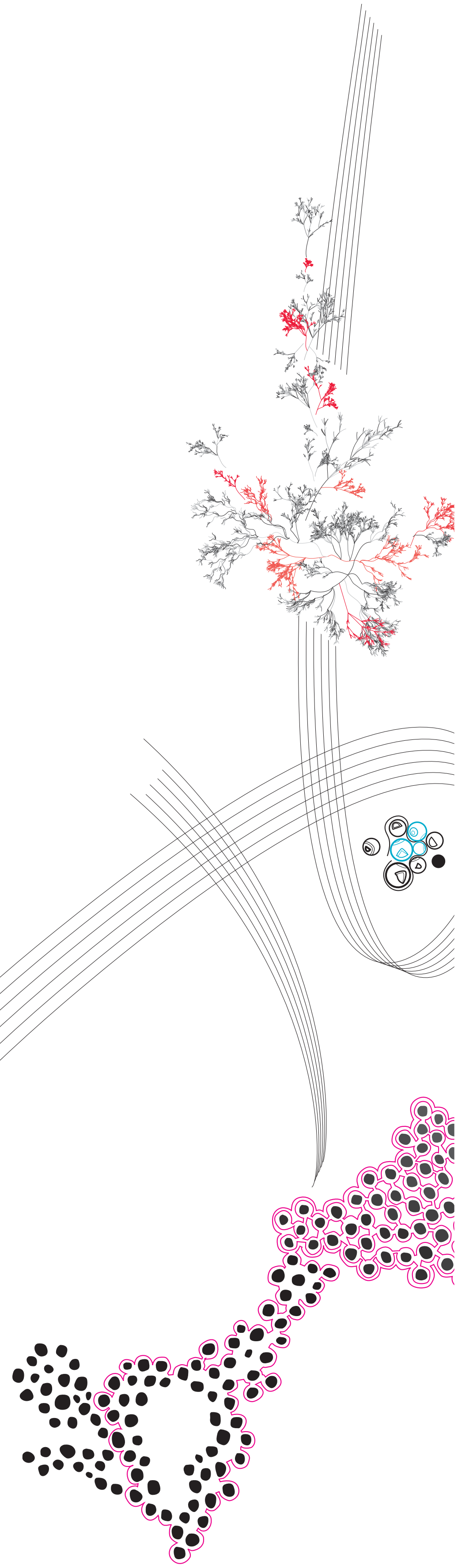
MSc Interaction Technology
Final Report



User-based Tailoring with Reinforcement Learning for an mHealth, COPD-focused Intervention for Increasing Physical Activity

S. Straková

Supervisors:
Dr. M. Poel,
Dr. A. Middelweerd,
Dr. ir. M. Tabak,
Dr. ir. W. D'Hollosy,
Dr. T.C. Beinema

August, 2023

Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

List of Abbreviations

AI	Artificial Intelligence
BS	Behaviour Stage
COPD	Chronic Obstructive Pulmonary Disorder
DSS	Decision Support Systems
EMA	Ecological Momentary Assessment
HAPA	Health Action Process Approach
IRL	Inverse Reinforcement Learning
JITAI	Just-in-Time Adaptive Intervention
MDP	Markov Decision Process
mHealth	Mobile Health
PA	Physical Activity
PG	Policy Gradient
PPO	Proximal Policy Optimisation
RL	Reinforcement Learning
SE	Self-efficacy
T2D	Type 2 Diabetes
TS	Thompson Sampling

Abstract

In this report, a user-based tailoring intervention with the use of Reinforcement Learning (RL) and behavioural theories was developed and evaluated for the MSc Final Project in Interaction Technology. The work was conducted within the context of existing projects, namely the E-Manager and RE-SAMPLE, which both aim to create individualised interventions for managing chronic illnesses, albeit for different user groups: people with Type 2 Diabetes (T2D) and Chronic Obstructive Pulmonary Disease (COPD) respectively. For people with (such) chronic conditions, adherence to physical activity guidelines is lacking, despite proven benefits. Previous studies have been conducted to tackle this problem via mobile health (mHealth) interventions and Just-in-Time adaptive interventions (JITAI). In order to create an effective solution, such interventions necessitate the use of behavioural theories and user-based tailoring. However, approaches that have made use of these concepts tend to often rely on inflexible, rule-based methods. As RL-based algorithms have shown success in adapting to individual user behaviours, an intervention has been developed to adapt intervention content using this approach. The created RL-based algorithm was then tested within a simulation, with user profiles that have differing needs related to e.g., the different behaviour stage that they are currently in. The algorithm selected messages at given times of day that were assessed by the algorithm as relevant to the current user context, and the user's physical activity was monitored. The results have shown a superior performance of the RL-based algorithm over an algorithm that selects the content at random, in delivering relevant content that helped the simulated users increase their engagement in physical activity. Using this approach has shown the potential of RL in creating a JITAI tailored to users with COPD, expanding upon and contributing to the E-Manager and RE-SAMPLE projects.

Keywords: just-in-time adaptive interventions, personalisation, digital intervention design, physical activity, mHealth, reinforcement learning

Chapter 1

Introduction

This chapter presents the motivation and problem statement of the Final Project, as well as its scope and the considered research questions.

1.1 Motivation

For individuals with chronic diseases, keeping up with a healthy lifestyle is imperative, as it provides a largely positive impact on their well-being and management of their disease, for example in terms of physical functioning or the resulting improved quality of life [1].

While a healthy lifestyle can be considered as an umbrella term constituting of various behaviours and habits, sufficient physical activity (PA) is an area that is lacking for many. For example, in the Netherlands, less than 50% of the population fulfils the provided guidelines on physical activity of at least 150 minutes of vigorous activity per week [2]. However, increasing one's physical activity to meet the established guidelines has been shown to encompass many health benefits, such as prevention of illness incidence, deceleration of its course or complications, increase in physical endurance, as well as musculoskeletal strength, and eventually, an extended lifespan, making exercise a beneficial part of disease management and treatment [3].

For example, an exercise-based rehabilitation provided to people with Chronic Obstructive Pulmonary Disease (COPD) is an effective means for functional muscle improvement [4] and can achieve significant increase in the individuals' general fitness and well-being [5], leading to, e.g., lower levels of breathlessness [6, 7, 8] and increased exercise tolerance [8]. Data from a study on older male, as well as female adults with COPD has shown that, when exposed to a 30-day exercise-based training intervention, participants can considerably improve pulmonary function and physical endurance, as well as experience positive psychological effects, such as enhanced cognitive functions and stress reduction [5]. Another study on the effects of exercise-based intervention on patients with COPD has found similar positive effects on cognition in the form of reduction in mood disturbances, such as depression or anxiety, boost in motivation, as well as functional improvements in the area of pulmonary capacity and limitations related to illness [9].

However, such rehabilitation is typically provided in the form of classes or prescribed physical exercises that the patient does not necessarily come into a frequent contact with. This can inhibit habit formation, which necessitates frequent and consistent repetition of the activity [10] without great lapses in between repetitions [11]. There is a general consensus that more long-term programs tend to produce larger and more sustainable effects, and so interventions that offer coaching sessions with a frequency of e.g., once a week do not seem to be sufficient [4]. For example, in a study of 10 weeks with a total of

37 sessions of exercise training, the participants reported a larger increase in the maximum rate of oxygen consumption (during PA) in comparison to previous, shorter and less intense interventions [9]. In a review on health coaching on patients with chronic diseases, more significant outcomes tend to arise in interventions whose length extended to 6-8 months [12].

Therefore, despite the positive effects of exercise on the management of COPD, long-term behavioural changes in PA habits are not observed when the provided interventions are not frequent/intense enough and do not stretch across a longer time period, which can cause that the gained improvements eventually entirely relapse [6]. This calls for intervention design that takes behaviour-change theory into account and allows for a sufficient intervention dose that encourages habit formation and the processing of lifestyle changes.

What is more, using methods such as telephone coaching or face-to-face contact, which can achieve good results within more long-term interventions for patients with chronic diseases [12], this type of approach might be too expensive and there may not be sufficient available resources to coach larger groups of patients. eHealth or mobile health (mHealth) coaching interventions can be useful in addressing this issue, as they can be easily integrated into people's lives in the form of daily coaching that provides frequent reminders and real-time motivational cues to the user to stimulate the target behaviour.

Besides implementing behavioural constructs, it is recommended that the intervention is personalised to an individual user, as continuous tailoring has been shown to achieve greater and more long-term effects [13]. Decision support systems (DSS) have been more commonly put into practice to aid in complex decision-making processes, also when it comes to the area of healthcare. Bonczek et al. define two main types of decision support systems, a procedural and a non-procedural DSS [14]. A procedural DSS follows exact instructions that are provided, while a non-procedural DSS is allowed to provide solutions and decisions on its own, relying on its ability to discover patterns from the data that may not be immediately observable for humans. Such support systems have been found useful within the field of healthcare, for example as expert systems aiding medical practitioners in decision-making on the selection of appropriate treatment [15]. The ability of such a system to decide on a fitting intervention based on the user's context can also be exploited in coaching systems that lead users to achieve a certain goal, following a strategy that is tailored to the individual based on various parameters. In this way, the DSS becomes part of a blended-care where the user is provided with consistent and context-appropriate support, while the data collected from the intervention provided by the DSS can be utilised for further insight and review by a medical practitioner.

1.2 Problem statement

The current study was conducted in the context of existing projects, namely the E-Manager [16] and RE-SAMPLE [17], which both have a similar goal in terms of developing personalised interventions for chronic illness management, but for users with Type 2 Diabetes (T2D) and Chronic Obstructive Pulmonary Disease (COPD) respectively.

Within the E-Manager project, a rule-based intervention based on behavioural theories with different health goals has been developed for the users with T2D. However, an intervention design that can also offer user-based tailoring may provide a more relevant content to the user, which can increase their intervention adherence [18], and so when the intervention is not tailored to the user, a limitation is placed on the ability of the intervention to reach its full potential in influencing the user's behaviour.

With the rise of intelligent DSS and artificial intelligence, Reinforcement Learning

(RL) has shown a potential for being used in such interventions due to its ability to adapt the actions within the intervention to a particular user [19]. However, besides tailoring, behavioural theories need to also be incorporated into the intervention that can address specific cognitive processes of the users that eventually influence whether the target health behaviours are performed [18].

Therefore, the aim of the present study is to develop a user-tailored, adaptive intervention that will, based on the user's context, determine the optimal message to be delivered to the user at an optimised time, in order to stimulate the user to engage in PA and aid in habit building to create positive long-term effects on the users' health and disease management. The intervention will be built upon the content developed for the E-Manager project to expand upon it with an RL approach and to contribute to the RE-SAMPLE project by including COPD-related factors within the intervention.

