Personalized real-time anomaly detection and health feedback for older adults

Article *in* Journal of Ambient Intelligence and Smart Environments · September 2019

DOI: 10.3233/AIS-190536

4 authors:

Parvaneh Parvin
Institute of Information Science and Technologies "Alessandro Faedo" - ISTI

Stefano Chessa
Università di Pisa

Maurits Kaptein
Tilburg University

Fabio Paternò
Italian National Research Council

Journal of Ambient Intelligence and Smart Environments 0 (0) 1
IOS Press

Personalized Real-time Anomaly Detection and Health Feedback for Older Adults

Parvaneh Parvin [a,*], Stefano Chessa [b], Maurits Kaptein [c] and Fabio Paternò [d]
[a] *Department of Computer Science, University of Pisa,Largo bruno pontecorvo 3, 56127 Pisa, Italy*
E-mails: parvaneh.parvin@di.unipi.it, stefano.chessa@unipi.it
[b] *WN Laboratory, ISTI-CNR,Via Giuseppe Moruzzi, 1, 56124 Pisa, Italy*
[c] *Jheronimus Academy of Data Science, Sint Janssingel 92, 5211 DA, 's-Hertogenbosch, The Netherlands*
E-mail: M.C.Kaptein@uvt.nl
[d] *HIIS Laboratory, ISTI-CNR,Via Giuseppe Moruzzi, 1, 56124 Pisa, Italy*
E-mail: fabio.paterno@isti.cnr.it

Abstract. Rapid population aging and the availability of sensors and intelligent objects motivate the development of healthcare systems; these systems, in turn, meet the needs of older adults by supporting them to accomplish their day-to-day activities. Collecting information regarding older adults daily activity potentially helps to detect abnormal behavior. Anomaly detection can subsequently be combined with real-time, continuous and personalized interventions to help older adults actively enjoy a healthy lifestyle. This paper introduces a system that uses a novel approach to generate personalized health feedback. The proposed system models user's daily behavior in order to detect anomalous behaviors and strategically generates interventions to encourage behaviors conducive to a healthier lifestyle. 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's architecture in detail and we provide example implementations for the anomaly detection and corresponding health feedback.

Keywords: Ambient assisted living, remote monitoring, elderly behavior analysis, anomaly detection, health interventions

1. Introduction

Nowadays, for the first time in history, the majority of people can expect to live into their 60s and beyond [1]. In 2008, the World Health Organization (WHO) declared that primary healthcare for the elderly is needed now more than ever before [2]. With more people living longer there will be larger numbers of vulnerable people who may experience different physical and mental impairments and may also need support to accomplish day-to-day activities. Most older adults benefit from structure in their day-to-day life and even begin to feel insecure when such structure is lacking. Hence, a strict daily routine provides a sense of security against unknowns. A strict routine also helps caregivers as they can use the routine to plan the elderly's activity. A daily routine simply sets in place the same activities at generally the same time on each day. It ensures that important activities get done without fails, such as medication management, regular nutritious meals, and daily hygiene. A proper daily routine can bring peace and predictability to the older adult's life. It can reduces stress and anxiety because they know exactly what will be happening, how the activity will be done, and when it will occur. A predictable routine can also help them to have a deeper sleep [3] and to feel more confident.

In the field of remote healthcare, current technologies offer plenty of smart objects and sensors that enable long-term monitoring of an elderly's in-home activities and detecting deviation from his/her daily routine i.e., anomalies. A true challenge to such context-

*Corresponding author. E-mail: parvaneh.parvin@di.unipi.it.

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 health interventions to help prevent serious health issues and maintain their routine behavior. Indeed, there is a strong need to encourage the elderly to change their anomalous behavior to a healthy lifestyle [6].

The capabilities of elderly users who live in a smart environment and their needs are the most important requirements to consider to build such a system. Researchers have focused on identifying the real needs of the elderly who have the ability to accomplish certain daily activities. In such research, the elderly involved need some kind of support from a caregiver or a relative, but they can live independently most of the time [4]. Previous work further identified that the most highly valued systems, according to the elderly themselves and the caregivers, are remote care systems and reminders [5]. Moreover, using remote care systems helps experts and caretakers to keep track of the overall h ealth condition of older adults and provide them with real-time feedback about anomalous behavior and remote support.

Some systems can detect the anomalies, but they suffer from high false alert rates because they do not consider the effect of other related contexts (e.g., user activity, the criticality level of that activity for each individual) and context history. Hence, the need to detect and intervene timely is not well satisfied within existing models of health care. There is a pressing need to develop comprehensive healthcare related approaches to provide support to family, caregivers and the elderly themselves.

For the older adults that are motivated to maintain their day-to-day routine activities but encounter difficulties when trying to do so (e.g., because of mild cognitive impairments), external interventions can be very effective [7]. In addition, older people generally have unique attitudes toward healthy behaviors and this means that some designed interventions may fail to satisfy their needs [8]. So, there is a need for personalized suggestions. In the majority of healthcare systems, such personalization is provided only through human health coaches. In this paper, we address this need by building an automated, personalized, health intervention generation system. To this end, we propose a method that models user daily activity, recognizes the user's behavior from the collected information, detects anomalous behavior and issues personalized interventions based on the detected anomalous activity.

