.

- Interventions for medication intake:
  - **Social Proof**:
    1. - 50% of the people take medication as prescribed.
    2. 55% of elderly people that fail to comply with medication regiments have poorer health.
    3. 26% of elderly people that fail to comply with medication regimens are potentially in serious danger according to their doctors.
    4. Clinical studies report that the average medication adherence ratio is between 43% and 78% among elderly people who are frequently taking medications.
    5. Poor adherence causes approximately 33% to 69% of medication-related hospitalizations and accounts for $100 billion in annual health care costs.
    6. Researchers have found that adults who don't take prescriptions as directed have poorer health in general, including more hospitalizations.

7. take your medicine now. 70% of people find it easier to take their medicine with a daily routine.
8. According to a 2009 study by Odette Gould, a psychology professor, late medication intake may not be as effective — or safe — as they are intended to be.
9. Most of older adults take their medications with a daily routine like brushing the teeth or getting ready for bed to be on time.

– **Reciprocity**:

1. If you take medication as directed for this week, you receive free health advice from our experts.
2. If you take medication as directed, the chance of better health improves dramatically.
3. You'll enjoy better overall health if you take your medicine on time.
4. Take your medicine on time and at the end of the month receive an exclusive pillbox.
5. Use a pill container that we gave you as a gift and take your medicine as instructed.
6. If you take your medicine on time, you'll better manage your ongoing conditions.
7. Keep a "medicine calendar" that we gave you, with your pill bottles and note each time you take a dose.

– **Authority**

1. Try to take your medications at the same time every day as written on your pillbox.
2. It is absolutely essential that you continue to take your medicine on time.
3. Based on expert opinions, cutting off your medication is likely to worsen your symptoms.
4. As your doctor said, your medication works best when taken every day.
5. Take your medication, please. There is a High Cost of not taking your medicines as prescribed.

6. It is time for your medication soon.
7. It is important to take your medication as prescribed.
8. Follow the orders on the pill bottle and take your medicine on time.
9. Do not forget to take your medicine as indicated on your pillbox
10. It is important to take your medicine as prescribed.
11. Time for medication!
12. Taking medication late is certainly much less effective as your doctor said.
13. it is much better for your health to follow your doctor's instructions.
14. Take your medication at the same time every day as your doctor said.
15. Take your pill as the doctor recommended.
16. Follow your doctor's instruction and take your medicine on time.

- Intervention messages for sleeping:
  - **Social Proof**
    1. Recent researches demonstrate that people who sleep at least 7 hours achieve better cognitive performance.
    2. Based on expert opinions, the ideal sleeping time for people of your age is 7 hours. It is healthy to respect that timing.
    3. It has been scientifically proven that sleeping well decreases the risk of dementia.
    4. Researchers say sleeping on time improves your immune system.
    5. Expert says that sleep gives our bodies time to rest and fight viruses and infections.
    6. It has been scientifically proven that sleep increases emotional well-being and you become happier.
    7. Researchers have demonstrated that lack of sleep makes you more irritable.
    8. Studies show that a night of good sleep can repair damaged cells and tissues.
    9. One of the largest studies done on sleeping time shows that sleeping late and short decreases longevity.

10. Studies show that sleep increase creativity which is crucial to healthy aging
11. 60% of studies show that, sleep helps to maintain a healthy weight
12. Researchers found that dieters who were well-rested lost 56% more fat than participants who were sleep-deprived
13. researchers who led the study about sleeping time found that sleep can help reduce the effects of stress.
14. Researchers found that having a regular sleep routine can calm your body.
15. Studies show that having a regular sleep routine can regulate your mood, and even improve decision making.
16. Studies show that 44% of seniors take longer to fall asleep, wake more often during the night.
17. Recent researches demonstrate that the ideal sleeping duration for older adults is 8 hours.

– **Reciprocity**

1. Take a little time for your favorite activity and then get ready for bedtime.
2. If you go to sleep on time, you will have more energy to do your hobbies.
3. Reduce screen time before bed, limiting blue light exposure and have night peace.
4. Liston to this music (link), and night peace!
5. push this button and modify the environment with blue light and comfortable temperature and go to sleep.
6. Follow this simple meditation, breathe deeply 5 times and go to sleep.
7. Follow the instruction in this link to do 5 minutes meditation and sleep tight.
8. Listen to this relaxing music and get ready for going to bed.

– **Authority**

1. Avoid irregular naps during the day as your caregiver said.
2. As the expert prescribes, Limit fluid intake before bed, especially alcohol.
3. As your doctor prescribed, reduce caffeine intake, especially in the afternoon, helps you sleep better.

4. You’ll enjoy better overall health, if you sleep based on your routine scheduled by your caregiver.
5. Do your breath exercise as your yoga instructor said and go to sleep.
6. Expert says that having a regular sleep routine can handle stress and anxiety.
7. Expert says that having a regular sleep routine reduces falls and accidents.
8. As many studies show, using your bed just for sleep and not anything else helps you sleep better.
9. Based on your doctor’s order you should go to bed at the right moment.
10. Stay within the sleeping pattern established with your caregiver.
11. You have to sleep at least 7 hours, as your caregiver told.

Appendix B

Personalized Intervention Simulation using $k-$arm Bernoulli Thompson sampling Algorithm