1.3 Scope

Figure 1.1 provides a graphical overview of the imagined ideal architecture of the intervention that has been developed in the present study. The diagram is an approximate outline of how the system as a whole could work, although some elements might be missing that are not part of the current assignment. For example, there may be more components necessary for connecting to the app, the wearable device, as well as for data collection and processing.

The overall design was inspired by the one of Hietbrink et al. [20], which was part of the E-Manager project, and on which the present study is based upon. There, the users made use of a mobile app, to which a smart watch was connected. The users would receive messages via the app and could react to it in the app as well. However, since they made use of rule-based techniques to decide which message to deliver to the user, the algorithmic part of the system has been added, incorporating the JITAI components from [20] and adapting them to suit the RL nature of the current design. This part was also partially inspired by Gönül et al. [21] where the opportune moment identification and intervention selection comprised of two parallel RL algorithms.

Within the ideal system, the algorithm would have two functions of intervention content selection and opportune moment identification. These could be carried out in a parallel or linear fashion, depending on the possibilities or requirements within the ideal system, hence the dashed connectors within the figure. The algorithm would be connected to an app with a user interface, which would subsequently also collect data from a wearable device (e.g., the user's PA). Additionally, the app would supplement the algorithm with other automatic measures, such as time, or measures from connections to other apps (e.g., calendar availability). This data would then be put together as the tailoring variables to represent the state and context of the user that the RL algorithm can observe, assess and eventually act upon. The decision rules could be used in combination with the RL algorithm to narrow down the set of eligible intervention options. The data collection of the user input, consisting of user reactions to the delivered messages, as well as their state of condition (e.g., presence of symptoms), would serve as a reward mechanism to the RL system. The RL algorithm would also have access to a previously trained general model for overcoming the cold-start problem (see Chapter 2.7.1) and would collect baseline user input at the beginning to assign a general strategy from known similar users. It would then continue to tailor this strategy by learning from the user's feedback on its actions in particular states.

However, due to time restrictions within the current project, only certain elements (marked in yellow) have been developed and implemented from the overall architecture. The RL-based algorithm learns to select an appropriate intervention content based on the received user state, and the timing of the delivery of this content, i.e., a message, is tailored based on selected timeslots. Overall, this resulted in a general model that can be further tailored to individuals when implemented in practice/a real-world scenario. When it comes to the existing elements of tailoring variables, decision rules and intervention options, these have been narrowed down and/or adapted to fit the current context. What is more, while the ultimate goal of such a system is to cater to users with many different chronic diseases, the current scope is restricted to considering the COPD population.

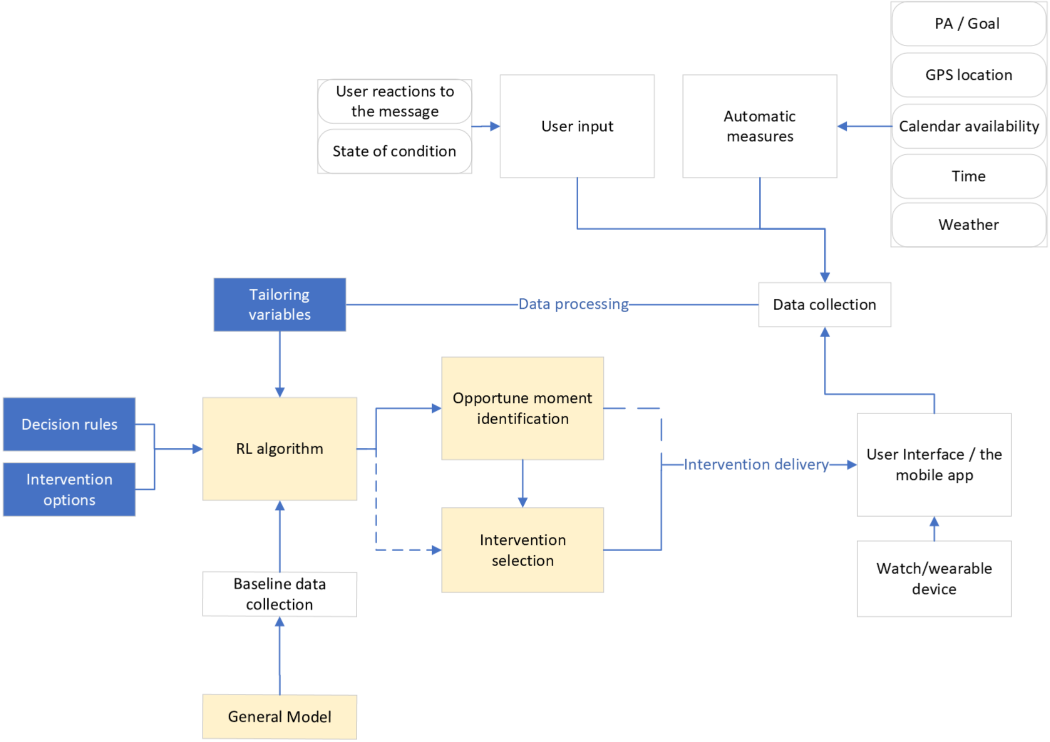


FIGURE 1.1: Graphical sketch of the ideal system as a whole. Blue rectangles denote already existing parameters created for previous studies within the current context that will be adapted to the current system design. Yellow squares refer to the main focus of the present study.

1.4 Research questions

Considering the given motivation and scope, together with the presented problem statement of the current study, the main research question (RQ$_m$), as well as the sub-research questions (RQ$_s$) for the current study have been formulated as follows:

RQ$_m$: *How to employ user-based tailoring in an mHealth, COPD-focused intervention for increasing users' engagement in PA?*

RQ$_{s1}$: *How can Reinforcement Learning be used within an mHealth, COPD-focused intervention that employs user-based tailoring for increasing users' engagement in PA?*

RQ$_{s2}$: *How effective is user-based tailoring in comparison to no tailoring at increasing users' engagement in PA, within an mHealth, COPD-focused intervention?*

1.5 Report structure

The report is organised as follows. First, Chapter 2 outlines the underlying concepts and basis of the research that has been conducted within the Final Project. Next, Chapter 3 presents relevant related work and Chapter 4 describes the steps undertaken within the project methodology in order to answer the given research questions. Finally, the results from carrying out the methodology are presented and discussed in chapters 5 and 6 respectively. Chapter 6 also elaborates on the strengths and limitations of the present study, the main contributions, as well as the possibilities for future research. Finally, Chapter 7 presents the conclusion of the present work.

Chapter 2

Background

This chapter outlines the concepts employed in developing the intervention design for the current study. First, the behavioural theories that have been used as a basis for developing the intervention content as part of the E-Manager project, as well as for the intervention framework design are laid out. Next, considerations for user-based tailoring are explained, and lastly, the basis for the intervention's algorithm is presented.

2.1 Behavioural theories

A relatively large number of studies has been conducted on the development of interventions aimed at increasing PA of users. A systematic review showed that interventions with a foundation in behavioural theories, combining various components involving behaviour change techniques, are more effective in promoting and increasing physical activity than those without, regardless of mode of delivery (e.g., digitally or non-digitally) [22]. The incorporation of behaviour theories in developing interventions targeting behaviour modification allows for a better understanding of why the designed intervention strategy succeeds or fails [23]. Additionally, they can provide a framework for supporting tailored promotion strategies that are adaptable and customised to each individual [24]. By employing behaviour change techniques, a user can be coached to embrace behaviours that correspond to their goals, by gradually influencing them through addressing and transforming the cognitive system that governs their behaviour [18]. Hence, a theoretical basis can aid in developing interventions that are specifically targeting behaviour-influencing factors.

Therefore, several behavioural theories and concepts are described below that have been used for the development of the intervention content in [20], which has also been used in the present study. Additionally, the designed intervention takes these concepts into account, so that appropriate content can be sent to the user at the right time.

2.1.1 HAPA model

When it comes to changing behaviour, one's intention is assumed to be an important indicator for performing an action and was once considered to be the mediator in translating attitudes into actual behaviours [25]. However, while intention is considered a crucial predictor of behaviour change, it is frequently insufficient on its own for facilitating the desired behaviour [26].

When it comes to translating the intention into a (health) behaviour, during this process, the intention is not isolated, but rather challenged by various factors, such as different barriers, other conflicting habits or forgetting [25]. Therefore, other constructs work together with intention that can help to overcome these obstacles in changing behaviour. A behaviour model that involves the postintentional constructs is the HAPA model [27].

The Health Action Process Approach (HAPA) is a way of modelling the relationship between a person's intention to perform a behaviour and the factors that actually lead to action [26]. The HAPA model can aid in designing interventions that target behaviour change, particularly when the goal of the intervention is to promote physical activity [27].

The model is divided into two main phases of behaviour change; the pre-intentional or motivational phase and the post-intentional or volitional phase (see Figure 2.1). Each phase contains specific constructs related to social or cognitive factors that may explain whether a user engages in the desired behaviour [26].

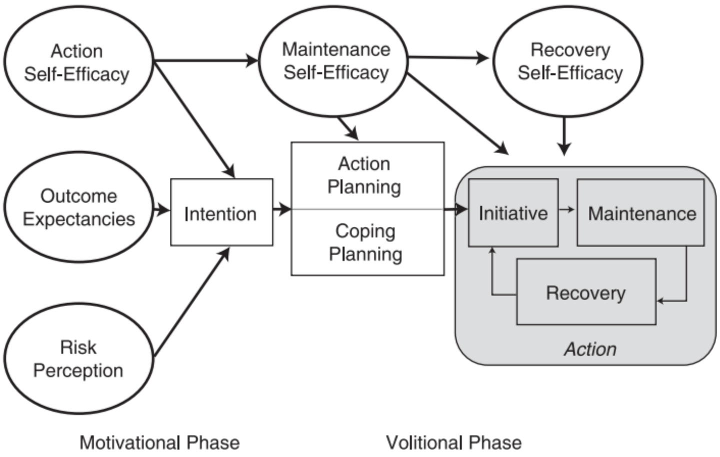


FIGURE 2.1: Diagram of the HAPA model as defined in [26]

The motivational phase constitutes a stage of processes that guide the user to establish an intention to engage in some desired behaviour [26, 27]. In this phase, three constructs play a role in facilitating the intention of the user to engage in a health behaviour (e.g., a physical activity). Action self-efficacy, in terms of a positive belief in one's abilities to perform the given behaviour, outcome expectancies in terms of the perceived balance in positive and negative outcomes of the behaviour, and risk perception in terms of evaluation of potential consequences. If these considerations result in a positive intention towards the action, this must then be translated into specific instructions for performing and maintaining the desired behaviour over time, which involves various strategies and skills, such as planning or recovery self-efficacy. The post-intentional/volitional phase then involves processes that transform the established intention into an actual behaviour [27].