1.1. Contribution

This paper presents a new integrated system which builds and improves on earlier work and supports more extensive actionable suggestions that take into account both users' anomalous behaviors and the user context. With respect to the preliminary version [9], the additional contributions include:

- Applying a fuzzy rule-based system to identify the intervention level based on the detected deviations in user behavior.
- Developing an integrated system that utilizes the user context (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.

1.2. Organization of the paper

The remainder of this paper is organized as follows. Section 2 summarizes the related work; first, we discuss prior work on anomaly detection and subsequently discuss possible behavioral interventions. In sections 3 and 4 we describe the proposed system in detail. Section 5 is dedicated to the presentation of the simulations performed and the results obtained. Lastly, in section 6 we further discuss the advantages of our system, critically reflect on the approach, and provide suggestions for future work.

2. Related Work

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' behavior.

2.1. Modeling Human Routine Behavior

Routines are sequences of activities regularly followed. In the area of remote monitoring, there are some works focused on the basic activities [10, 11] but these systems detect a limited number of activities. Machine learning techniques are broadly used to detect daily behavioral routines and have deployed supervised [12] and unsupervised [13, 14] learning techniques. Supervised machine learning approaches often require lengthy and expensive data labeling by domain experts. Instead, unsupervised learning methods cluster behaviors without prior knowledge of labels, but generally, there is no guarantee that the resulting clusters represent routines, and they also require long-term context history to learn the behavioral patterns.

In Kruger et al., they used symbolic models to describe user behavior in terms of preconditions and effects [15]. 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 [16].

Banovic et al. [17] present an approach to manually exploring what behaviors characterize routines. They present an algorithm based on an existing Inverse Reinforcement Learning (maximum casual entropy) which enables the system to automatically reason about routines. In this work, they weakly label instances using people's demonstrated routine and classify new activities based on the probability that they belong to the routine model. In previous work [9], we modeled user routine activities according to the ConcurTaskTrees (CTT) language [18] which is defined in terms of a hierarchical composition of tasks connected by various operators. These operators describe the temporal relationships among tasks. Subsequently, we developed an inference technique to map the events in the user context with the task model.

2.2. Abnormal Behavior Detection

Besides behavior modeling, detecting behavior changes is another crucial and challenging task. 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. Fine [19] 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 [9].

Monekosso and Remagnino [20] proposed a model-based behavior analysis system. They use Hidden Markov Models (HMM) to model user behavior from sensed data. They identified activities (i.e., cooking, eating, etc.) corresponding to the pattern of events gathered by sensors. The system is able to recognize the anomalies (e.g., repeated patterns) against a normal behavior. But unlike our work, the detected anomalies need further examination by a domain expert to precisely indicate the nature of the anomaly. Candas et al., [21] propose and validate an automatic data mining method based on physical activity measurements. Abnormal human behavior is detected as an increase or decrease of the physical activity according to the historical data. Historical data is used to model human behavior without assuming theoretical models, but rather an information about the last physical activity levels of the user. 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. Finally, [22] 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.

In this work, we use a similar method [21, 22] using the fuzzy rule-based system to decide about the degree of the detected anomaly and consequently the level of intervention needed. While in [21] 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 the anomalies is different from the one in [22].

2.3. Personalized Persuasive Interventions

Persuasive technologies play an important role in improving and effectively employing large scale, personalized interventions [23] to change behaviors. Health behavior interventions have, for example, been used to give step-by-step instruction to users for performing their daily activities [8]. In this work, the authors 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 [24].

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. For instance, MyBehavior [25] is a personalized healthy lifestyle recommendation system to help users toward healthier lifestyle regarding physical activity and dietary behavior. Here the authors used a decision-making algorithm based on a Multi-Armed Bandit formulation to generate context-dependent personalized interventions. Although their approach results in maximizing the calorie loss in individuals, it still required manual entry of food photos.

The MAB formalization has been used for personalized recommendation in other domains too. For example, they have been used for stress reduction [26], learning action selection for the student [24], modern service economy[27], suggesting personalized news articles on Yahoo [28] and serving advertisement in Google [29]. Inspired by these earlier results we also adopt a (contextual) MAB formalization to personalize persuasive interventions after detecting behavioral anomalies.

3. A Motivating Scenario

In this section, we describe a real-life daily routine activity of an older adult as a motivation scenario for our system. Sara, 76 years-old, is alone. She likes to wake up around 7 a.m. and after toileting, she takes red medicine for her Cardiovascular disease. 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 watching television 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 about 10 p.m. On average, she uses the bathroom 15 times a day, and she rests anytime during her daily activities. In our model, a situation is said to be anomalous when any behavioral changes (e.g., going to bed late, forgetting to take 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.

4. Proposed Architecture

We previously published early ideas of health feedback automation along with a system that detects the anomalies in the daily activity of older adults [9]. The previous version used a profiling strategy to model the user behavior detected through a Context Manager (CM), a middleware software which detects the events corresponding to the user behavior. Later, by comparing the logged data with the user daily activity model and considering the task-related attributes (e.g., task time, task criticality level, etc.) the system detects the deviations in the user behavior and provides detailed information about the anomaly type and the time of occurrence.