Thompson sampling is an algorithm for online decision problems where actions are taken sequentially in a manner that must balance between exploiting what is known to maximize immediate performance and investing to accumulate new information that may improve future performance. It works by maintaining a prior on the the mean rewards of its arms. In this, it follows a beta-binomial model with parameters alpha and beta, sampling values for each arm from its prior and picking the arm with the highest value. When an arm (action) is selected and a Bernoulli reward is observed, it modifies the prior based on the reward. This procedure is repeated for the next arm (action) selection. the implemented code in R language is as follow:

```
# Starting values:
true.prob <- data.frame(
   user = rep(c(0,1),each=6),
   activity = rep(c("sleep", "med"), each=3, 2),
   message = rep(c("A", "S", "R"), 4),
   p = c(.1,.1,.7,.1,.1,.6,.2,.6,.2,.9,.1,.1)
)
```

```
sims <- 1000
t <- 200

r.rand <- r.opt <- r.thomp <- matrix(99, nrow=sims, ncol=t)
  for(sim in 1:sims){
  beta <- data.frame(
  user = rep(c(0,1),each=6),
  activity = rep(c("sleep", "med"), each=3, 2),
  message = rep(c("A", "S", "R"), 4),
  a = 1,
  b = 1
  )
    for(i in 1:t){
    user <- rbinom(1,1,.6)
    activity <- sample(c("sleep", "med"),1)

    # random:
    action.rand <- sample(c("A", "S", "R"),1)
    r.rand[sim, i] <- rbinom(1,1,true.prob[     true.prob$user==user &
    true.prob$activity==activity&
    true.prob$message==action.rand,]$p)

    # optimal:
    r.opt[sim, i] <- rbinom(1,1,max(true.prob[true.prob$user==user &
    true.prob$activity==activity,]$p))

    # thompson
    options <- beta[beta$user==user & beta$activity==activity,c(4,5)]
    arm <- which.max(apply(options, 1, function(x){rbeta(1, x[1], x[2])}))
    r.thomp[sim, i] <- rbinom(1,1,true.prob[true.prob$user==user &
    true.prob$activity==activity &
    true.prob$message==c("A", "S", "R")[arm],]$p)
    beta[beta$user==user & beta$activity==activity &
```

```
    beta$message==c("A", "S", "R")[arm],4] <-

    beta[beta$user==user & beta$activity==activity &
    beta$message==c("A", "S", "R")[arm],4] + r.thomp[sim, i]

    beta[beta$user==user & beta$activity==activity &
    beta$message==c("A", "S", "R")[arm],5] <-

    beta[beta$user==user & beta$activity==activity &
    beta$message==c("A", "S", "R")[arm],5] + (1-r.thomp[sim, i])
    }
  }

rand <- colMeans(r.rand)
opt <- colMeans(r.opt)
thomp <- colMeans(r.thomp)
opt.true <- .6*.5*.7+.6*.5*.6+.4*.5*.6+.4*.5*.9
rand.true <- .6*.5*((.1+.1+.7)/3)+.6*.5*((.1+.1+.6)/3)+
             .4*.5*((.2+.6+.2)/3)+.4*.5*((.9+.1+.1)/3)

lo <- loess(thomp~c(1:t), span=0.40)
plot(thomp, type="n", ylim=c(.1,.9),
main="Performance of Thompson samplign", ylab="Expected reward", xlab="Time")
lines(predict(lo), col='red', lwd=2)
abline(h=round(mean(rand),2), lty=2)
abline(h=round(mean(opt),2), lty=2)
```

Bibliography

[1] IBM. Composite events, 2015. URL https://www.ibm.com/support/knowledgecenter/SSGMCP_5.1.0/com.ibm.cics.ts.doc/dfhp9/dfhp9j11.gif.

[2] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.

[3] World Health Organization. *World report on ageing and health*. World Health Organization, 2015.

[4] Johnny D Farah, Natalie Baddour, and Edward D Lemaire. Design, development, and evaluation of a local sensor-based gait phase recognition system using a logistic model decision tree for orthosis-control. *Journal of neuroengineering and rehabilitation*, 16(1):22, 2019.

[5] Anna Zisberg, Nurit Gur-Yaish, and Tamar Shochat. Contribution of routine to sleep quality in community elderly. *Sleep*, 33(4):509–514, 2010.

[6] Janine L Wiles, Annette Leibing, Nancy Guberman, Jeanne Reeve, and Ruth ES Allen. The meaning of 'aging in place' to older people. *The gerontologist*, 52(3): 357–366, 2012.

[7] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM, 2008.

[8] Joyca Lacroix, Privender Saini, and Annelies Goris. Understanding user cognitions to guide the tailoring of persuasive technology-based physical activity in-

terventions. In *Proceedings of the 4th International Conference on Persuasive Technology*, Persuasive '09, pages 9:1–9:8, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-376-1. doi: 10.1145/1541948.1541961. URL `http://doi.acm.org/10.1145/1541948.1541961`.

[9] Jesse Hoey, Craig Boutilier, Pascal Poupart, Patrick Olivier, Andrew Monk, and Alex Mihailidis. People, sensors, decisions: Customizable and adaptive technologies for assistance in healthcare. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(4):20, 2012.

[10] Raul Igual, Inmaculada Plaza, Carlos Medrano, and Maria Angeles Rubio. Personalizable smartphone-based system adapted to assist dependent people. *Journal of Ambient Intelligence and Smart Environments*, 6(6):569–593, 2014.

[11] Ahmad Lotfi, Caroline Langensiepen, Sawsan M Mahmoud, and Mohammad Javad Akhlaghinia. Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of ambient intelligence and humanized computing*, 3(3):205–218, 2012.

[12] Céline Franco, Jacques Demongeot, Christophe Villemazet, and Nicolas Vuillerme. Behavioral telemonitoring of the elderly at home: Detection of nycthemeral rhythms drifts from location data. In *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, pages 759–766. IEEE, 2010.

[13] Marco Manca, Parvaneh Parvin, Fabio Paternò, and Carmen Santoro. Detecting anomalous elderly behaviour in ambient assisted living. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 63–68. ACM, 2017.

[14] Rita Orji and Karyn Moffatt. Persuasive technology for health and wellness: State-of-the-art and emerging trends. *Health informatics journal*, 24(1):66–91, 2018.

[15] Parvaneh Parvin, Stefano Chessa, Marco Manca, and Fabio Paterno. Real-time anomaly detection in elderly behavior with the support of task models. *Proceedings of the ACM on Human-Computer Interaction*, 2(EICS):15, 2018.

[16] Ye Liu, Liqiang Nie, Li Liu, and David S Rosenblum. From action to activity: sensor-based activity recognition. *Neurocomputing*, 181:108–115, 2016.

[17] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. A knowledge-driven approach to composite activity recognition in smart environments. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 322–329. Springer, 2012.

[18] Li Ye, Zhi-Guang Qin, Juan Wang, and Jing Jin. Anomaly event detection in temporal sensor network data of intelligent environments. In *2010 2nd International Conference on Computer Engineering and Technology*, volume 7, pages V7–414. IEEE, 2010.

[19] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462, 2017.

[20] Matthew L Bolton. A task-based taxonomy of erroneous human behavior. *International Journal of Human-Computer Studies*, 108:105–121, 2017.

[21] Philippe Palanque and Célia Martinie. Designing and assessing interactive systems using task models. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 976–979, 2016.

[22] Benjamin T Fine. Unsupervised anomaly detection with minimal sensing. In *Proceedings of the 47th Annual Southeast Regional Conference*, page 60. ACM, 2009.

[23] Parvaneh Parvin, Stefano Chessa, Maurits Kaptein, and Fabio Paternò. Personalized real-time anomaly detection and health feedback for older adults. *Journal of Ambient Intelligence and Smart Environments*, 11(5):453–469, 2019.

[24] Arif Reza Anwary, Hongnian Yu, and Michael Vassallo. Optimal foot location for placing wearable imu sensors and automatic feature extraction for gait analysis. *IEEE Sensors Journal*, 18(6):2555–2567, 2018.