The maintenance of behaviour over time is not considered as a separate phase within the HAPA model, but rather is contained within the volitional phase. However, the maintenance stage may contain different behavioural determinants than the other two stages that are critical for sustaining a behaviour in the long-term [20]. Therefore, for developing the intervention content, i.e., the messages to be delivered to the user, the volitional phase of the HAPA model has been split into two phases of action and maintenance, with their respective determinants that address reframing intentions into actually changing a behaviour within the action stage and sustaining that over extended period of time within the maintenance stage [20].

2.1.2 Self-efficacy

Self-efficacy is one of the key behavioural determinants that is present at each behavioural stage of the model created in [20] (for the development of the intervention content), which is based on the HAPA model, with an added maintenance stage.

Self-efficacy refers to an individual's perceived ability to perform a behaviour successfully in the pre-intentional phase, and one's ability to overcome barriers or setbacks in the later stages [26, 28]. It can help explain the behaviours that people decide to engage in [28] and it is an important predictor for the maintenance of PA [11, 28, 4]. Lower levels of self-efficacy are often present in people suffering from an illness [28], which can prevent a successful behaviour change in lifestyle [4].

It can be influenced in different ways, for example by employing strategies to reach mastery, or different persuasive techniques [11]. Consequently, increased self-efficacy can then play a significant role in influencing exercise capacity [29] and the higher confidence levels in one's ability of achieving a (behaviour) goal lead to better adherence, which increases the likelihood of habit building [18].

Within the current context of COPD users, self-efficacy could be negatively influenced by the presence of different symptoms during the day, that might constitute a barrier towards engaging in PA. Thus, when addressing patients with COPD, the designed interventions should prioritise not only the enhancement of physical functioning but also the improvement of self-efficacy beliefs [30].

One way to address and strengthen perceived self-efficacy is through education, and improved understanding of the present illness [28]. The provided intervention can also emphasise the benefit of self-management skills with which increased control of the illness can be reached via behaviour modification [4].

The concept of self-efficacy also stipulates that the perceived confidence in performing a behaviour is also influenced by successful past experiences with engaging in that behaviour [30] or similar behaviours or goals [31]. When the goals or behaviours are perceived as too challenging by the individual, they are likely to be abandoned due to the perception of a low probability of success [18]. Therefore, due to the individual nature of self-efficacy, an effective intervention should take it into account and be tailored to each user's unique experience and needs. Adaptive goals that result in repeated successful behaviour performance are crucial for increased self-efficacy and, in turn, higher behaviour adherence and longer retention of effects gained from a rehabilitation or intervention [30]. Consequently, the users would be more likely to maintain PA despite encountering barriers.

2.1.3 Reminding

Reminding is another aspect that serves to aid users in sustaining target behaviours. For example, by sending recurrent messages or reminders to the users, these can act as a supporting medium until the user becomes accustomed to engaging in the target behaviour [32].

The content of the messages can also make use of behavioural theories and be designed in a way that it targets the cognitive processes of the user, which when sent to the user at the right time, the supportive content together with the effect of reminding have an additive influence on the user's success of performing the goal behaviour, resulting in improved adherence to the intervention [18].

2.2 mHealth systems

mHealth, which is a part of the broader term eHealth [33], is defined as "*the use of mobile computing and communication technologies in health care and public health*" [34] and is a fast-growing area of research.

mHealth interventions make use of mobile devices, such as mobile phones for purposes such as data collection or clinical decision support systems, with a primary goal of disease management or behaviour change in the form of e.g., text messaging [34].

Interventions utilising mobile technology can target various user groups and has a wide applicability, such as enhancing the diagnosis process, promote adherence to treatment guidelines, improve patient information, and enhance administrative efficiency [33].

However, research suggests that while numerous interventions make use of mobile technology as the medium, many are still often missing any personalisation or adaptation to individual users [18].

2.2.1 Intervention Tailoring

Tailoring a health intervention to specific users is crucial for increased effectiveness [35]. For example, self-efficacy can vary between people with different characteristics [36]. Moreover, in [37], a study on modelling the determinants with a potential for influencing PA levels, Tummers et al. found gender-driven differences in the roles of certain determinants. For instance, attitude was found to take precedence for males, while for females, planning played a bigger role in the intervention structure instead [37]. Accounting for the differences in such needs and characteristics of individual participants can have a positive impact on the the likelihood of PA engagement due to the increased relevance of the intervention to the user [38].

Implementing user-based tailoring in interventions is particularly crucial for people with chronic diseases, since specific health-related factors can and should be taken into account. For example, when prescribing PA interventions for treatment and/or management of chronic diseases, it is advised to factor in individual patient considerations, such as their medical history, fitness-related capabilities, and personal preferences [1]. Exercise-focused interventions should also consider specific limitations of each individual patient when it comes to PA, which can encompass factors such as disease severity, co-morbidities and presence of symptoms while attempting to achieve maximum physiological training benefits [4].

Tailored exercise interventions for people with chronic diseases tend to be typically prescribed as classes or at-home exercises that, however, usually do not become a steady part of the patient's life. Consequently, due to the lack of sufficient repetition of the target behaviour, habit formation is constrained [10]. Therefore, in order to integrate a PA-related coaching support into the patients' everyday life to support habit formation, mHealth as a mode of intervention delivery can be exploited due to easy accessibility. Additionally, mobile devices can also usually be connected to other (wearable) devices that can contribute vital information on e.g., the status of the user's PA without much of the user's (manual) involvement, facilitating further personalisation.

Previous research has investigated what kind of data (extracted e.g., from mobile phones or wearables) could provide the best results for tailored interventions. The findings create a consensus on the combination of group- and individual-level user data for producing the most promising results for creating a sufficiently adaptive intervention [39, 40]. Although a group-level data collected from a wider population may give a starting point for tailoring an intervention, in the end, it is the individual user's opinion or belief that makes a difference in whether the user is successful in performing the target behaviour [18]. Therefore, it is crucial for any tailoring mHealth system to (learn from and) adapt to each user on an individual basis, in order to provide relevant recommendations for ensuring repeated success and intervention adherence. However, according to a meta-analysis on personalised feedback applications conducted in [41], interventions that focus on specific users, rather than user groups are still lacking.

2.2.2 Just-in-Time Adaptive Interventions

A just-in-time adaptive intervention (JITAI) is a type of personalised intervention within mHealth that can provide time-specific and time-adaptive, tailored coaching, aiming to address the user's changing support requirements [42]. In health interventions, the concept of JITAI involves identifying the precise moment when individuals in the intervention require assistance and providing them with the appropriate type of support [43].

The timely reminders are then more relevant to the users' context who are, in turn, more likely to follow their instruction, even when facing barriers, which is in part thanks to creating awareness of favourable circumstances for performing the target behaviour [44].

A framework (see Figure 2.2 for designing effective JITAIs developed by Nahum-Shani et al. elucidates that the intervention should be delivered at the intersection of the user's state of vulnerability/opportunity (i.e., transient states that facilitate engagement in maladaptive behaviours or provide opportunities for positive behaviour modification) and their state of receptivity (i.e., where the timing of the intervention delivery would not be perceived as distracting or disruptive in the user's current context) [42]. The key elements of a JITAIs therefore comprise decision points (i.e., selection of time of delivery), intervention options (i.e., a set of eligible types of support), tailoring variables (i.e., static or dynamic factors that inform intervention selection) and decision rules that put time and content selection into action [42].

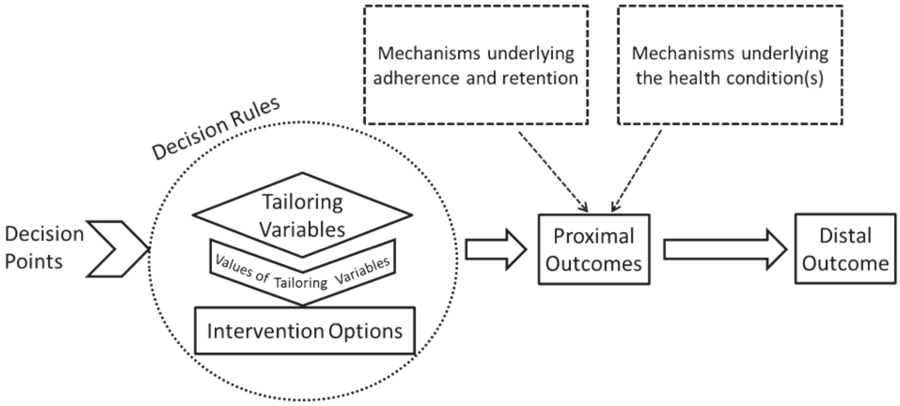

Mechanisms underlying adherence and retention
Mechanisms underlying the health condition(s)
Decision Rules
Decision Points
Tailoring Variables
Values of Tailoring Variables
Intervention Options
Proximal Outcomes
Distal Outcome

Figure 2.2: Framework of a JITAI model and its components as defined by Nahum-Shani et al. [42]

Although the focus within JITAI design often lies within opportune moment identification, delivering content tailored to the user's individual needs is also crucial [18]. The range of messages should also not be small [44]; for example, relevant messages can be selected from a larger collection that contains messages relevant in different ways [42]. A systematic review on JITAIs has also found that, for example only providing feedback on the target behaviour as the content of the intervention is insufficient for changing behaviour; rather, it should also contain additional behaviour change techniques, such as prompts or goal setting [44].

A meta-analytical review on tailored interventions found that JITAIs have more pronounced effects on health outcomes compared to non-treatment controls or alternative treatments [43]. The study also found that dynamic tailoring, as implemented within JITAIs, is an effective component in promoting behavior change. Furthermore, JITAIs were found to be successful across different populations and intervention types, suggesting a generalisable use.

2.3 Markov Decision Process

Markov Decision Processes (MDP) provide a framework for modelling sequential decision-making scenarios. One type of context where the MDP is used, is the formalisation of a problem of learning from interactions, in order to accomplish a specific objective [45]. This scenario is composed of an agent and an environment, and the agent interacts with the environment where the agent selects actions and the environment responds to these actions by presenting new scenarios to the agent (see Figure 2.3). Additionally, the environment provides rewards, which are numerical values that the agent aims to maximize over time by choosing appropriate actions (see Section 2.4).

The agent's actions have an impact not only on immediate rewards, but also on future situations or states, thereby influencing future rewards as well [46].This problem formulation according to the MDP form the basis for Reinforcement Learning (RL) algorithms.

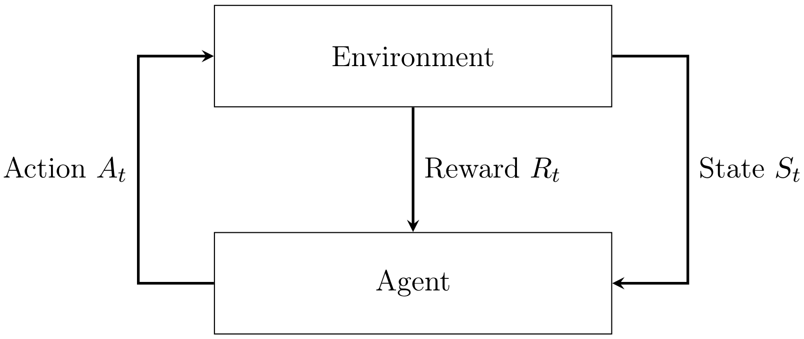


FIGURE 2.3: The interaction between the environment and the agent in an MDP.