In this paper, we briefly describe our daily activity modeling along with the online activity recognition module and the mechanism of detecting the complex events. Further, we explain the method of decision-making to identifying the true abnormality with their degree of the anomaly using a fuzzy rule-based system. Next, an automated multi-armed bandit persuasive suggestion generation module will be presented that utilizes the anomalous behavior data and the user context to suggest changes aim to maximize the chances of losing bad habits of older people. We use a novel sequential decision platform called StreamingBandit to personalize interventions based on detected anomalies. StreamingBandit allows for the easy implementation of sequential decision policies and provides us with the opportunity to experiment with different policies.

Our system is designed to close the loop between the production of user's log data and personalized health suggestions. We preliminary present the overall architecture of our system, and then we briefly discuss the different parts of the system which aim at adapting the personalized health suggestion to the user behavior.

Figure 1 shows the architecture of the system that we have designed to achieve our goals. Its structure can be described according to three main phases i) modeling the user behavior and configuring the analysis ii) online activity recognition to detect the deviations; iii) personalizing persuasive interventions.

The first phase consists of building the user daily routine model and mapping each basic activity to the events in the user context for further analysis. The second phase performs the online activity recognition and finds the anomalies by passing the recognized activities to the online anomaly detection algorithm. Later, 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. The third phase implements a personalized intervention suggestion engine using contextual multi-armed bandit formalization which according to the detected intervention level and the user activity generates personalized messages to help the user increase their life quality.

4.1. Daily Activity Modeling

The first phase has been explained in detail in our preliminary paper [9]. In summary, we created the task model which represents the elderly routine daily activities with the previously existing ConcurTaskTrees (CTTE) tool [18]. For simplicity, seven activities of daily living (ADLs) have been considered. These activities are sleeping, waking up, eating, toileting/showering, walking, cooking, and resting. Each activity has been 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 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 home. 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

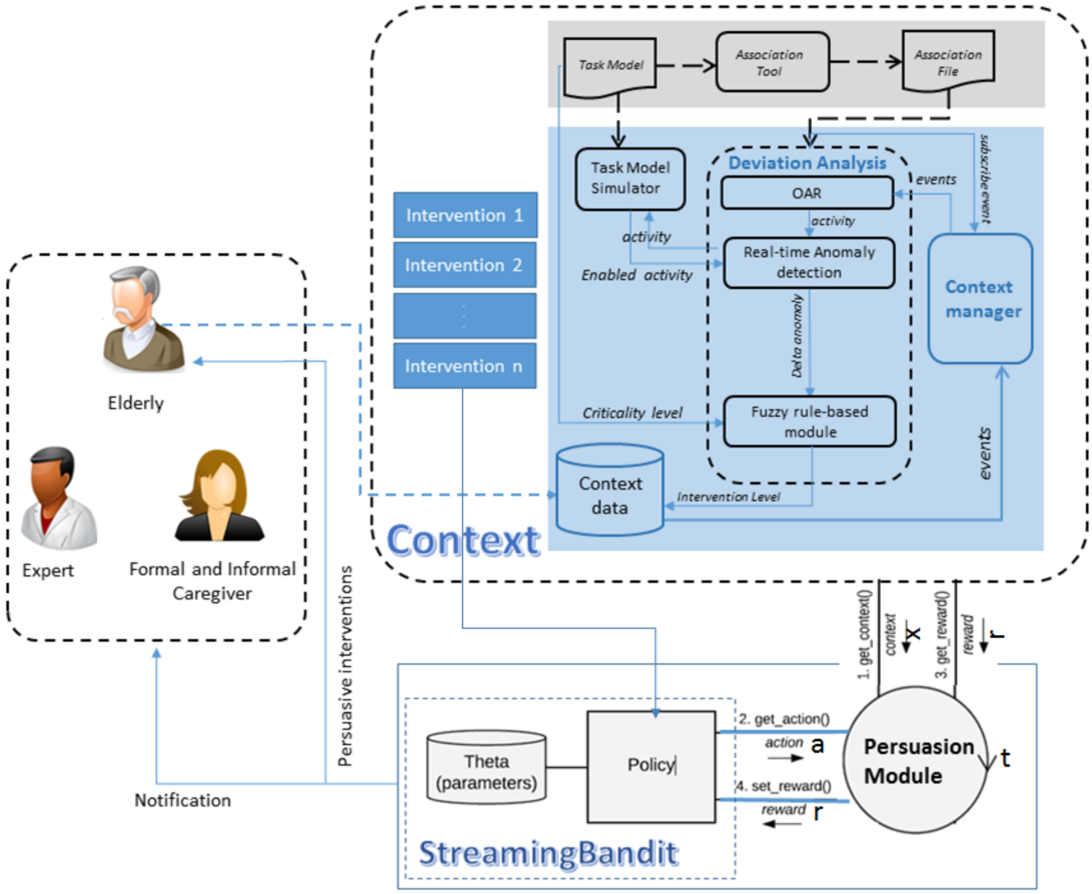


Fig. 1. The Architecture of the System

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 [30] in order to receive the notification each time an event occurs.