[25] Julius Hannink, Thomas Kautz, Cristian F Pasluosta, Karl-Günter Gaßmann, Jochen Klucken, and Bjoern M Eskofier. Sensor-based gait parameter extraction with deep

convolutional neural networks. *IEEE journal of biomedical and health informatics*, 21(1):85–93, 2016.

[26] Jorge Luis Reyes Ortiz. *Smartphone-based human activity recognition*. Springer, 2015.

[27] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

[28] Mohammed Mehedi Hassan, Md Zia Uddin, Amr Mohamed, and Ahmad Almogren. A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81:307–313, 2018.

[29] Michele Amoretti, Sergio Copelli, Folker Wientapper, Francesco Furfari, Stefano Lenzi, and Stefano Chessa. Sensor data fusion for activity monitoring in the persona ambient assisted living project. *Journal of Ambient Intelligence and Humanized Computing*, 4(1):67–84, 2013.

[30] Chun Zhu and Weihua Sheng. Motion-and location-based online human daily activity recognition. *Pervasive and Mobile Computing*, 7(2):256–269, 2011.

[31] Li Liu, Yuxin Peng, Ming Liu, and Zigang Huang. Sensor-based human activity recognition system with a multilayered model using time series shapelets. *Knowledge-Based Systems*, 90:138–152, 2015.

[32] Eunju Kim, Sumi Helal, and Diane Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing/IEEE Computer Society [and] IEEE Communications Society*, 9(1):48, 2010.

[33] Liming Chen, Chris D Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):961–974, 2011.

[34] Chen Li, William K Cheung, Jiming Liu, and Joseph K Ng. Automatic extraction of behavioral patterns for elderly mobility and daily routine analysis. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(5):54, 2018.

[35] Oya Aran, Dairazalia Sanchez-Cortes, Minh-Tri Do, and Daniel Gatica-Perez. Anomaly detection in elderly daily behavior in ambient sensing environments. In *International Workshop on Human Behavior Understanding*, pages 51–67. Springer, 2016.

[36] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pages 197–205. IEEE, 2014.

[37] Charissa Ann Ronao and Sung-Bae Cho. Recognizing human activities from smartphone sensors using hierarchical continuous hidden markov models. *International Journal of Distributed Sensor Networks*, 13(1):1550147716683687, 2017.

[38] Piyush Gupta and Tim Dallas. Feature selection and activity recognition system using a single triaxial accelerometer. *IEEE Transactions on Biomedical Engineering*, 61(6):1780–1786, 2014.

[39] Billur Barshan and Aras Yurtman. Investigating inter-subject and inter-activity variations in activity recognition using wearable motion sensors. *The Computer Journal*, 59(9):1345–1362, 2016.

[40] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International conference on pervasive computing*, pages 158–175. Springer, 2004.

[41] Stephen S Intille, Kent Larson, Emmanuel Munguia Tapia, Jennifer S Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. Using a live-in laboratory for ubiquitous computing research. In *International Conference on Pervasive Computing*, pages 349–365. Springer, 2006.

[42] Samaneh Zolfaghari, Raziyeh Zall, and Mohammad Reza Keyvanpour. Sonar: Smart ontology activity recognition framework to fulfill semantic web in smart homes. In *2016 Second international conference on web research (ICWR)*, pages 139–144. IEEE, 2016.

[43] Alexander Artikis, Marek Sergot, and Georgios Paliouras. A logic programming approach to activity recognition. In *Proceedings of the 2nd ACM international workshop on Events in multimedia*, pages 3–8. ACM, 2010.

[44] Jing Liao, Yaxin Bi, and Chris Nugent. Using the dempster–shafer theory of evidence with a revised lattice structure for activity recognition. *IEEE Transactions on Information Technology in Biomedicine*, 15(1):74–82, 2010.

[45] Bruno Bouchard, Sylvain Giroux, and Abdenour Bouzouane. A smart home agent for plan recognition of cognitively-impaired patients. *JCP*, 1(5):53–62, 2006.

[46] Gorka Azkune, Aitor Almeida, Diego López-de Ipiña, and Liming Chen. Extending knowledge-driven activity models through data-driven learning techniques. *Expert Systems with Applications*, 42(6):3115–3128, 2015.

[47] Parisa Rashidi, Diane J Cook, Lawrence B Holder, and Maureen Schmitter-Edgecombe. Discovering activities to recognize and track in a smart environment. *IEEE transactions on knowledge and data engineering*, 23(4):527–539, 2010.

[48] Parisa Rashidi and Diane J Cook. Com: A method for mining and monitoring human activity patterns in home-based health monitoring systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):64, 2013.

[49] Diane J Cook, Narayanan C Krishnan, and Parisa Rashidi. Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics*, 43(3): 820–828, 2013.

[50] Diane J Cook, Aaron S Crandall, Brian L Thomas, and Narayanan C Krishnan. Casas: A smart home in a box. *Computer*, 46(7):62–69, 2012.

[51] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. Activity recognition from accelerometer data. In *Aaai*, volume 5, pages 1541–1546, 2005.

[52] Xinxi Wang, David Rosenblum, and Ye Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 99–108. ACM, 2012.

[53] Jesse Hoey, Thomas Plötz, Dan Jackson, Andrew Monk, Cuong Pham, and Patrick Olivier. Rapid specification and automated generation of prompting systems to assist people with dementia. *Pervasive and Mobile Computing*, 7(3):299–318, 2011.

[54] Frank Krüger, Kristina Yordanova, Christoph Burghardt, and Thomas Kirste. Towards creating assistive software by employing human behavior models. *Journal of Ambient Intelligence and Smart Environments*, 4(3):209–226, 2012.

[55] Patrice C Roy, Sylvain Giroux, Bruno Bouchard, Abdenour Bouzouane, Clifton Phua, Andrei Tolstikov, and Jit Biswas. A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer's patients. In *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58. Springer, 2011.

[56] Franz Baader. Basic description logics. In *In [11*. Citeseer, 2002.

[57] Alberto Salguero, Macarena Espinilla, Pablo Delatorre, and Javier Medina. Using ontologies for the online recognition of activities of daily living. *Sensors*, 18(4): 1202, 2018.

[58] Macarena Espinilla, Javier Medina, Alberto Calzada, Jun Liu, Luis Martínez, and Chris Nugent. Optimizing the configuration of an heterogeneous architecture of sensors for activity recognition, using the extended belief rule-based inference methodology. *Microprocessors and Microsystems*, 52:381–390, 2017.

[59] Tongda Zhang, Wensi Fu, Jinchao Ye, and Martin Fischer. Learning movement patterns of the occupant in smart home environments: an unsupervised learning approach. *Journal of Ambient Intelligence and Humanized Computing*, 8(1):133–146, 2017.