Within a finite formulation of the MDP, the state, action and reward (S, A, R) are defined within their respective spaces with a finite number of elements and they are dependent only on the previous state and action [46, 45]. This means that the possible values of the state and reward each have a probability of occurring at a time t, given the values of S and R in the previous step, as shown in Equation 2.1 [45].

$$p(s', r|s, a) = Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\} \tag{2.1}$$

where for all s', $s \in S$, $r \in R$, and $a \in A(s)$. The function p determines the dynamics of the MDP, as the probabilities specified by p fully describe the dynamics of the environment [45].

The MDP framework is versatile and adaptable, lending itself to various problem domains and approaches. For instance, the time steps in MDPs are not limited to fixed intervals of real time; instead, they can represent any consecutive stages of decision-making and action-taking [45]. Part of the state representation may encompass recollections of past sensations or purely mental and subjective elements. Likewise, certain actions can be entirely cognitive or computational in nature.

2.4 Reinforcement Learning

Reinforcement Learning is a sub-area of machine learning techniques that allow a model to learn the best actions to take in observed states by using a trial-and-error approach over multiple iterations, with the goal of maximising return [45]. An RL algorithm is typically composed of three primary components; a policy, a reward signal and a value function, and sometimes it also includes an optional fourth element, a model of the environment [45].

The policy of the algorithm, denoted as π, is the learned approach towards choosing an action A at a time t, within a certain observed state S, and can be represented as

$$\pi(A_t = a | S_t = s) \tag{2.2}$$

where each action is taken from the given action space and the observed state belongs into a space of possible observations:

$$\forall A_t \in A(s), S_t \in S \tag{2.3}$$

This approach is acquired from first interacting with the environment over many iterations where, at each time step t, the environment provides a state $St_2 \in S$ to the agent, according to which the agent selects an action $At_2 \in A(s)$ [45] Following each action, a reward $R_{t+1} \in R \subseteq \mathbb{R}$ is assigned to the agent, which refers to a numerical value that the agent aims to maximise over time. Subsequently, the agent receives a new state S_{t+1}. This cycle, thus, results in a trajectory, as shown in Equation 2.4 [45].

$$\{S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, ...\} \tag{2.4}$$

Each action and its resulting reward can have an impact on subsequent states and actions, leading to a change in the probability distribution of potential actions not only in the present state but also in future states. The agent can exploit the reward signal in order to adjust its policy by, e.g., favouring actions that tend to result in higher rewards in certain states or under certain conditions. Over time, the agent learns the associations between the observed states of the environment and the appropriate action for the agent to take, which determines its behaviour.

In contrast to the immediate feedback provided by a reward signal, a value function is a measure of the long-term appeal of a specific state, considering the projected accumulation of future rewards from subsequent states. This allows the agent to weigh potential outcomes and make decisions that optimise its overall performance.

Lastly, when the RL algorithm has access to a model of the environment, the model acts as a replica of the environment it will operate in, from which it can infer the environment's behaviour. Using these components, the RL agent is capable of learning the optimal mappings between the observed state and its actions.

Reinforcement learning has made significant achievements in terms of theoretical and technical advancements in various areas, such as generalisation or efficiency. As a result, RL has become increasingly relevant for addressing real-life problems, one of them being healthcare related applications [47] where RL-based algorithms can help to facilitate personalised medicine [48].

Thanks to such learning capabilities, RL algorithms constitute a suitable approach towards personalising health interventions, as the agents can adapt its intervention strategies to the needs and behaviour of each individual user.

2.5 Reinforcement Learning taxonomy

There are different ways of categorising RL-based methods. Sutton & Barto categorise, on the highest level, a division between tabular and approximate methods [45]. According to this distinction, while tabular methods look for exact solutions to a given problem, i.e., an exact, optimal value function or policy, approximate methods only approximate a solution, which makes them more suitable for dealing with much larger problems. Some examples to the tabular methods involve multi-armed bandits or Q-learning, while for approximate methods, Sarsa or REINFORCE are common algorithms.

For a further classification, according to Zhang & Yu, RL-based methods can be divided based on several criteria; the use of a model, policy or value function and an off- or on-policy approach [49].

Table 2.1 summarises each of these methods together with some of their main advantages and disadvantages.

2.5.1 Model-based vs. model-free

The major distinction between model-based and model-free RL approaches is that model-based methods are able to access the model of the environment they will operate in (e.g., the reward and transition functions), while model-free methods must learn this model in the process of interacting with the environment [50].

Some of the elements contained in RL include the action space, state space, reward function, transition probability and a discount factor to the reward. If these elements were known, we would be able to employ planning techniques without ever having to interact with the environment. However, in most cases, the agent is required to learn the reward and transition functions via numerous trial-and-error interactions with the environment to observe and eventually exploit the reward feedback. [49]

To achieve this learning, a model-free or a model-based methods can be employed. A model-based algorithm takes actions in the environment to gain sufficient samples, which can be used to predict the reward and transition functions, and consequently, directly use planning methods [49]. On the other hand, a model-free approach aims to directly find the optimal policy by searching for maximum rewards [49].

The fundamental advantage of having access to the model of the environment constitutes the ability of inferring the environment's behaviour, which allows the algorithm to plan ahead and consequently extract the policy from this process [49]. What is more, these methods are more efficient in comparison to model-free methods, being able to learn from fewer interactions with the environment.

On the other hand, there are several disadvantages to model-based approaches. The model of the environment can be difficult to create and in the end, it can encompass biases that lead the algorithm to score well in testing but fail when deployed in the real environment. What is more, these methods tend to be more computationally expensive and when changes in the environment occur, the algorithm needs retraining to account for these changes.

Therefore, while model-free methods suffer from potentially reduced efficiency, they tend to be simpler in implementation and tuning, and are generally more popular to use [50].

2.5.2 Policy- vs. value-based

The distinction between policy and value-based RL algorithms lies in *what* that the model learns, e.g., a stochastic or a deterministic policy, action-value function (Q-function) or a value function [50].

Value-based algorithms aim to first optimise the value function, from which favourable policies can be derived [51]. In each state, the value, or goodness, of each state or state-action pair is evaluated, which involves predicting the expected future reward [51].

Policy-based methods, on the other hand, learn a parameterised policy capable of selecting actions without relying on a value function [45]. While value-based algorithms first acquire knowledge about the values assigned to actions and then select an action based on their estimated values [45], in policy-based algorithms, policy optimisation is employed to find the optimal combination when creating a mapping between a state and the actions or a distribution of actions [51]. One of the advantages of the value-based algorithms is that they are much less likely to get stuck in a local optimum, however, their disadvantage is that they often cannot deal with continuous action spaces [49].

The main advantages of policy-based algorithms lie in their better convergence properties thanks to the algorithm's smaller-paced updates, which can be effective in high-dimensional action spaces where it might be too computationally expensive to always aim to identify the optimal action with the highest value [52, 51]. Instead, incremental changes are used to gradually arrive at the best action to take, rather than directly estimating it at each step.

In this way, policy-based algorithms can learn stochastic policies, which ensures exploration (see Section 2.7.2). For example, if a deterministic policy is used in playing rock, paper, scissors, such as always playing rock, this can easily be exploited by the opponent and the algorithm eventually loses every time. Applying this to the current context, with a deterministic policy, the same message would always be sent within an intervention, which is not desirable, as it would quickly result in participant drop outs.

Policy parametrisation (optimisation) may additionally be easier to estimate than action-value optimisation, resulting in faster learning [45]. On the other hand, with the use of policy-based algorithms arises the risk of converging to a local rather than a global optimum due to the smaller scope of updates in the optimisation process [52].

2.5.3 On- vs. Off-policy

The distinction between on- and off-policy methods lies in whether the algorithm is evaluating and improving the same or a different policy [51].

On-policy methods work with a single policy that is always evaluated and improved. In this way, the algorithm rather reaches a near-optimal policy where exploration is still involved [45].

Off-policy algorithms work with two or more policies; a target policy, i.e., the one being learned and optimised, and a behaviour policy, i.e., the one that is generating the agent's behaviour [45]. In this way, the behaviour policy is used for exploratory purposes, while the

RL Model Taxonomy			
Model type	Definition	Advantages	Disadvantages
Model-free	Has to learn the model of the environment	Easier to implement and tune	Requires more interactions for learning
Model-based	Has access to the model of the environment	Can learn from fewer interactions	Biases in the model can undermine real-world performance
Value-based	Optimises a value function	Does not easily get stuck in local optimum	Cannot handle continuous action spaces
Policy-based	Optimises the policy directly	Better convergence properties, stochastic policy	Risk of getting to local optimum
On-Policy	Evaluates and improves the same policy	Simpler to implement	Cannot make use of other policies
Off-Policy	Works with multiple policies	Can learn a more general policy	Can suffer from high variance and slower convergence

Table 2.1: Summary of the main categorisation in RL-based methods.

target policy is being improved and eventually becomes the optimal policy/value function, without necessarily conforming to the behaviour policy [51].

While the off-policy methods may benefit from eventually learning a more capable and general policy, they might suffer from higher variance and slower convergence due to the multi-policy approach [45]. On-policy algorithms tend to be simpler and are often considered as the first option.

2.6 Policy Gradient Methods

Considering the taxonomy outlined above, Policy Gradient Methods can be categorised as a group of model-free, policy-based, on-policy algorithms that aim to find an approximate solution. These algorithms can be used with both discrete and continuous (user) state and action spaces, however, the discrete configuration is relevant for the present study, and so the theory is limited to that.

Policy Gradient (PG) methods do not make use of the value function in order to select an action within a state, but rather optimise the policy directly [45, 51].

In policy parametrisation (i.e., approximation), the process of optimising the policy, the algorithm makes use of a parameter vector $\theta \in \mathbb{R}^d$ with d dimensions and the policy is parameterised according to Equation 2.5 as the probability that an action a is taken at a time t, given an observed state s and the parameter θ at a time t [45].

$$\pi\left(a|s,\theta\right)=\Pr\left\{A_t=a\left|S_t=s,\theta_t=\theta\right|\right\} \tag{2.5}$$

The goal of the algorithm is to find a policy that maximises the expected reward, and so this involves learning the parameters θ that maximise the policy's performance. This can be done by updating the parameters using gradient ascent ∇ of a performance measure $J(\theta)$ in regard to the parameter vector, utilising an update rule as shown in Equation 2.6 [45, 53].

$$\theta_{t+1}=\theta_t+\alpha\nabla J(\theta_t) \tag{2.6}$$

When using a neural network for policy parametrisation, the vector of parameters θ represents the set of weights and biases within the network [53].