4.2. Online Activity Recognition

Activity recognition (AR) aims to identify elderly activities based on a series of sensor readings. In this section, 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" AND "turning off the light". When all these events associated with the activity "sleeping" occur, regardless of temporal relations, 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 event id, 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 ex-

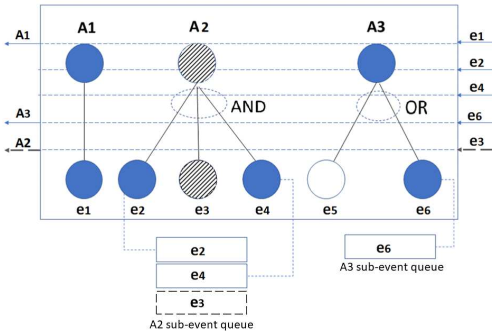


Fig. 2. The sketch of the Activity Recognition taken from IBM [31].

pression AND and OR between them) have arrived, it means that the activity has been completed.

Figure 2 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.

4.3. Online and Personalized Anomaly Detection

Trough the events received via the Context Manager, the online AR module recognizes the associated activity 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. Some details on the process of finding anomalies along with results from the systematic simulation to determine the feasibility of finding anomalous behavior can be found in [9]. The experimental results showed that our online Anomaly detection Algorithm has an accuracy of 95%. The algorithm detects the activity that triggered the anomaly and indicates the anomaly type and the time of the occurrence. Although the simulation carried in the previous study was done considering a short-term user activity (a day), the online anomaly detection algorithm is not limited to this time frame, as it can detect anomalies with respect to the task model. If the task model is extended to cover a longer period, say a week, the algorithm will work accordingly.

When anomalies have been detected, the next step is to measure the Delta error, which shows 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 between a situation characterized as No intervention, Mild intervention, and Strong intervention. We calculated the Delta errors as follows:

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

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.

Our previous model [9] 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. The "emergency treatment" for 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) 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.

We used 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. These rules should be elaborated based on expert opinions.

4.4. *Intervention Level Classification Using Fuzzy Rule-Based System*

We have developed a decision-making system that implements the decision-making model presented in Figure 3. This model is a Mamdani-type Fuzzy Rule-Based System (FRBSs) [32] with three main components: (i) Fuzzification interface, simply modifies and converts inputs into linguistic values to be compared to the rules in the rule base; (ii) a decision rule base includes all fuzzy rules (which holds the knowledge in the form of a set of If-Then rules) for decision-making and an inference engine (which determine how the rules are activated); (iii) Defuzzification is when all the rules that have been activated are combined and converted to a single crisp output in charge of selecting the intervention level.

In fuzzy logic 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). The variables presented in our model are restricted to the activities performed by the user, the $\delta_{anomaly}$ and the criticality level of activities. Amongst the possible elderly activities, in this section, for the sake of simplicity, we only consider "sleeping" and "taking medicine". Using this information as a starting point, we aim to infer the degree of abnormal activity, which

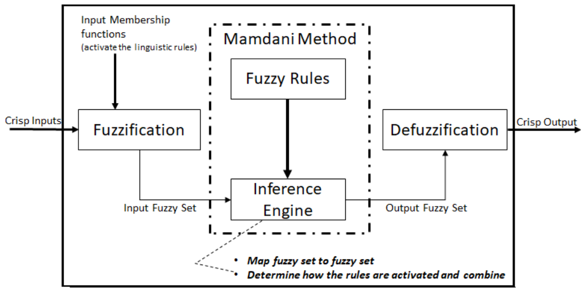


Fig. 3. Mamdani-type Fuzzy Rule-Based System

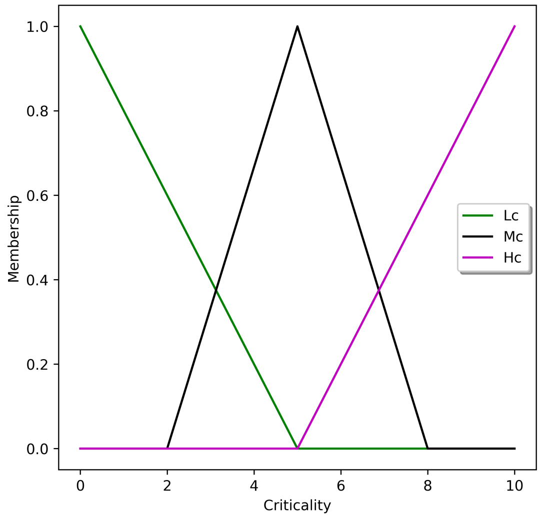
forms a continuous variable with values from 0 to 1 and allows us to distinguish between situations characterized as "No intervention", "Mild Intervention" and "Strong Intervention". The followings are the main components of the fuzzy system for intervention level selection.