[60] Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

[61] J Gregory Trafton, Laura M Hiatt, Anthony M Harrison, Franklin P Tamborello II, Sangeet S Khemlani, and Alan C Schultz. Act-r/e: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, 2(1): 30–55, 2013.

[62] Frank Krüger, Kristina Yordanova, Albert Hein, and Thomas Kirste. Plan synthesis for probabilistic activity recognition. In *ICAART (2)*, pages 283–288, 2013.

[63] Vlad I Morariu and Larry S Davis. Multi-agent event recognition in structured scenarios. In *CVPR 2011*, pages 3289–3296. IEEE, 2011.

[64] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 345–354. ACM, 2013.

[65] Hung Hai Bui, Dinh Q Phung, Svetha Venkatesh, and Hai Phan. The hidden permutation model and location-based activity recognition. In *AAAI*, volume 8, pages 1345–1350, 2008.

[66] Barry Kirwan. Human error identification in human reliability assessment. part 2: detailed comparison of techniques. *Applied ergonomics*, 23(6):371–381, 1992.

[67] M Helander. Towards a practical goms model methodology for user interface design. In *Handbook of humancomputer interaction*, pages 135–158. North-Holland, 1988.

[68] Matthias Giese, Tomasz Mistrzyk, Andreas Pfau, Gerd Szwillus, and Michael Von Detten. Amboss: a task modeling approach for safety-critical systems. In *Engineering Interactive Systems*, pages 98–109. Springer, 2008.

[69] Matthew L Bolton, Radu I Siminiceanu, and Ellen J Bass. A systematic approach to model checking human–automation interaction using task analytic models. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41 (5):961–976, 2011.

[70] Célia Martinie, Philippe Palanque, and Marco Winckler. Structuring and composition mechanisms to address scalability issues in task models. In *IFIP Conference on Human-Computer Interaction*, pages 589–609. Springer, 2011.

[71] Fabio Paternò, Cristiano Mancini, and Silvia Meniconi. Concurtasktrees: A diagrammatic notation for specifying task models. In *Human-Computer Interaction INTERACT'97*, pages 362–369. Springer, 1997.

[72] Célia Martinie, Philippe Palanque, David Navarre, Marco Winckler, and Erwann Poupart. Model-based training: an approach supporting operability of critical interactive systems. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, pages 53–62. ACM, 2011.

[73] José Creissac Campos, Camille Fayollas, Marcelo Gonçalves, Célia Martinie, David Navarre, Philippe Palanque, and Miguel Pinto. A more intelligent test case generation approach through task models manipulation. *Proceedings of the ACM on Human-Computer Interaction*, 1(1):9, 2017.

[74] Masaya Ogawa, Katsuya Honda, Yoshihiro Sato, Shunsuke Kudoh, Takeshi Oishi, and Katsushi Ikeuchi. Motion generation of the humanoid robot for teleoperation by task model. In *2015 24th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pages 71–76. IEEE, 2015.

[75] Laila Paganelli and Fabio Paternò. Tools for remote usability evaluation of web applications through browser logs and task models. *Behavior Research Methods*, 35(3):369–378, 2003.

[76] Matthew L Bolton and Ellen J Bass. Generating erroneous human behavior from strategic knowledge in task models and evaluating its impact on system safety with model checking. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(6):1314–1327, 2013.

[77] Steven Holtzen, Yibiao Zhao, Tao Gao, Joshua B Tenenbaum, and Song-Chun Zhu. Inferring human intent from video by sampling hierarchical plans. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1489–1496. IEEE, 2016.

[78] Martin Giersich, Peter Forbrig, Georg Fuchs, Thomas Kirste, Daniel Reichart, and Heidrun Schumann. Towards an integrated approach for task modeling and human behavior recognition. *Human-Computer Interaction. Interaction Design and Usability*, pages 1109–1118, 2007.

[79] Giulio Mori, Fabio Paternò, and Carmen Santoro. Ctte: support for developing and analyzing task models for interactive system design. *IEEE Transactions on software engineering*, 28(8):797–813, 2002.

[80] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[81] Christoph Burghardt, Maik Wurdel, Sebastian Bader, Gernot Ruscher, and Thomas Kirste. Synthesising generative probabilistic models for high-level activity recognition. In *Activity Recognition in Pervasive Intelligent Environments*, pages 209–236. Springer, 2011.

[82] Weilie Yi and Dana H Ballard. Behavior recognition in human object interactions with a task model. In *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pages 64–64. IEEE, 2006.

[83] Fatiha Latfi, Bernard Lefebvre, and Céline Descheneaux. Ontology-based management of the telehealth smart home, dedicated to elderly in loss of cognitive autonomy. In *OWLED*, volume 258, 2007.

[84] Daniele Riboni, Linda Pareschi, Laura Radaelli, and Claudio Bettini. Is ontology-based activity recognition really effective? In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 427–431. IEEE, 2011.

[85] Daniele Riboni and Claudio Bettini. Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[86] Liming Chen, Chris Nugent, and George Okeyo. An ontology-based hybrid approach to activity modeling for smart homes. *IEEE Transactions on human-machine systems*, 44(1):92–105, 2013.

[87] Alberto G Salguero and Macarena Espinilla. Ontology-based feature generation to improve accuracy of activity recognition in smart environments. *Computers & Electrical Engineering*, 68:1–13, 2018.

[88] Dino Pedreschi, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, and Franco Turini. Meaningful explanations of black box ai decision systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9780–9784, 2019.

[89] Anne Birge James. Activities of daily living and instrumental activities of daily living. 2009.

[90] Aditi Kapoor, KK Biswas, and Madasu Hanmandlu. Unusual human activity detection using markov logic networks. In *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–6. IEEE, 2017.

[91] Zitao Liu and Milos Hauskrecht. Clinical time series prediction: Toward a hierarchical dynamical system framework. *Artificial intelligence in medicine*, 65(1): 5–18, 2015.

[92] Marek Novák, Miroslav Biňas, and František Jakab. Unobtrusive anomaly detection in presence of elderly in a smart-home environment. In *2012 ELEKTRO*, pages 341–344. IEEE, 2012.

[93] UABUA Bakar, Hemant Ghayvat, SF Hasanm, and SC Mukhopadhyay. Activity and anomaly detection in smart home: A survey. In *Next Generation Sensors and Systems*, pages 191–220. Springer, 2016.

[94] Fabien Cardinaux, Simon Brownsell, Mark Hawley, and David Bradley. Modelling of behavioural patterns for abnormality detection in the context of lifestyle reassurance. In *Iberoamerican Congress on Pattern Recognition*, pages 243–251. Springer, 2008.