When considering discrete action spaces that are not too large, a commonly used parameterisation approach is the soft-max distribution:

$$\pi\left(a|s,\theta\right)=\frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}} \tag{2.7}$$

where $h(s,a,\theta)\in\mathbb{R}$ represent the function that determines the numerical preference of choosing an action a in a state s, based on the current policy parameter θ [45]. The e represents the base of the natural logarithm and the denominator ensures that the probability distribution in each state sums up to 1. The action with the highest preference in each state subsequently obtains the highest probability for being chosen.

Some advantages of the PG algorithms include that thanks to their ability to learn stochastic policies [45], the action preferences (probability of taking a certain action) undergo gradual and smooth change over time, ensuring better convergence properties [54]. What is more, via outputting the probability distribution for the actions, the algorithms explores the state space without invariably following the same trajectory. However, the probability of random actions is still minimised when parameterising according to soft-max, as it can eventually approach a deterministic policy [45].

On the other hand, due to the step-wise updates of the policy parameters, it can take longer (in terms of wall-clock time) to train the algorithm and altering the policy parameter in a manner that guarantees improvement may appear challenging, as performance relies on both the selection of actions and the distribution of states in which those actions are taken [45]. The agent can then end up converging to a local maximum instead of a global optimum [54]. Moreover, the algorithm can suffer from high variance, although there exist techniques that deal with this, see section 2.6.1

2.6.1 REINFORCE

REINFORCE [55] is an algorithm belonging to the group of PG methods [55], which makes use of a Monte Carlo approach to estimate policy gradients [51].

Keeping in mind the overall approach of stochastic gradient ascent defined in Equation 2.6, in REINFORCE policy updates, the complete return from time t is used, encompassing all future rewards up until an episode is finished, and instead of updating the parameter at each time step, the entire trajectory of an episode is sampled with θ as a static parameter, which is only updated retrospectively once the episode is finished [45, 56].

Within the episodic updates, the increment is determined by the product of the return (a cumulative discounted reward at the end of an episode), denoted as G_t, and a vector corresponding to the gradient of the probability associated with taking the action that was actually chosen, divided by the probability of selecting that specific action, as shown in Equation 2.8 [45, 56]. The vector signifies the direction in the parameter space that maximises the likelihood of repeating the action A_t during future visits to state S_t. The update adjusts the parameter vector by scaling it in this direction, proportionally to the return (ensuring a stronger movement in directions favouring actions with higher returns) and inversely proportional to the action probability (preventing frequently selected actions from gaining an unfair advantage, as the updates would predominantly favour them, even if they do not yield the highest return).

$$\theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)} \tag{2.8}$$

where α denotes the learning rate, which represents the size of the update on the current parameter vector [57].

The Equation 2.8 can then be rewritten into a more compact representation of the REINFORCE update, such that

$$\theta_{t+1} = \theta_t + \alpha G_t \nabla \ln \pi(A_t|S_t, \theta_t) \tag{2.9}$$

which follows from $\nabla \ln x = \frac{\nabla x}{x}$.

REINFORCE with baseline

However, one of the downfalls of the REINFORCE algorithm is that, due to the usage of the entire return, which can show great variability between episodes, the gradients consequently often suffer from significant variance. Such variance then results in unstable updates during learning, which lead to slow convergence, and consequently hinder learning of the optimal policy.[58].

Therefore, to improve the stability and learning of the algorithm, REINFORCE with baseline algorithm subtracts a baseline function from the returns at each update step when calculating the gradient [45, 58], as shown in Equation 2.10.

$$\theta_{t+1} = \theta_t + \alpha(G_t - b(S_t))\nabla \ln \pi(A_t|S_t, \theta_t) \tag{2.10}$$

There exist different approaches towards defining the baseline function, one of them being *whitening* of the returns of the episodes, which scales the returns G_t using mean and standard deviation [58]. This approach, often applied in the area of deep learning, is often considered crucial for effective optimisation.

2.7 Common challenges in RL

When implementing an RL algorithm, there exist numerous challenges that arise, the most common ones being the cold-start problem and the exploration-exploitation trade-off.

2.7.1 The cold-start problem

When developing an RL algorithm, in order for the algorithm to learn the optimal actions that yield the greatest reward in the long-term, i.e., that are able to provide the most suitable personalisation for a user, many iterations of interacting with the environment, and thus the user, are necessary to find the best policy [59]. However, at the beginning, there is typically no or very little data available of the user (i.e., experiences of the model with the user) that the agent can make use of to provide sufficient personalisation, which also known as the cold-start problem [60].

One way to approach this issue is to make use of available data from other users that can be clustered into groups and each new user can be assigned to a cluster based on certain characteristics, which can then provide more insight into their potential behavioural patterns (e.g., in terms of PA) [60]. In this way, a generalised model can be created that is not fully tailored to each individual but that can at least initially personalise its actions on a group level, as opposed to random message selection, in order to find an optimal strategy.

2.7.2 Exploration vs Exploitation

Another challenge specific to RL is the exploration-exploitation trade-off [45]. Within an RL algorithm, the agent aims to maximise the return, and thus, is motivated to prefer actions that have been found successful (in yielding high rewards) in previous interactions. However, in order to find such actions and discover additional effective strategies, the agent also needs to experiment with new actions that have not yet been tested.

In the context of mobile health interventions and JITAIs, although a predominant exploration may be valuable for (re)learning users' preferences that are an important element for intervention tailoring, burdening the users with interventions that are ineffective is best avoided [19]. Likewise, solely relying on exploitation is a sub-optimal approach, as this consists of adhering to previously learned strategies without exploring the possibility of changes that might have arisen in the user's responsiveness to the an intervention. Thus, the agent must balance between leveraging past experiences for acquiring reward (exploitation) and searching for new actions to improve future action choices (exploration) [45, 19].

When it comes to PG methods and the exploration-exploitation trade-off, in the REIFNORCE algorithm, the policy never becomes deterministic, which ensures exploration, however, it can *approach* a deterministic policy, which allows it to minimise the probability of random actions [45].

Chapter 3

Related Work

In this section, first, studies that developed interventions based on modelling or rules are reviewed. Then, several studies are reviewed that have also made use of RL algorithms in their pursuit of adapting either the content, the timing of the intervention or both.

3.1 Model and Rule-based Interventions

Several studies have explored implementing DSS within various health interventions targeting, for example, weight loss or increasing PA, in order to select a sufficiently personalised intervention content and/or timing of the delivery to users based on, for example, initially collected information (e.g., user demographics), user clusters or individual user profiles.

In [61], Klein et al. have used model-based reasoning techniques for tailoring the intervention's coaching methods to the needs of the users. A dedicated app called Active2Gether collected data from questionnaires delivered to the users, as well as sensor measurements, which the model could interpret, in order to determine the type of necessary support (educative, coaching or feedback) for each user, which was updated every three weeks to keep adapting to the user's state. Additionally, once a week, the model predicted the most promising behaviour determinants (e.g., intention or self-efficacy) that would, besides factors such as the environmental context, help determine the most effective intervention content (i.e., messages) to be delivered to the user. Within the coaching phase, the user was prompted to select an activity domain, as well as a weekly goal related to the selected domain. The dashboard of the app then showed a virtual coach greeting the user, the user's progress towards their weekly goal, graphical visualisations of step counts, as well as data of other users, and coaching messages.

The designed intervention was evaluated in two separate user studies for effects on PA levels [62] and user experience of the delivered messages [63]. Although no significant differences in PA levels were found between the control group and the participants using the Active2Gether app, this seems to have been influenced by implementation and recruitment challenges [62]. On the other hand, the results of the user experience evaluation study of the coaching messages showed that the app's coaching aspect was perceived positively by 42% of the users, although a balance in the frequency of the user-intervention interactions must still be found. Moreover, the adherence of users that were using the Active2Gether app, rather than self-monitoring Fitbit app, was higher for a larger amount of time, as indicated by a dropout rate of about 54% in the Fitbit condition, showcasing the potential of the Active2Gether intervention [63].

In [18], Mohan created an adaptive and interactive mHealth intervention, PARCCoach for weight-related behaviour change based on cognitive theories underlying the principles of behaviour change, as well as individual tailoring. The intervention was deployed in the form of an app that allowed the user to select a goal and its corresponding target behaviours. Additionally, the users were asked to identify the context cues, such as time of day, location, social context, reminder period (e.g., 30 minutes prior to engaging in the behaviour) that would increase the likelihood of them engaging in the selected behaviours. The participants were then sent daily reminders to perform a target behaviour when appropriate contextual cues predefined by the user were detected. The main goal of the system was to strengthen the associations between the context and the target behaviours to ultimately achieve habit building and sustained behaviour performance. Besides receiving a reminder, the users were also prompted to fill in daily reports on whether they performed the target behaviour, as well as their perceptions pertaining to the difficulty of and their self-efficacy towards (future) performing of the behaviour. The developed system was evaluated in a 4-week long study and the findings showed a positive impact of the reminders on behaviour compliance, as well as the importance of individual tailoring. The reminders resulted in a higher number of filled reports (on the target behaviour) by the participants, and adherence to successfully performing the target behaviours, contributing towards habit building by creating an association between the context cues and the target behaviour. The individual perceptions of the participants towards the difficulty of the target behaviours and their perceived self-efficacy towards completing it played a significant role in engaging with the behaviour, regardless of whether the task was deemed objectively easy or hard. Negative difficulty estimations and lower self-efficacy led to lower compliance, while participants with higher self-efficacy were more likely to perform the target behaviours. These results indicate the importance of not only adapting the intervention task levels to each user individually, but also of potentially positively influencing users' self-efficacy for increased self-belief, and thus, behaviour compliance.

In the context of the E-Manager project, a recent study in [20] has been conducted with users with T2D where the content of the intervention messages was, for the duration of the intervention, tailored to individual users based on several variables: *duration of intervention use*, *type of chronic disease*, *time of day*, *type of behaviour goal*, *goal achievement*, and *identified barrier towards goal achievement*. Each of these variables were adapted to the information collected at onboarding or throughout the study, and they were incorporated into a fixed set of rules according to which the system made a decision on which message to send to the user. The rules were formed based on behaviour change theories, so that, for example, the motivational messages would reflect the stage of behaviour change that the user was currently in, as well as the relevant determinants of behaviour. The messages were delivered to the user via a mobile app and the user could like or dislike the messages. The results of the study have shown that the participants had a positive outlook on receiving the messages, which were found to be motivating and informative, and having positive effects on lifestyle, although they were perceived as not always fully tailored to the users' context.