A. *Fuzzification:*

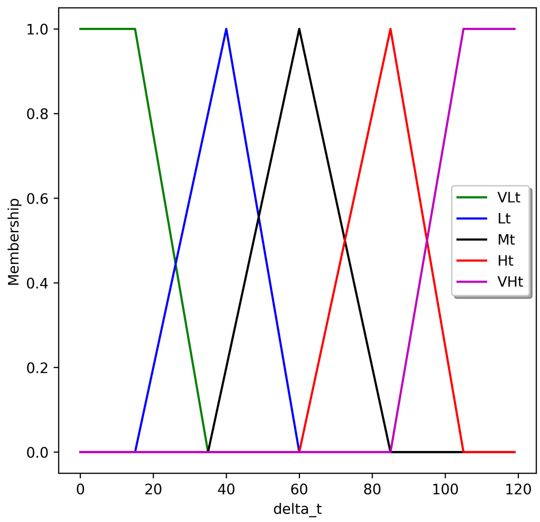
To determine the level of intervention, two inputs are required, defined as $\delta_{anomaly}$ and criticality level. We get the former through the algorithm that calculates the $\delta_{anomaly}$ for each anomalous activity and the latter from the user task model. All input variables are called linguistic variables that later will be separated into linguistic values. To determine the fuzzy sets we used triangular and trapezoidal membership functions (MFs) [33].

For the first input, in case of having $Time$ 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 represent 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 repetition. The second input, criticality level, has a range from 0 to 10 and has three input sets defined as Low criticality (Lc), Medium (Mc) and High (Hc).

The membership functions for both inputs (i.e., criticality level and the $\delta_{anomaly}$) are shown in Figure 4 where every set is depicted as described pre-



(a) Criticality MFs



(b) Delta Time MFs

Fig. 4. Membership Functions of System Inputs

viously. For the sake of simplicity, we only display criticality and δ_{time} which is an element of $\delta_{anomaly}$ in this figure.

B. *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 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:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_1 \ AND....AND \ x_n \ is \ A_n \\ &THEN \ y \ is \ B. \end{aligned} \tag{1}$$

Where x_i (i=1, 2,..., n) are input variables and y is the output variable. $A1, ..., An$ and B 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 Eq. (1), 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 [34]. 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.

C. *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 the 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 Eq. (1). 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 kind 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 [35, 36] 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 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 the anomaly which requires no intervention, and one is the highest degree of the anomaly which requires the strongest intervention. For example, if the input of the system is $criticality = 8$ and $\delta_{time} = 75$, the final output of the system is a crisp value "0.8" as shown in Figure 5. According to the defined output membership functions, this calculated value "0.8" represents the degree of the detected anomaly which corresponds to the "Strong intervention" with the full membership of 1.0 as shown in Figure 5.

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 "strong intervention", it informs the health professionals for further support. Thus, in the following section, we assume an intervention message will only be sent when a "mild intervention" level is encountered.

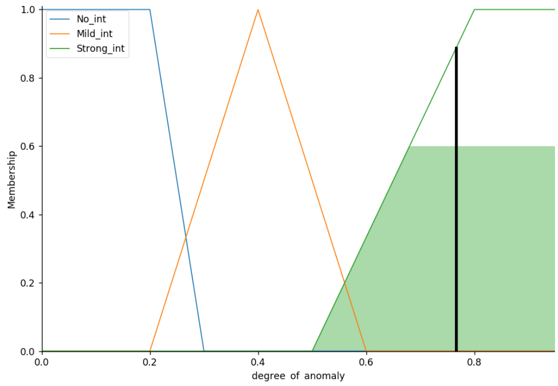


Fig. 5. Defuzzification

4.5. Personalized Persuasive Interventions

Our persuasion module consists of an intelligent suggestion engine, StreamingBandit (see [37]), the design of which was inspired by the contextual multi-armed bandit (cMAB) problem [38]. The cMAB problem is a well-known sequential decision-making problem (see 4.5.1 below) and its formalization 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 which are defined based on the six persuasive principles as described by Cialdini [39].

Interventions: Interventions using short messages may be most effective as a reminder system to support behavioral management [40]. We create several short messages that implement different social influence strategies as defined in [39] that can be delivered to individual users [41]. The six persuasive principles by Cialdini are: *reciprocation* (giving before you get), *consistency* (act consistently with prior commitments), *socialproof* (do as others do), *liking* (the more we like people the more we want to say yes to them), *authority* (trust me, I'm a doctor), and *scarcity* (the less available resource the more we want it). We propose to use three of these in our system: reciprocity (e.g., listen to this relaxing song and go to sleep, including a link to the song), social proof (e.g., 70% of the healthy Adolescent Populations go to sleep on a regular time.) and authority (e.g., your doctor says that sleeping late increase 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.

4.5.1. Contextual Multi-armed Bandit Problem

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 context, the personalized persuasive interventions correspond to the different actions, and the users' performance on future activities correspond to the rewards [42].

4.5.2. StreamingBandit

StreamingBandit is an open-source RESTful web application 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 [43] : 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 [44]: 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 [45]: 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 [46]: Similar to Thompson sampling, however, in this policy the posterior distribution is approximated using an online Bootstrap distribution for computational ease.

4.5.3. 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 (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 1 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 actions 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.

Table 1 lists the possible values the different variables can have in our simple scenario; in reality the number of contextual variables is much larger.