[95] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Association analysis: basic concepts and algorithms. In *Introduction to Data mining*, volume 321321367. Addison-Wesley Boston, MA, 2005.

[96] Neda Noori, Leila Boti, and Ebrahim Nowzarpoor Shami. Surveying different aspects of anomaly detection and its applications. *JOURNAL OF MATHEMATICS AND COMPUTER SCIENCE-JMCS*, 4(2):129–138, 2012.

[97] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[98] Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pages 59–63, 2012.

[99] Rikard Laxhammar, Goran Falkman, and Egils Sviestins. Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In *2009 12th International Conference on Information Fusion*, pages 756–763. IEEE, 2009.

[100] Wonjoon Kang, Dongkyoo Shin, and Dongil Shin. Detecting and predicting of abnormal behavior using hierarchical markov model in smart home network. In *2010 IEEE 17Th International Conference on Industrial Engineering and Engineering Management*, pages 410–414. IEEE, 2010.

[101] Abdallah Abbey Sebyala, Temitope Olukemi, Lionel Sacks, and Dr Lionel Sacks. Active platform security through intrusion detection using naive bayesian network for anomaly detection. In *London Communications Symposium*, pages 1–5. Citeseer, 2002.

[102] Enea Cippitelli, Samuele Gasparrini, Ennio Gambi, and Susanna Spinsante. A human activity recognition system using skeleton data from rgbd sensors. *Computational intelligence and neuroscience*, 2016:21, 2016.

[103] Markus Wurzenberger, Florian Skopik, Max Landauer, Philipp Greitbauer, Roman Fiedler, and Wolfgang Kastner. Incremental clustering for semi-supervised anomaly detection applied on log data. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, page 31. ACM, 2017.

[104] Hui-Huang Hsu and Chien-Chen Chen. Rfid-based human behavior modeling and anomaly detection for elderly care. *Mobile Information Systems*, 6(4):341–354, 2010.

[105] Robert Gwadera, Mikhail J Atallah, and Wojciech Szpankowski. Reliable detection of episodes in event sequences. *Knowledge and Information Systems*, 7(4):415–437, 2005.

[106] Giuseppe Fenza, Domenico Furno, and Vincenzo Loia. Enhanced healthcare environment by means of proactive context aware service discovery. In *2011 IEEE International Conference on Advanced Information Networking and Applications*, pages 625–632. IEEE, 2011.

[107] Dorothy N Monekosso and Paolo Remagnino. Behavior analysis for assisted living. *IEEE Transactions on Automation science and Engineering*, 7(4):879–886, 2010.

[108] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[109] Rakesh Agrawal, Ramakrishnan Srikant, et al. Mining sequential patterns. In *icde*, volume 95, pages 3–14, 1995.

[110] Louis Atallah and Guang-Zhong Yang. The use of pervasive sensing for behaviour profiling—a survey. *Pervasive and mobile computing*, 5(5):447–464, 2009.

[111] Bingchuan Yuan and John Herbert. Context-aware hybrid reasoning framework for pervasive healthcare. *Personal and ubiquitous computing*, 18(4):865–881, 2014.

[112] Abdur Rahim Mohammad Forkan, Ibrahim Khalil, Zahir Tari, Sebti Foufou, and Abdelaziz Bouras. A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. *Pattern Recognition*, 48(3):628–641, 2015.

[113] Juan Luis Carús Candás, Víctor Peláez, Gloria López, Miguel Ángel Fernández, Eduardo Álvarez, and Gabriel Díaz. An automatic data mining method to detect abnormal human behaviour using physical activity measurements. *Pervasive and Mobile Computing*, 15:228–241, 2014.

[114] Lei Meng, Chunyan Miao, and Cyril Leung. Towards online and personalized daily activity recognition, habit modeling, and anomaly detection for the solitary elderly through unobtrusive sensing. *Multimedia Tools and Applications*, 76(8):10779–10799, 2017.

[115] Vikramaditya Jakkula, Diane J Cook, et al. Anomaly detection using temporal data mining in a smart home environment. *Methods of information in medicine*, 47(1): 70–75, 2008.

[116] James F Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531–579, 1994.

[117] Valerie Guralnik and Karen Zita Haigh. Learning models of human behaviour with sequential patterns. In *Proceedings of the AAAI-02 workshop "Automation as Caregiver*, pages 24–30, 2002.

[118] Jenny Weisenberg, Paul Cuddihy, and Vrinda Rajiv. Augmenting motion sensing to improve detection of periods of unusual inactivity. In *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, page 2. ACM, 2008.

[119] Sawsan M Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. Behavioural pattern identification in a smart home using binary similarity and dissimilarity measures. In *2011 Seventh International Conference on Intelligent Environments*, pages 55–60. IEEE, 2011.

[120] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, 2010.

[121] Pei Sun, Sanjay Chawla, and Bavani Arunasalam. Mining for outliers in sequential databases. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 94–105. SIAM, 2006.

[122] Dan Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge university press, 1997.

[123] Mohamed Al Sharnouby. Anomaly detection using sequences of system calls, March 26 2019. US Patent App. 10/241,847.

[124] Steven A Hofmeyr, Stephanie Forrest, and Anil Somayaji. Intrusion detection using sequences of system calls. *Journal of computer security*, 6(3):151–180, 1998.

[125] Chilukuri K Mohan and Kishan G Mehrotra. Anomaly detection in banking operations. *IDRBT JOURNAL OF*, page 16, 2017.

[126] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[127] Varun Chandola, Varun Mithal, and Vipin Kumar. Comparative evaluation of anomaly detection techniques for sequence data. In *2008 Eighth IEEE international conference on data mining*, pages 743–748. IEEE, 2008.

[128] Alexei J Drummond, Richard D Newcomb, Thomas R Buckley, Dong Xie, Andrew Dopheide, Benjamin CM Potter, Joseph Heled, Howard A Ross, Leah Tooman, Stefanie Grosser, et al. Evaluating a multigene environmental dna approach for biodiversity assessment. *GigaScience*, 4(1):46, 2015.

[129] Xinghao Pan, Jiaqi Tan, Soila Kavulya, Rajeev Gandhi, and Priya Narasimhan. Ganesha: Blackbox diagnosis of mapreduce systems. *ACM SIGMETRICS Performance Evaluation Review*, 37(3):8–13, 2010.

[130] Suratna Budalakoti, Ashok N Srivastava, Ram Akella, and Eugene Turkov. Anomaly detection in large sets of high-dimensional symbol sequences. 2006.

[131] Miao Wang, Cheng Zhang, and Jingjing Yu. Native api based windows anomaly intrusion detection method using svm. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, volume 1, pages 6–pp. IEEE, 2006.