3.2 Content-based RL interventions

In [64], Forman et al. conducted a feasibility study on the use of RL-based algorithm for the selection of an intervention type within a weight-loss programme. After a 4-week long, in-person group sessions targeting weight-loss that occurred 4 times a week, the participants were followed through a 12-week long remote coaching. The remote coaching was received twice a week by the participants that were divided into two groups, receiving either a non-optimised intervention or an RL-optimised one. There were three types of interventions that the users could receive: a call with the coach on goal progress, a text exchange or an automated text. Within the optimised condition, the RL-based algorithm adapted the intervention based on an individual or a group level, taking into account the received reward and time it took to complete each intervention (e.g., 0 min for sending a message and 12 min for a call). The results of the study showed that both of the RL-optimised conditions showed a significant reduction in time costs (2.5 times less than the non-optimised condition) for the clinicians, while retaining the weight-loss progress of the participants. What is more, the RL-based algorithm was able to adapt to the participants' needs by selecting either a more intensive intervention when necessary or a less intensive and more cost-saving one when this was deemed sufficient for the user.

In [65], Zhou et al. utilised an RL-based algorithm on top of predictive modelling of a user's PA based on historical data for adaptive goal setting via an app interface to encourage PA increase. Within the first week of the study, data was collected on the participant's step count when the goal increased each day at fixed and predetermined intervals, identically for all participants. After one week, the step goals were then set individually per user in an adaptive fashion. The implemented algorithm adapting the users' goals consisted of two parts. First, inverse reinforcement learning (IRL) was used to estimate each user's self-efficacy, the number of steps they would decide to take, as well as the rate of how the user's happiness would increase when the user approaches their goal. The parameter of measuring a user's rate of happiness was based on the concept that individuals aim to increase their overall satisfaction. Therefore, a person has a preferred daily step count that involves a trade-off between a low number of daily steps, which can lead to dissatisfaction due to physical inactivity, and a high number of daily steps, which requires a significant amount of effort and time to achieve, but can result in happiness from approaching the goal. Additionally, self-efficacy was selected as it plays a significant role in goal setting, referring to a user's confidence in their ability to perform a series of actions successfully, which can predict the levels of PA that the user will engage in. Given the estimation of these parameters, an RL-based algorithm was employed, tasked with searching for the optimal policy, i.e., the optimal goal for the user. The algorithm was then applied at the end of each week to dynamically estimate step goal for the following week from the previous week's PA data, with the aim to challenge the users, but at the same time, keep the goals attainable. The goal updates were kept to once per week to reduce possible influence of significant changes in the goal values. Push notifications were delivered twice a day to the user as a reminder of their goal however, the focus of the intervention was not on the message content itself. Rather, the users were shown their goal progression on a dashboard within the dedicated mobile application. The system was evaluated within a 10-week study, and was able to significantly increase step count within the intervention group in comparison to the control group, which resulted in an average difference of 2220 daily steps between the two groups at the end of the study. Within qualitative interviews with the participants, it was shown that adaptive goal-setting is an important feature that provides the users with insight on how well they are able to keep up with their PA levels (i.e., their PA is insufficient if the goal decreases), which provides motivation to engage in PA. The significance of estimating a user's self-efficacy also transpired in the results, as it was shown that when the step goal remained challenging and fixed (within the control group), it deterred users that were repeatedly unable to complete it from even trying to work towards it.

In [66], Martinho et al. developed a multi-agent system to identify the sequence of messages that corresponds to individual user preferences and has the highest potential to influence the user to perform health-related behaviours. The developed system involved a personal agent (that interacts with the user (i.e., sends them messages)) and a coaching agent (that interacts with the personal agent), which both have different knowledge on the context and preferences of the user and where the coaching agent makes use of a Q-learning-based (off-policy) RL algorithm to identify suitable messages to deliver to the user. Since the personal agent had insight into the user's preferences on certain types of messages, as well as other contextual preferences (e.g., which messages would have the most impact at what time of the day), this agent could reject the messages suggested by the coaching agent, which then had to generate a new message until the frequency restriction of three messages per day was fulfilled, producing an optimal sequence to send to the user.

The system was evaluated in a simulation and its performance was contrasted against the same system without the RL element. The results showed that the system including the RL algorithm outperformed the one without in terms of the number of times the coaching agent was able to produce an accepted sequence of messages after 30 simulated days, with an average of 28 and 2 successful sequences respectively. Unlike the non-RL-based agent, the RL-based one was also able to sustain this performance over multiple simulation scenarios with increasing number of considered messages.

Additionally, the evaluated accuracy of each system, i.e., the number of successful sequences in relation to the number of produced messages, was found to be superior for the RL-based system, with an accuracy of 100% after reaching day 4, 10 and 13 in the simulation across three scenarios with increasing number of messages (2, 4 and 6 messages), whereas the accuracy of the non-RL-based agent remained below 55% throughout the first scenario and below 20% in the following two scenarios.

3.3 Time-based RL interventions

Liao et al. used RL to decide whether to send a message to the user at pre-specified times of the day and when availability was determined, aimed at switching their sedentary behaviour to being more physically active [67]. The intervention content delivery was context-triggered and an exercise-based reminder was sent at five user-specified times of the day when prolonged sedentary behaviour was detected. At each decision point, the algorithm decided whether to send the message to the user or not, also in adherence to a constrained total number of messages to be sent per day. Additionally, the state of availability of the user was used to determine the appropriateness of the timing, for example, a message was not delivered while the user was engaged in driving. The reward for the algorithm was based on the total number of steps recorded within a 30-minute window following a delivered message. The algorithm was then updated at the end of the day with this information, in order to take into account the entire day's context. The model was tested within a simulation that made use of organic participant PA data collected previously from a healthy population of users. The designed algorithm, which also considered the potential long-term effects of its current action, was compared to a Thompson Sampling (TS) Bandit model, whose action selection aims to maximise the immediate reward. The two models were compared in terms of increase in the average of total rewards (i.e., increase in user steps) and the main algorithm showed a 6% increase over the TS

Bandit model in raw step count of the simulated users within the 30-minute interval following a delivered reminder. On top of a simulation, the algorithm was also evaluated in a pilot study with human subjects. Comparing the participants' number of steps within the 30-minute windows following each decision point with or without receiving a walking suggestion, the presence of the reminder positively influenced the participants' PA, with an average increase of 125 steps [67].

The work of Wang et al. [68] is the most relevant, as the code of their algorithm was used and adapted to the context of the present study. Within their study, they developed an algorithm to find the most suitable moment during a day to send a reminder to the users to go running based on the user's current context. They made use of an RL algorithm, which was trained within a simulation on a user model constructed from psychological insights, as well as historical data on user's performance of PA under varying weather conditions. The data was mainly used to estimate the likelihood that a user would want to go running given certain weather. The user would receive a maximum of 14 reminders per week, with the goal to maximise reminder delivery at appropriate moments, in order to increase user's PA. The RL algorithm was compared to other algorithms within the simulation that had either fixed or random times to deliver the reminder to the user. The results showed that the RL algorithm was able to learn to find the most opportune moment to deliver a reminder to the user and it outperformed the comparison algorithms in terms of the total reward, representing the number of times that a user went running after receiving the reminder.

In [69], Wang et al. then evaluated the developed RL algorithm from [68] within a user study. By first learning a generalised strategy for delivering reminders within a simulation with data of users with similar characteristics to the target user group, the lack of initial knowledge about the users, was circumevented, and thus, random actions were minimised.

Within the user evaluation study, the model first collected the participants' data on PA for 3 weeks to improve personalisation and consequently delivered just-in-time reminders for one week. The two main outcome measures consisted of capturing the number of times the users acknowledged (clicked on) a notification, as well as the number of times the given reminder triggered an activity. Performing the target behaviour following a received reminder (and before the next decision time point) also served as a positive reward to the algorithm. The results of the evaluation study showed diverse reactions of the participants to the delivered reminders where some participants acknowledged a lot, if not most of the reminders without engaging in the target behaviour, while in others, PA was triggered only by a few reminders. However, every participant expressed the significance of receiving reminders to commence a running or walking session, and to prevent negligence, as they allow the users to acknowledge their lack of participation in PA [69]. On the other hand, some participants did not consider the number of received reminders sufficient, and thus perceived the timing more negatively, and only about half of the participants considered the reminders themselves motivating enough to engage in PA, with about 22% of the sent reminders triggering a clicking reaction from the users.

3.4 Time- and Content-based RL interventions

Combining both the timing and content for tailoring purposes, in [21], Gönül et al. developed two RL algorithms, one for opportune moment identification and one for content optimisation that were together evaluated in a simulation. Their approach consisted of two main steps, the training phase and the actual experiment [21]. In the training phase, a State Classifier was trained to minimise the number of random actions taken initially or in previously unseen states, a similar goal as in [69, 65]. The two RL models for opportune moment identification and for intervention selection ran in parallel. During the training phase, the opportune-moment-identification model checked for a state transition, i.e. a shift in the immediate circumstances or environment of the individual. The model then used only the selective eligibility traces method, which rewards the most recent actions, in this case, the actions which resulted in the user engaging with the intervention. Additionally, the rewarded action was always made to be one that delivered an intervention, even if originally a deliver-nothing action was taken, so that in future states, an intervention delivery would be favoured at that decision point. On the other hand, the intervention-selection model took into account the habitual context of the persona within the simulation to select an action based on the current state using a greedy policy from the eligible set of interventions (which included a deliver-nothing action). After the training phase, the State Classifier became available for the actual experiment, where the opportune-moment-identification model used this technique for further improvement of the learning process. If a certain intervention type was selected, the opportune-moment-identification model determined whether to deliver the intervention at each decision point based on the contextual state of the user using a greedy policy, and if needed, the State Classifier was used to select an action in an unknown state. Within the simulations, the person's reaction to the intervention and behaviour were recorded. The reward was obtained for the action taken, eligibility traces were updated, and the transition within the environment (based on the taken action) was recorded. The loop iterated for all given time frames, and once the episode (i.e., a simulated day) was over, the policy was updated with the collected data. The adaptive algorithm was evaluated within a simulation, investigating the number of delivered interventions over 100 episodes for each simulated persona with differing characteristics on commitment intensity, habit strength, intervention type preferences and daily activities. The number of delivered interventions was plotted against habit strength, showing that the algorithm was capable of adapting the frequency of the intervention delivery over time, so that personas with the predisposition of developing habits faster received less reminders sooner. Additionally, the algorithm learned individual patterns for each user considering their defined characteristics, identifying the appropriate user states for delivering an intervention that would result in PA engagement. For example, not only was the algorithm able to identify the user's preferences on the intervention type, but the results also showed that when the temporal difference between the intervention delivery and behaviour performance was considered, the majority of the reminders sent to the user were followed by the user's engagement in PA within 15 minutes, indicating that the system learned to identify conditions within the user's context suitable for performing PA.