In some respects, the integrated platform (i.e., the online anomaly detection and persuasion module using StreamingBandit) emulates the behavior of a physician who meets different elderly patients sequentially (at each interaction t). For each elderly the physician observes 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.

Table 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"*

5. Simulation and Results

To verify the performance of the two most important modules in our system, namely the "Deviation Analysis" and the "Persuasion module", we ran two separated simulation studies as described below.

5.1. Simulation of the Deviation Analysis Module

In order to have a systematic analysis of the system ability to detect the anomalies and calculate the intervention level needed for the user, we accomplished a laboratory evaluation by simulating the activities received from the Activity Recognition module. We adopted the following simulation methodology:

1. Preparation of the ground truth. To this aim, we create a task model considering user routine that spans a week (i.e., the user follows the same routine during the weekday and a different routine on the weekend). This routine includes activities such as taking medicine, showering, cooking, sleeping, outdoor activity (e.g., the user goes to church on the weekend). The activities of a weekday have different order and time interval $[T_s, T_e]$ from those on the weekend. In this example task model, multiple activities can be freely chosen or performed concurrently.
2. Simulation of the user normal behavior. We obtain 100 normal sequences ($\mathcal{S}$) 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). More detailed information on the task model simulator process can be found in [9]. Afterward, the simulator allocates a timestamp ($t \in [T_s, T_e]$) to each activity in each sequence. The normal sequences $\mathcal{S}$ serve as a ground truth for the system performance validation. Thus, an activity sequence s on $\mathcal{S}$ can be formally express as a triple (s, T_s, T_e), where: $T_s <= T_e$ and T_s is the starting time, and T_e is the ending time, and $s =< (A_1, t_1), (A_2, t_2), ..., (A_n, t_n) >$ is an ordered sequence of activities such that $A_i \in \mathcal{S}$ 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. After generating all possible activity sequences permissible by the task model, the anomalous sequences' simulator takes as input all 100 normal sequences in $\mathcal{S}$ and from each sequence, s selects

Table 2
Experimental results with the simulation

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

randomly an activity. Next, by manipulating and applying one of the seven anomalies described in [9], section 4, generates other 700 anomalous sequences ($\mathcal{N}$). The process of applying the anomalies on randomly chosen activity in each sequence s and generate anomalous sequences $n \in \mathcal{N}$ is as follows:

- *Less*: Omitting the chosen activity.
- *Time_anomaly* (i.e., *Difference-Early-time*, *Difference-Later-time*): Setting the activity time after its start time (t_s) and end time (t_e) respectively. The changes are made in such a way to still respect the activities order.
- *Difference-Order*: Picking a random number between $[0, sequencelength - 1]$, selecting the activity in that index (A_i) and switching it with the next activity (A_{i+1}) in a way that the task time remains in its time interval.
- *Difference-Time-order*: Implying both order and time anomalies as mentioned above.
- *More-number*: Creating another instance of the same activity and locate it in a position that is allowed by the task model with respect to the other activities time and order.
- *More-order*: Creating another instance of the same activity and locate it in a position that is not allowed by the task model with respect to the other activities time and order.

4. Execution of the Anomaly Detection Algorithm over the simulated data. For each generated sequence of activities (n and s), 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 resulting output sequences are used to construct confusion matrices and calculate the algorithm performance measures.

5.1.1. Results

The core of the Deviation Analysis module is the anomaly detection algorithm that analyses one by one, the incoming activities in order 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 5.1. The results of the simulation are presented in Table 2. 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 any 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 3. The measured sensitivity and specificity indicate the capacity of the system of correctly identifying true abnormal activity and the capacity of the system of not generating false positives, respectively. The FPR (false positive rate) shows the proportion of all the cases in which abnormal activ-

Table 3

Deviation analysis performance summary

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

ities have been identified as a normal one. Since each anomaly may have, in general, different interpretation, 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.

5.2. *Personalized Intervention Simulation*

To validate our suggested "Persuasion Module" we implemented a simulation study based on the scenario introduced in 4.5.3. To do so we first specify the data generating mechanism. In concordance with Table 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 [46] where we define an arm for each possible context-action combination and we use a $\text{Beta}(1, 1)$ 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 4.
5. Finally, the observed reward is used to update θ (and thus in this case the $\text{Beta}(\alpha,\beta)$ posterior for each arm).

Table 4

Probabilities for rewards in our data-generating process. 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.

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 6 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 6. 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.

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.

6. Conclusions and Future Work

This paper proposes a real-time and personalized solution performing daily activity recognition, anomaly detection and generation of personalized and persuasive message interventions. The central objective of our method is assisting older adults to maintain functional ability to do their daily routines and live as independently and healthy as possible. The proposed solution models the users' daily routine using a task model specification and performs the online activity recognition through a Context Manager, which detects relevant contextual events occurring in their lives. In addition, the user daily behavior model serves as a personalized knowledge base for detecting abnormal behavior. The system reduces false alarms by combining the user context and detected anomalies using a Mamdani-type fuzzy rule-based system. Finally, the system suggests personalized interventions to users aiming to minimize their anomalous behavior.