[132] Suratna Budalakoti, Ashok N Srivastava, and Matthew Eric Otey. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(1):101–113, 2008.

[133] Stephanie Forrest, Steven A Hofmeyr, Anil Somayaji, and Thomas A Longstaff. A sense of self for unix processes. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, pages 120–128. IEEE, 1996.

[134] Nisha Gupta, Kritika Anand, and Ashish Sureka. Pariket: mining business process logs for root cause analysis of anomalous incidents. In *International Workshop on Databases in Networked Information Systems*, pages 244–263. Springer, 2015.

[135] João BD Cabrera, Lundy Lewis, and Raman K Mehra. Detection and classification of intrusions and faults using sequences of system calls. *Acm sigmod record*, 30 (4):25–34, 2001.

[136] Wael Khreich, Eric Granger, Robert Sabourin, and Ali Miri. Combining hidden markov models for improved anomaly detection. In *2009 IEEE International Conference on Communications*, pages 1–6. IEEE, 2009.

[137] Ondrej Linda, Todd Vollmer, and Milos Manic. Neural network based intrusion detection system for critical infrastructures. In *2009 international joint conference on neural networks*, pages 1827–1834. IEEE, 2009.

[138] Anjan Kumar Ray, Gang Leng, T Martin McGinnity, Sonya Coleman, and Liam Maguire. Development of cognitive capabilities for smart home using a self-organizing fuzzy neural network. *IFAC Proceedings Volumes*, 45(22):447–454, 2012.

[139] Adithyan Palaniappan, R Bhargavi, and V Vaidehi. Abnormal human activity recognition using svm based approach. In *2012 International Conference on Recent Trends in Information Technology*, pages 97–102. IEEE, 2012.

[140] Xiaolei Li, Jiawei Han, Sangkyum Kim, and Hector Gonzalez. Roam: Rule-and motif-based anomaly detection in massive moving object data sets. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 273–284. SIAM, 2007.

[141] Yulong Fu, Zheng Yan, Jin Cao, Ousmane Koné, and Xuefei Cao. An automata based intrusion detection method for internet of things. *Mobile Information Systems*, 2017, 2017.

[142] Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine learning*, 25(2-3): 117–149, 1996.

[143] Eleazar Eskin, Wenke Lee, and Salvatore J Stolfo. Modeling system calls for intrusion detection with dynamic window sizes. In *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, volume 1, pages 165–175. IEEE, 2001.

[144] Wenke Lee, Salvatore J Stolfo, and Philip K Chan. Learning patterns from unix pro-

cess execution traces for intrusion detection. In *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, pages 50–56. New York;, 1997.

[145] William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995.

[146] German Florez-Larrahondo, Susan M Bridges, and Rayford Vaughn. Efficient modeling of discrete events for anomaly detection using hidden markov models. In *International Conference on Information Security*, pages 506–514. Springer, 2005.

[147] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[148] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM, 2004.

[149] Eamonn Keogh, Jessica Lin, and Ada Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. Ieee, 2005.

[150] Amol Ghoting, Srinivasan Parthasarathy, and Matthew Eric Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining and Knowledge Discovery*, 16(3):349–364, 2008.

[151] Jessica Lin, Eamonn Keogh, Ada Fu, and Helga Van Herle. Approximations to magic: Finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 329–334. Citeseer, 2005.

[152] Ada Wai-Chee Fu, Oscar Tat-Wing Leung, Eamonn Keogh, and Jessica Lin. Finding time series discords based on haar transform. In *International Conference on Advanced Data Mining and Applications*, pages 31–41. Springer, 2006.

[153] Yingyi Bu, Tat-Wing Leung, Ada Wai-Chee Fu, Eamonn Keogh, Jian Pei, and Sam Meshkin. Wat: Finding top-k discords in time series database. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 449–454. SIAM, 2007.

[154] Soumen Chakrabarti, Sunita Sarawagi, and Byron Dom. Mining surprising patterns using temporal description length. In *VLDB*, volume 98, pages 606–617, 1998.

[155] Robert Gwadera, Aristides Gionis, and Heikki Mannila. Optimal segmentation using tree models. *Knowledge and Information Systems*, 15(3):259–283, 2008.

[156] Eamonn Keogh, Stefano Lonardi, and Bill'Yuan-chi' Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 550–556. ACM, 2002.

[157] Mikhail Atallah, Wojciech Szpankowski, and Robert Gwadera. Detection of significant sets of episodes in event sequences. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 3–10. IEEE, 2004.

[158] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15 (2):107–144, 2007.

[159] Linda Lankewicz and Mark Benard. Real-time anomaly detection using a nonparametric pattern recognition approach. In *Computer Security Applications Conference, 1991. Proceedings., Seventh Annual*, pages 80–89. IEEE, 1991.

[160] Gladys Block, Barbara Sternfeld, Clifford H Block, Torin J Block, Jean Norris, Donald Hopkins, Charles P Quesenberry Jr, Gail Husson, and Heather Anne Clancy. Development of alive!(a lifestyle intervention via email), and its effect on health-related quality of life, presenteeism, and other behavioral outcomes: randomized controlled trial. *Journal of medical Internet research*, 10(4), 2008.

[161] Wijnand IJsselsteijn, Yvonne De Kort, Cees Midden, Berry Eggen, and Elise Van Den Hoven. Persuasive technology for human well-being: setting the scene. In *International conference on persuasive technology*, pages 1–5. Springer, 2006.

[162] Jesse Dallery, Allison Kurti, and Philip Erb. A new frontier: Integrating behavioral and digital technology to promote health behavior. *The Behavior Analyst*, 38(1): 19–49, 2015.

[163] Ian Li, Anind Dey, and Jodi Forlizzi. A stage-based model of personal informatics systems. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 557–566. ACM, 2010.

[164] Aniek J Lentferink, Hilbrand KE Oldenhuis, Martijn de Groot, Louis Polstra, Hugo Velthuijsen, and Julia EWC van Gemert-Pijnen. Key components in ehealth interventions combining self-tracking and persuasive ecoaching to promote a healthier lifestyle: a scoping review. *Journal of medical Internet research*, 19(8):e277, 2017.

[165] Harri Oinas-Kukkonen and Marja Harjumaa. A systematic framework for designing and evaluating persuasive systems. In *International conference on persuasive technology*, pages 164–176. Springer, 2008.

[166] Jennifer Dickman Portz, Amy Miller, Brittany Foster, and Lindsey Laudeman. Persuasive features in health information technology interventions for older adults with chronic diseases: a systematic review. *Health and Technology*, 6(2):89–99, 2016.

[167] Brian J Fogg. Persuasive technology: using computers to change what we think and do. *Ubiquity*, 2002(December):5, 2002.