3.5 Conclusion for the current study

In conclusion, several studies have investigated the possibilities of implementing adaptive algorithms, with approaches that could tailor the timing, the content of the intervention, or both. Personalisation of both timing and content is crucial, as focusing on only one of the components may not be enough. For example, even when an appropriate timing is estimated, the intervention content may not be sufficient to motivate the users to engage in the target behaviour [69]. On the other hand, while adaptive content may increase the frequency of performing the target behaviour and adherence to the intervention over a longer period of time [65], delivering reminders to the users at inappropriate times, e.g. too frequently, when the user is not available or able to engage in PA can increase the burden on the user, invoke frustration and lead to cessation of willingness to further take part in the intervention [69, 65].

While some studies have investigated different implementations of RL-based algorithms within the context of PA, additional insights could be gained from further exploration on how RL can be utilised for optimal intervention tailoring. Especially, how (and if) more complex parameters pertaining to behaviour change theories and determinants of behaviour can be incorporated within RL-based approaches. Similarly, it remains to be seen how such RL models would perform in simulations that take these parameters (e.g., self-efficacy or motivation) into account [21]. Lastly, while it is most often healthy users that are considered as the target group, it should be investigated further how health-related complications may affect the decision-making processes within RL-based adaptive interventions.

Therefore, the goal of the current study was to build upon studies that combined behavioural theories and determinants with more rule-based techniques, and studies with RL-based techniques, by incorporating behavioural theories, as well as health-related factors into a more flexible, RL-based system that can learn from the users and adapt its strategies to individual preferences and needs.

Chapter 4

Methodology

In this chapter, the methodology towards reaching the research objectives of the current study is presented. The main aim of the current study was to develop (RQ$_{s1}$) and evaluate (RQ$_{s2}$) an intervention capable of employing user-based tailoring with the use of an RL-based algorithm that would help answer the main research question (RQ$_{m}$). This intervention was based on the architecture of the ideal system described in Chapter 1.3, scaled down to fit the available scope of the present study.

In order to achieve the research goals, the steps outlined in Figure 4.1 have been followed within the methodology.

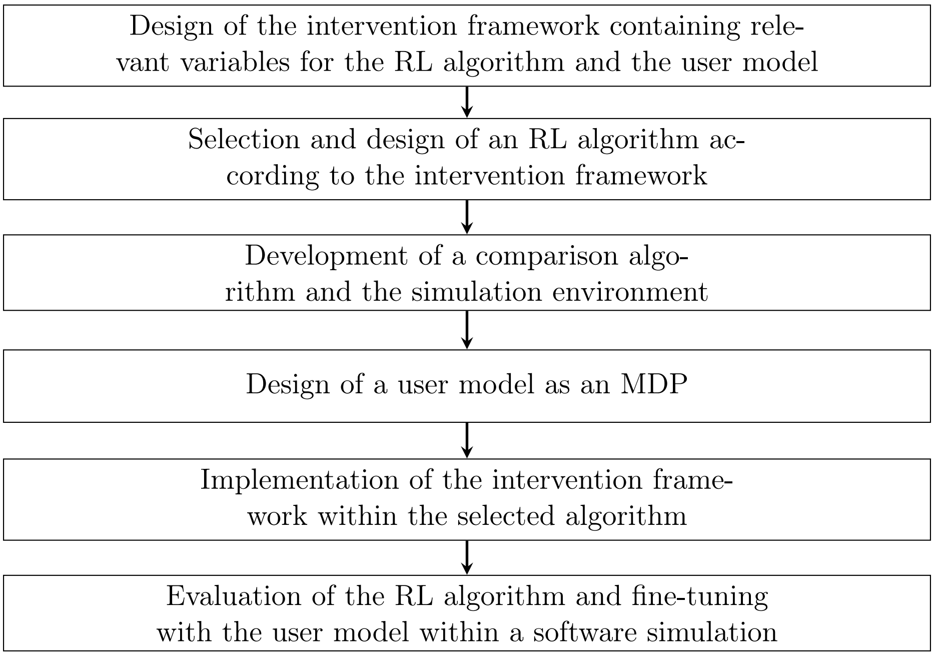

Design of the intervention framework containing relevant variables for the RL algorithm and the user model
Selection and design of an RL algorithm according to the intervention framework
Development of a comparison algorithm and the simulation environment
Design of a user model as an MDP
Implementation of the intervention framework within the selected algorithm
Evaluation of the RL algorithm and fine-tuning with the user model within a software simulation

Figure 4.1: A flowchart outlining the steps conducted within the current study in order to answer the given research questions.

4.1 Intervention framework

First, an intervention framework was defined according to JITAI guidelines and design principles [38, 70], aiming to define the basis for the intervention as a whole, including the adaptations to be made within the intervention based on time and content.

According to [42], a JITAI framework should contain 4 main elements: decision points, intervention options, tailoring variables and decision rules. Therefore, the intervention framework was designed with these components in mind. The sub-components for each element have been selected based on previous literature, as well as from the intervention already designed within the E-Manager project [20].

Decision points

The decision points correspond to times within a day (or other specified time window) when the algorithm should decide whether to send a message (and what kind) to the user. Within the current implementation, the algorithm operates within a software simulation, in which the decision points occur 12 hours per day (from 8AM until 7PM), 7 days a week. At each decision point, the algorithm makes a decision on whether to send a message to the user and if yes, the content selection (RL) algorithm selects the type of message to send. The decision of whether to send the message is based on the user's preferences on the *time of the day* for when it might be a good time to send them a message. A decision rule, therefore, evaluates at each point whether the current hour matches the specified hour.

Intervention options

When it comes to the content of the delivered messages, the interventions options consisted of the content already developed for the E-Manager project, which was used in [20]. These messages were divided into several categories based on what their content was addressing: which behaviour stage and which determinants (self-efficacy or other). Once a message category was selected, a specific message from that category was selected by means of a random selection. This was done, so that the number of possible actions to explore by the algorithm was reduced, allowing for faster, and more accurate learning of which messages would be appropriate for the user and their current context. Additionally, by having the algorithm learn message categories, rather than specific messages, a certain degree of variability in the messages delivered to the user can be ensured, which is necessary to retain salience of the delivered messages [23]. If the algorithm was to learn the most optimal message for a user, this could potentially result in the user often receiving the same message. What is more, this strategy also allows for changes in the available intervention content without requiring the model to update after each change. For example, new messages within a certain category could be added or old messages could be deleted without affecting the algorithm.

Tailoring variables

Tailoring variables dictate how the content and/or the timing of the intervention should be adapted, addressing the needs of each individual user within their current context and any changes occurring therein. As mentioned above, two variables were used for dividing the available set of messages into categories: behaviour stage and behaviour determinants (self-efficacy or other). These two variables have been chosen due to their importance in determining which type of intervention content could best support (and motivate) the user in their current situation, as described in Chapter 2.1. At every behaviour change stage, different determinants are addressed in the message content that could best influence the user to reframe their intentions, provide instructions on how to engage in PA better or more, and how to sustain this behaviour over long-term. In addition, the self-efficacy determinant plays an important role at each stage, and the users can have frequently varying levels of self-efficacy throughout the intervention. Thus, addressing and potentially increasing self-efficacy when necessary is crucial to aid in increasing the likelihood that a user will engage in and maintain PA [11].

Therefore, the two variables of the behaviour stage that the user is currently in, as well as whether a message addressing self-efficacy needs to be chosen are specified to the RL algorithm as the user's context, according to which the algorithm then selects the fitting message category.

Decision rules

According to the JITAI guidelines, decision rules are used for narrowing down the action space for the RL algorithm to consider, in order to limit the possible actions to the ones relevant to the user's context [38, 70].

In the current implementation, the possible action space for the RL algorithm to consider is already reduced by means of only considering certain message categories (according to behaviour stage and self-efficacy). Thus, instead of using decision rules to narrow down the category and letting the algorithm choose the best message, the algorithm chooses the best category and subsequently rules are used to make the final choice of a specific message. The rules filter the message set based on the category chosen by the model, consider only PA goals and then take a random sample from the remaining eligible messages.

Moreover, decision rules are also used to determine whether a message should be sent to the user in the first place. Since in the current implementation, the timing of the message delivery is not learned by an RL algorithm, a preferred timing is set for receiving the messages and a decision rule is used to evaluate whether the current time corresponds to the selected one.

4.2 System Design

Following the definition of the intervention framework, in order to answer RQ$_{s1}$, an RL-based algorithm was built to implement the framework, as such algorithms have shown a potential in learning adaptive, user-based strategies in health interventions, as described in Chapter 3.

The REINFORCE with baseline algorithm was used for the content selection algorithm to choose the most appropriate message category according to the user's context. The code of the algorithm was derived from the open-source code provided by Wang et al. [68], available on GitHub[1]. This code was used as the basis for the REINFORCE with baseline algorithm, as well as the simulation environment within the current implementation.

The original code was adapted to fulfil the main goal of appropriate content selection, rather than focusing on opportune moment identification, the user simulator was changed to the context (and behavioural theories) of the present study, and the network was adapted to fine-tune the performance of the algorithm.

Next to the main algorithm, a comparison algorithm was created that implements no tailoring and only makes use of the specified time preference on when to send a notification and then selects a random message category from the available action space, rather than learning which choice is the most optimal one.

Both algorithms were evaluated within a software simulation described in section 4.4, of which the results were used to provide answers to the main research question, as well as to RQ$_{s1}$ relating to the applicability of RL within tailored interventions, and RQ$_{s2}$, relating

[1] https://github.com/sw1989/RLforPAUL

to the comparison of the two interventions (with and without tailoring) in increasing the PA of users.

The technical implementation details of the code, as well as the links to the Github repository where the code is stored can be found in Appendix A.

4.2.1 Simulation Environment

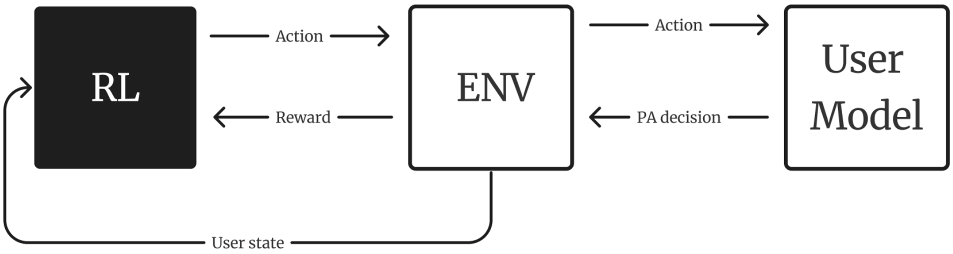


FIGURE 4.2: Diagram of the high level interaction between the RL algorithm, the simulation environment (ENV) and the user model (simulating a human user).