We propose the use of a sequential decision policy, implemented in the novel StreamingBandit platform, to select messages adopting distinct persuasive strategies for each individual so that compliance is maximized. 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. The system performs real-time detection of anomalous behavior and is able to send notifications in case a patient needs attention. Overall, the developed system can improve the quality of support in context-aware remote health-care systems and help users to improve significantly the quality of their lives. 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. Such an evaluation should allow us to experiment with different sequential allocation policies in StreamingBandit and properly assess the effectiveness of the proposed system. In addition, there are various health-behavior theories and models for developing effective persuasive intervention messages that will help improve people's health state. While in this paper, 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.

Furthermore, we plan to investigate how to improve the activity recognition by considering the time window for the complex events gathered through sensors and received via Context Manager. In our work, we assumed the complex events are received in an acceptable time window and we did not model the uncertainty of the event timestamps. Finally, we aim to validate our model with real data from monitoring older adults' daily activities in a field trial.

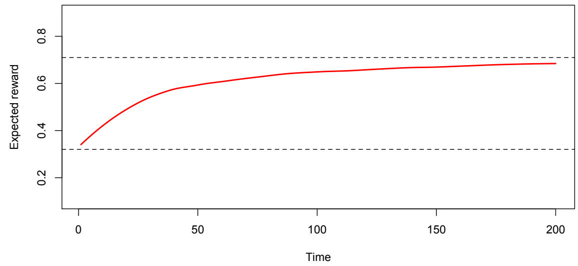


Fig. 6. Performance of Thompson sampling for allocating personalized persuasive messages.

Acknowledgement

We thank Jules Kruijswijk and Robin van Emden for their assistance for implementation in StreamingBandit.

References

[1] U. DESA, World Economic and Social Survey 2007-Development in an Ageing World, Technical Report, E/2007/50/Rev. 1, ST/ESA314, 2007.

[2] W.H. Organization, *World report on ageing and health*, World Health Organization, 2015.

[3] T.G. Moss, C.E. Carney, P. Haynes and A.L. Harris, Is daily routine important for sleep? An investigation of social rhythms in a clinical insomnia population, *Chronobiology International* **32**(1) (2015), 92–102, PMID: 25187987. doi:10.3109/07420528.2014.956361. https://doi.org/10.3109/07420528.2014.956361.

[4] N. Pavón-Pulido, J.A. López-Riquelme, J. Ferruz-Melero, M.Á. Vega-Rodríguez and A.J. Barrios-León, A service robot for monitoring elderly people in the context of ambient assisted living, *Journal of Ambient Intelligence and Smart Environments* **6**(6) (2014), 595–621.

[5] R. Igual, I. Plaza, C. Medrano and M.A. Rubio, Personalizable smartphone-based system adapted to assist dependent people, *Journal of Ambient Intelligence and Smart Environments* **6**(6) (2014), 569–593.

[6] R. Orji and K. Moffatt, Persuasive technology for health and wellness: State-of-the-art and emerging trends, *Health Informatics Journal* **24**(1) (2018), 66–91. doi:10.1177/1460458216650979. https://doi.org/10.1177/1460458216650979.

[7] J. Lacroix, P. Saini and A. Goris, Understanding User Cognitions to Guide the Tailoring of Persuasive Technology-based Physical Activity Interventions, in: *Proceedings of the 4th International Conference on Persuasive Technology*, Persuasive '09, ACM, New York, NY, USA, 2009, pp. 9–198. ISBN 978-1-60558-376-1. doi:10.1145/1541948.1541961. http://doi.acm.org/10.1145/1541948.1541961.

[8] J. Hoey, C. Boutilier, P. Poupart, P. Olivier, A. Monk and A. Mihailidis, People, sensors, decisions: Customizable and adaptive technologies for assistance in healthcare, *ACM Transactions on Interactive Intelligent Systems (TiiS)* **2**(4) (2012), 20.

[9] P. Parvin, S. Chessa, M. Manca and F. Paterno, 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.

[10] M. Amoretti, S. Copelli, F. Wientapper, F. Furfari, S. Lenzi and S. Chessa, Sensor data fusion for activity monitoring in the PERSONA ambient assisted living project, *Journal of Ambient Intelligence and Humanized Computing* **4**(1) (2013), 67–84, ISSN 1868-5145. doi:10.1007/s12652-011-0095-6. https://doi.org/10.1007/s12652-011-0095-6.

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

[12] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu and J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in: *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*, IEEE, 2014, pp. 197–205.

[13] C. Li, W.K. Cheung, J. Liu and J.K. Ng, Automatic Extraction of Behavioral Patterns for Elderly Mobility and Daily Routine Analysis, *ACM Transactions on Intelligent Systems and Technology (TIST)* **9**(5) (2018), 54.

[14] O. Aran, D. Sanchez-Cortes, M.-T. Do and D. Gatica-Perez, Anomaly detection in elderly daily behavior in ambient sensing environments, in: *International Workshop on Human Behavior Understanding*, Springer, 2016, pp. 51–67.

[15] F. Krüger, K. Yordanova, A. Hein and T. Kirste, Plan Synthesis for Probabilistic Activity Recognition., in: *ICAART (2)*, 2013, pp. 283–288.