[168] Kirsi Halttu and Harri Oinas-Kukkonen. Persuading to reflect: role of reflection and insight in persuasive systems design for physical health. *Human–Computer Interaction*, 32(5-6):381–412, 2017.

[169] Susan Michie, Maartje M van Stralen, and Robert West. The behaviour change wheel: a new method for characterising and designing behaviour change interventions. *Implementation science*, 6(1):42, 2011.

[170] Susan Michie and Robert West. Behaviour change theory and evidence: a presentation to government. *Health Psychology Review*, 7(1):1–22, 2013.

[171] Susan Michie, Natasha Hyder, Asha Walia, and Robert West. Development of a taxonomy of behaviour change techniques used in individual behavioural support for smoking cessation. *Addictive behaviors*, 36(4):315–319, 2011.

[172] Susan Michie, Charles Abraham, Craig Whittington, John McAteer, and Sunjai Gupta. Effective techniques in healthy eating and physical activity interventions: a meta-regression. *Health Psychology*, 28(6):690, 2009.

[173] Charles Abraham and Susan Michie. A taxonomy of behavior change techniques used in interventions. *Health psychology*, 27(3):379, 2008.

[174] MS Hagger, David Keatley, Derwin Chan, et al. Calo-re taxonomy of behavior change techniques. 2014.

[175] Susan Michie, Stefanie Ashford, Falko F Sniehotta, Stephan U Dombrowski, Alex Bishop, and David P French. A refined taxonomy of behaviour change techniques to help people change their physical activity and healthy eating behaviours: the calo-re taxonomy. *Psychology & health*, 26(11):1479–1498, 2011.

[176] Brian J Fogg. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*, page 40. ACM, 2009.

[177] Harri Oinas-Kukkonen and Marja Harjumaa. Persuasive systems design: Key issues, process model, and system features. *Communications of the Association for Information Systems*, 24(1):28, 2009.

[178] Robert B Cialdini. Harnessing the science of persuasion. *Harvard business review*, 79(9):72–81, 2001.

[179] Robert B Cialdini. The science of persuasion. *Scientific American Mind*, 14(1): 70–77, 2004.

[180] W Randy Clark and Katie J Kemp. Using the six principles of influence to increase student involvement in professional organizations: A relationship marketing approach. *Journal for Advancement of Marketing Education*, 12, 2008.

[181] Shelly Chaiken. Heuristic versus systematic information processing and the use of source versus message cues in persuasion. *Journal of personality and social psychology*, 39(5):752, 1980.

[182] Berardina De Carolis and Irene Mazzotta. A user-adaptive persuasive system based on 'a-rational'theory. *International Journal of Human-Computer Studies*, 108:70–88, 2017.

[183] Maurits Kaptein, Panos Markopoulos, Boris de Ruyter, and Emile Aarts. Can you be persuaded? individual differences in susceptibility to persuasion. In *IFIP Conference on Human-Computer Interaction*, pages 115–118. Springer, 2009.

[184] Shlomo Berkovsky, Jill Freyne, and Harri Oinas-Kukkonen. Influencing individually: fusing personalization and persuasion. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(2):9, 2012.

[185] Jesse Hoey, Pascal Poupart, Axel von Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

[186] Alex Mihailidis, Jennifer N Boger, Tammy Craig, and Jesse Hoey. The coach prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC geriatrics*, 8(1):28, 2008.

[187] Andrew S Lan and Richard G Baraniuk. A contextual bandits framework for personalized learning action selection. In *EDM*, pages 424–429, 2016.

[188] Yuzhong Lin, Joran Jessurun, Bauke de Vries, and Harry Timmermans. Motivate: context aware mobile application for activity recommendation. In *International Joint Conference on Ambient Intelligence*, pages 210–214. Springer, 2011.

[189] Inbal Nahum-Shani, Shawna N Smith, Ambuj Tewari, Katie Witkiewitz, Linda M Collins, Bonnie Spring, and S Murphy. Just in time adaptive interventions (jitais): An organizing framework for ongoing health behavior support. *Methodology Center technical report*, 2014:14–126, 2014.

[190] Sunny Consolvo, Predrag Klasnja, David W McDonald, and James A Landay. Goal-setting considerations for persuasive technologies that encourage physical activity. In *Proceedings of the 4th international Conference on Persuasive Technology*, page 8. ACM, 2009.

[191] Abby C King, Eric B Hekler, Lauren A Grieco, Sandra J Winter, Jylana L Sheats, Matthew P Buman, Banny Banerjee, Thomas N Robinson, and Jesse Cirimele. Harnessing different motivational frames via mobile phones to promote daily physical

activity and reduce sedentary behavior in aging adults. *PloS one*, 8(4):e62613, 2013.

[192] Stephanie Bauer, Judith de Niet, Reinier Timman, and Hans Kordy. Enhancement of care through self-monitoring and tailored feedback via text messaging and their use in the treatment of childhood overweight. *Patient education and counseling*, 79(3):315–319, 2010.

[193] Andrew Quanbeck, Ming-Yuan Chih, Andrew Isham, Roberta Johnson, and David Gustafson. Mobile delivery of treatment for alcohol use disorders: a review of the literature. *Alcohol research: current reviews*, 36(1):111, 2014.

[194] Katie Witkiewitz and G Alan Marlatt. Relapse prevention for alcohol and drug problems: that was zen, this is tao. *American psychologist*, 59(4):224, 2004.

[195] Dror Ben-Zeev, Susan M Kaiser, Christopher J Brenner, Mark Begale, Jennifer Duffecy, and David C Mohr. Development and usability testing of focus: A smartphone system for self-management of schizophrenia. *Psychiatric rehabilitation journal*, 36(4):289, 2013.

[196] William Riley, Jami Obermayer, and Jersino Jean-Mary. Internet and mobile phone text messaging intervention for college smokers. *Journal of American College Health*, 57(2):245–248, 2008.

[197] Kevin Patrick, Fred Raab, Marc Adams, Lindsay Dillon, Marion Zabinski, Cheryl Rock, William Griswold, and Gregory Norman. A text message-based intervention for weight loss: randomized controlled trial. *Journal of medical Internet research*, 11(1):e1, 2009.

[198] Eric Granholm, Dror Ben-Zeev, Peter C Link, Kristen R Bradshaw, and Jason L Holden. Mobile assessment and treatment for schizophrenia (mats): a pilot trial of an interactive text-messaging intervention for medication adherence, socialization, and auditory hallucinations. *Schizophrenia bulletin*, 38(3):414–425, 2011.