[16] J. Hoey, T. Plötz, D. Jackson, A. Monk, C. Pham and P. Olivier, Rapid specification and automated generation of prompting systems to assist people with dementia, *Pervasive and Mobile Computing* **7**(3) (2011), 299–318.

[17] N. Banovic, T. Buzali, F. Chevalier, J. Mankoff and A.K. Dey, Modeling and understanding human routine behavior, in: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 2016, pp. 248–260.

[18] G. Mori, F. Paternò and C. Santoro, CTTE: support for developing and analyzing task models for interactive system design, *IEEE Transactions on software engineering* **28**(8) (2002), 797–813.

[19] B.T. Fine, Unsupervised anomaly detection with minimal sensing, in: *Proceedings of the 47th Annual Southeast Regional Conference*, ACM, 2009, p. 60.

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

[21] J.L.C. Candás, V. Peláez, G. López, M.Á. Fernández, E. Álvarez and G. Díaz, An automatic data mining method to detect abnormal human behaviour using physical activity measurements, *Pervasive and Mobile Computing* **15** (2014), 228–241.

[22] A.R.M. Forkan, I. Khalil, Z. Tari, S. Foufou and A. Bouras, A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living, *Pattern Recognition* **48**(3) (2015), 628–641.

[23] B.J. Fogg, A behavior model for persuasive design, in: *Proceedings of the 4th international Conference on Persuasive Technology*, ACM, 2009, p. 40.

[24] A.S. Lan and R.G. Baraniuk, A Contextual Bandits Framework for Personalized Learning Action Selection., in: *EDM*, 2016, pp. 424–429.

[25] M. Rabbi, M.H. Aung, M. Zhang and T. 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*, ACM, 2015, pp. 707–718.

[26] P. Paredes, R. Gilad-Bachrach, M. Czerwinski, A. Roseway, K. Rowan and J. Hernandez, PopTherapy: coping with stress through pop-culture, in: *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 109–117.

[27] S.L. Scott, Multi-armed bandit experiments in the online service economy, *Applied Stochastic Models in Business and Industry* **31**(1) (2015), 37–45.

[28] X. Zhang, Q. Zhou, T. He and B. Liang, Con-CNAME: A Contextual Multi-armed Bandit Algorithm for Personalized Recommendations, in: *International Conference on Artificial Neural Networks*, Springer, 2018, pp. 326–336.

[29] T. Lu, D. Pál and M. Pál, Contextual multi-armed bandits, in: *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, 2010, pp. 485–492.

[30] G. Ghiani, M. Manca, F. Paternò and C. Santoro, Personalization of context-dependent applications through trigger-action rules, *ACM Transactions on Computer-Human Interaction (TOCHI)* **24**(2) (2017), 14.

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

[32] E.H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *International journal of man-machine studies* **7**(1) (1975), 1–13.
[33] C. Wang, A study of membership functions on mamdani-type fuzzy inference system for industrial decision-making (2015).
[34] N.K. Verma and A. Ghosh, *Computational Intelligence: Theories, Applications and Future Directions-Volume I: ICCI-2017*, Vol. 798, Springer, 2018.
[35] M. Masoum, E. Fuchs, M. Masoum and E. Fuchs, Optimal Placement and Sizing of Shunt Capacitor Banks in the Presence of Harmonics, *Power Quality in Power Systems and Electrical Machines* (2015), 887–959.
[36] V. Plevris, G. Kremmyda and Y. Fahjan, *Performance-based Seismic Design of Concrete Structures and Infrastructures*, IGI Global, 2017.
[37] M. Kaptein and J. Kruijswijk, Streamingbandit: A platform for developing adaptive persuasive systems (2016).
[38] J. Langford and T. Zhang, The epoch-greedy algorithm for multi-armed bandits with side information, in: *Advances in neural information processing systems*, 2008, pp. 817–824.
[39] R.B. Cialdini, *Influence*, Vol. 3, A. Michel Port Harcourt, 1987.
[40] F.J. Schwebel and M.E. Larimer, Using text message reminders in health care services: A narrative literature review, *Internet interventions* (2018).
[41] M. Kaptein, B. De Ruyter, P. Markopoulos and E. Aarts, Adaptive persuasive systems: a study of tailored persuasive text messages to reduce snacking, *ACM Transactions on Interactive Intelligent Systems (TiiS)* **2**(2) (2012), 10.
[42] I. Manickam, A.S. Lan and R.G. Baraniuk, Contextual multi-armed bandit algorithms for personalized learning action selection, in: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, IEEE, 2017, pp. 6344–6348.
[43] Y. Xia, T. Qin, W. Ma, N. Yu and T.-Y. Liu, Budgeted Multi-Armed Bandits with Multiple Plays., in: *IJCAI*, 2016, pp. 2210–2216.
[44] M. Collier and H.U. Llorens, Deep Contextual Multi-armed Bandits, *arXiv preprint arXiv:1807.09809* (2018).
[45] M. Kaptein, The use of Thompson sampling to increase estimation precision, *Behavior research methods* **47**(2) (2015), 409–423.
[46] D. Eckles and M. Kaptein, Thompson sampling with the online bootstrap, *arXiv preprint arXiv:1410.4009* (2014).