[199] Ólöf Birna Kristjánsdóttir, Egil A Fors, Erlend Eide, Arnstein Finset, Tonje Lauritzen Stensrud, Sandra van Dulmen, Sigrid Hørven Wigers, and Hilde Eide. A smartphone-based intervention with diaries and therapist-feedback to reduce

catastrophizing and increase functioning in women with chronic widespread pain: randomized controlled trial. *Journal of medical Internet research*, 15(1):e5, 2013.

[200] Amaury Trujillo and Maria Claudia Buzzi. Towards a fuzzy rule-based systems approach for adaptive interventions in menopause self-care. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 53–56. ACM, 2018.

[201] Jaroslaw Jankowski. Increasing website conversions using content repetitions with different levels of persuasion. In *Asian Conference on Intelligent Information and Database Systems*, pages 439–448. Springer, 2013.

[202] Dimitris K Kardaras, Bill Karakostas, and Xenia J Mamakou. Content presentation personalisation and media adaptation in tourism web sites using fuzzy delphi method and fuzzy cognitive maps. *Expert Systems with Applications*, 40(6):2331–2342, 2013.

[203] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462, 2018.

[204] Mashfiqui Rabbi, Min Hane Aung, Mi Zhang, and Tanzeem Choudhury. Mybehavior: automatic personalized health feedback from user behaviors and preferences using smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 707–718. ACM, 2015.

[205] Ambuj Tewari and Susan A Murphy. From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*, pages 495–517. Springer, 2017.

[206] Pablo Paredes, Ran Gilad-Bachrach, Mary Czerwinski, Asta Roseway, Kael Rowan, and Javier Hernandez. Poptherapy: coping with stress through pop-culture. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pages 109–117. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[207] Steven L Scott. Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry*, 31(1):37–45, 2015.

[208] Xiaofang Zhang, Qian Zhou, Tieke He, and Bin Liang. Con-cname: A contextual multi-armed bandit algorithm for personalized recommendations. In *International Conference on Artificial Neural Networks*, pages 326–336. Springer, 2018.

[209] Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, pages 485–492, 2010.

[210] Linda M Collins, Susan A Murphy, and Karen L Bierman. A conceptual framework for adaptive preventive interventions. *Prevention science*, 5(3):185–196, 2004.

[211] Abby C King, David K Ahn, Brian M Oliveira, Audie A Atienza, Cynthia M Castro, and Christopher D Gardner. Promoting physical activity through hand-held computer technology. *American journal of preventive medicine*, 34(2):138–142, 2008.

[212] Brian Suffoletto, Clifton Callaway, Jeff Kristan, Kevin Kraemer, and Duncan B Clark. Text-message-based drinking assessments and brief interventions for young adults discharged from the emergency department. *Alcoholism: Clinical and Experimental Research*, 36(3):552–560, 2012.

[213] Dror Ben-Zeev, Christopher J Brenner, Mark Begale, Jennifer Duffecy, David C Mohr, and Kim T Mueser. Feasibility, acceptability, and preliminary efficacy of a smartphone intervention for schizophrenia. *Schizophrenia bulletin*, 40(6):1244–1253, 2014.

[214] Eunju Kim, Sumi Helal, Chris Nugent, and Mark Beattie. Analyzing activity recognition uncertainties in smart home environments. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(4):52, 2015.

[215] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. Personalization of context-dependent applications through trigger-action rules. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(2):14, 2017.

[216] Khamla Non-Alisavath, Somphone Kanthavong, Khanthanou Luangxaysana, and Xaythavy Louangvilay. Context-awareness application to control multiple sensors for monitoring smart environment. In *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 920–923. IEEE, 2017.

[217] MC Kaptein and JMA Kruijswijk. Streamingbandit: A platform for developing adaptive persuasive systems. 2016.

[218] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

[219] James Reason. *Human error*. Cambridge university press, 1990.

[220] Erik Hollnagel. The phenotype of erroneous actions. *International Journal of Man-Machine Studies*, 39(1):1–32, 1993.

[221] J Morreale, BE Gronbeck, and HW Simons. Persuasion in society, 2001.

[222] Lotfi A Zadeh et al. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[223] Robert B Cialdini. The science of persuasion. *Scientific American*, 284(2):76–81, 2001.

[224] Frank J Schwebel and Mary E Larimer. Using text message reminders in health care services: A narrative literature review. *Internet interventions*, 13:82–104, 2018.

[225] Robert B Cialdini. *Influence*, volume 3. A. Michel Port Harcourt, 1987.

[226] Maurits Kaptein, Boris De Ruyter, Panos Markopoulos, and Emile Aarts. Adaptive persuasive systems: a study of tailored persuasive text messages to reduce snacking. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(2):10, 2012.

[227] Susan T Fiske and Shelley E Taylor. Social cognition, 1991. *Social Cognition (2nd Ed.). Xviii, 717 Pp. New York, NY, England: Mcgraw-Hill Book Company*, 1991.

[228] Chonghua Wang. A study of membership functions on mamdani-type fuzzy inference system for industrial decision-making. 2015.

[229] Nishchal K Verma and AK Ghosh. *Computational Intelligence: Theories, Applications and Future Directions-Volume I: ICCI-2017*, volume 798. Springer, 2018.

[230] MAS Masoum, EF Fuchs, MAS Masoum, and EF Fuchs. Optimal placement and sizing of shunt capacitor banks in the presence of harmonics. *Power Quality in Power Systems and Electrical Machines*, pages 887–959, 2015.

[231] Vagelis Plevris, Georgia Kremmyda, and Yasin Fahjan. *Performance-Based Seismic Design of Concrete Structures and Infrastructures*. IGI Global, 2017.

[232] Indu Manickam, Andrew S Lan, and Richard G Baraniuk. Contextual multi-armed bandit algorithms for personalized learning action selection. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 6344–6348. IEEE, 2017.

[233] Jules Kruijswijk, Robin van Emden, Petri Parvinen, and Maurits Kaptein. Streamingbandit; experimenting with bandit policies. *arXiv preprint arXiv:1602.06700*, 2016.

[234] Yingce Xia, Tao Qin, Weidong Ma, Nenghai Yu, and Tie-Yan Liu. Budgeted multi-armed bandits with multiple plays. In *IJCAI*, pages 2210–2216, 2016.

[235] Mark Collier and Hector Urdiales Llorens. Deep contextual multi-armed bandits. *arXiv preprint arXiv:1807.09809*, 2018.

[236] Maurits Kaptein. The use of thompson sampling to increase estimation precision. *Behavior research methods*, 47(2):409–423, 2015.

[237] Dean Eckles and Maurits Kaptein. Thompson sampling with the online bootstrap. *arXiv preprint arXiv:1410.4009*, 2014.

[238] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. Explainable agents and robots: Results from a systematic literature review. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems, 2019.