The environment that the RL algorithm interacted with was derived from the implementation of Wang et al. [68], making use of Gym[2], an OpenAI's open-source Python library for developing environments to test RL agents.

Within the environment, the available action space for the RL algorithm was defined; 6 discrete actions represented as {0,...,5}, as well as the user observation space; a compilation of information on the number of notifications left to send, time from last PA, the weekday and hour, and the user's behaviour stage and self-efficacy.

The RL agent would interact with the environment (see Figure 4.2) in a number of iterations, which were broken down into episodes where each episode represented one simulated week. Each episode contained a number of steps, which corresponded to the number of decision points within the week. The number of decision points in each day amounted to 12, representing 12 hours during which a message could be sent to the user. Thus, the total number of decision points, i.e., steps in each episode where the RL agents interacted with the environment equalled to 84.

At the start of each episode, the environment generated a calendar as a matrix of values of all the decision points within the episode that stored information on the user's context (i.e., information from the user observation space) at each hour (decision point), whether the user performed PA at a previous decision point, their probability to engage in PA, as well as the total number of times they performed PA, and the total reward obtained by the agent.

At each step of each episode, the information of the corresponding decision point was extracted from the calendar and sent to the agent. Making use of the information on the user's context, the agent would select an action to take, which was sent back to the environment. The calendar information was then, together with the selected action, sent to a human simulator where this information and the selected action were used to calculate the probability of the user engaging in PA. The decision of the user (not) performing PA was then sent back to the environment where the information within the calendar was updated based on the interactions with the agent and the human simulator, and a reward was assigned to the agent (based on the relevance of the selected message to the user state).

[2] https://github.com/openai/gym

At the end of the episode, the accumulated raw rewards were then sent to the agent for a policy update. This process was then repeated until all episodes were finished.

4.2.2 User Model

The user model that was connected to the environment (as shown in Figure 4.2) and simulated a human was based off of an underlying MDP where the user is in a certain state, which constitutes the user's intention to exercise, their behaviour stage, self-efficacy and memory of past PA. Subsequently, they receive a message (action) from the (RL) algorithm, and based on the message, as well as other behavioural factors decide to perform PA or not.

This model was then extended with other variables that would affect the user's state, and consequently their decision to exercise, outside the delivered message. The values within the user model were inspired by psychological insights from previous literature and the code of Wang et al. [68], however, the values are mostly an estimation.

The model consists of several variables:

– I: user's intention to exercise

– M: the message being sent to the user

– SE: User's self-efficacy

– BS: the user's behaviour stage

– PA: the user's physical activity

and is visualised in Figure 4.3, depicting the relationships between the different variables.

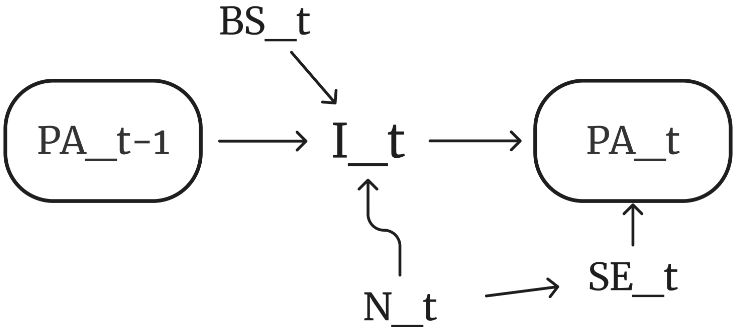


Figure 4.3: A sketch of the user model, visualising the variables that influence the user's final decision on whether to engage in PA.

The human simulator is run at every decision point within the simulator, regardless of whether a message is sent to the user, therefore, the user can decide to engage in PA even without receiving a message.

The baseline value for user's intention to exercise is generated at the start of each episode, as well as each simulation day, (accounting for the night break that can have a positive influence on the user's intent after being rested) and it is linearly increased by a small increment each hour that the user does not exercise [68]. Afterwards, the intent value is adjusted according to the behaviour stage that the user is in, with users in the e.g., initiation stage being much less likely to engage in PA on their own accord in contrast to users in the last (maintenance) stage [10].

The message itself has three roles in influencing the likelihood that a user engages in PA: it must be relevant to the user to increase likelihood of exercise, it reminds the user to do the exercise, and it can influence self-efficacy (SE) with messages that are targeting it. The relevance and reminding factors of the message then have a combined effect on the user's intention to engage in PA, with users being more likely to act according to the message if it is relevant to them and when they remember to do the behaviour [20, 18]. The user's intention at time t is additionally influenced by their past activity, PA_{t-1}, which represents the cumulative memory of the recency of their last PA, as it might be less likely they want to exercise if they did so recently. If that is the case, the sent message is then a feedback to the user, rather than a motivation to engage in PA. The user's intent is then set to 0.001 for the next 12 hours to simulate a recovery time.

Together, the intention with the SE influence the likelihood of the user exercising at a time t and once their values have been adjusted, the final probability for engaging in PA was calculated. The relationships between the variables and the computation of the final probability to engage in PA are concertised into the corresponding transition probability Equations 4.1 to 4.3.

$$P(I_t|PA_{t-1}, M_t) = \begin{cases} I_t, I_t + c & \text{if} \quad PA_{t-1} = 0; \\ I_t, 0 & \text{if} \quad PA_{t-1} = 1; \\ I_t, I_t + c' & \text{if} \quad M_t = 1; \\ I_t, I_t & \text{if} \quad M_t = 0. \end{cases} \tag{4.1}$$

where the intention value I at a time t is given by previous physical activity PA_{t-1} and the received message M_t. I_t is increased linearly by a constant c, if there is no recent PA (denoted as 0), and by c' if the received message is relevant (denoted as 1). Otherwise, I_t remains the same, or becomes 0.

$$P(SE_t|M_t) = \begin{cases} SE_t, SE_t + c'' & \text{if} \quad M_t = 1; \\ SE_t, SE_t & \text{if} \quad M_t = 0. \end{cases} \tag{4.2}$$

where the self-efficacy value SE at a time t is given by the received message M_t. The value of SE_t is increased by a constant c'' if the received message is relevant (denoted as 1). Otherwise, the SE_t value remains the same.

$$P(PA_t|I_t, SE_t) = I_t \cdot SE_t. \tag{4.3}$$

where the final probability of physical activity at a time t is given by the values obtained for I_t and SE_t.

Using the cases defined in Equations 4.1 to 4.3, the user model was implemented within the human simulator as follows:

- Intention
  - The baseline intention value (I), initialised at the beginning of each episode, was generated as a random value between 0.12 and 0.8.

  - The generated value was then adjusted per behaviour stage, i.e., it was multiplied by a value given according to the average number of weekly steps in each cluster determined within the RE-SAMPLE project, averaged by the number of recommended daily steps for COPD users (5000 according to the guidelines of KNGF [71]).
  - During each simulated day, the intent increases by the intent constant (c) at each hour that the user does not engage in PA. The constant is based on [68] and calculated as 1/20 (0.05), assuming it takes about 20 hours to recover from PA.
  - At the first hour of the simulated day, i.e. after a night break, the constant defined in the point above is added to intent in a cumulative fashion that considers the hours of the night. Originally, in [68], the constant value was multiplied by 12 (to account for 12 hours overnight), however, this resulted in too high intent values in the first hours of the morning, therefore, the multiplier was changed from 12 to 6 to create more realistic outcomes.
  - If the user has already exercised during the day, the intention value was set to 0.001, to keep the probability of another exercise during the same day low.
  - For the notification effect, if the delivered message was relevant, the intention value was increased by a constant (c') of 0.65, representing the positive combined value of relevance and reminding factors.

- Self-efficacy (SE)
  - The baseline self-efficacy value was (like intention) generated randomly, however, it was generated each simulation day, to take into account the variable nature of self-efficacy and the barrier that the presence of symptoms may present on some days. The SE value was generated as a random value in the range of [0.6 - 1] if it was not necessary to address and in the range of [0.1 - 0.5] if it was.
  - Similarly to the intention value, SE was adjusted according to the behaviour change stages and increased by 0.2 and 0.3 in the later stages. This aims to represent the fact that users in later stages have had more experience with successfully performing the target behaviour and are, therefore, more likely to believe in their capacity to do so.
  - When the delivered message was relevant to SE (targeting it if necessary), SE was increased by a constant (c'') of 0.3. Additionally, SE was also increased by 0.05 when feedback message was given, accounting for the effect of positive feedback on SE [18]). This value after the feedback was smaller, mainly due to the fact that the user has already engaged in PA that day.
  - If the message was irrelevant to the user state, there was no effect on intent or self-efficacy.

- For the final probability of the user engaging in PA, the intent and self-efficacy values were multiplied and the resulting probability was used in a random decision on whether to exercise.

4.2.3 Content Selection Algorithm: Summary of approach

The intervention framework defined in section 4.1 was then implemented in the code in steps according to the pipeline in Figure 4.4. The first step in implementing the intervention framework was to train the content selection algorithm to, when prompted, select the message category corresponding to the user state observation sent from the (simulation) environment.

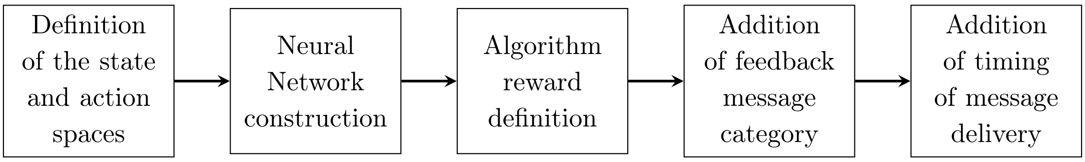

Definition of the state and action spaces
Neural Network construction
Algorithm reward definition
Addition of feedback message category
Addition of timing of message delivery

FIGURE 4.4: Pipeline of development of the content selection algorithm.

In order to speed up the algorithm's learning, there was no element of timing in the message delivery, thus, the algorithm was simply learning the appropriate message category at each given user state, i.e., the observation-action mappings to find an optimal policy.

Once the (near) optimal policy has been found, it was saved, so it could be reloaded later to transfer the algorithm's learning. Next, the element of timing the message delivery was incorporated into the system, so that the algorithm would be asked to select the message category for the user only at certain times, rather than at every step in the simulation.

Lastly, the learned model (with time restriction) was used in combination with the user model defined in Section 4.2.2. The following sections describe the process of developing the content selection algorithm in more detail.

4.2.4 Content Selection (RL) Algorithm

The main goal of the content selection algorithm was to learn an optimal mapping between a given user state and the available actions (with the aim to maximise the received reward/returns).

The action space was defined in the environment as a discrete space of six values, from zero to five {0,...,5}. This is because the action space was comprised of 6 types of message categories. All message categories were stored within a separate *.json* file where each message category had an id [0-5] and a message descriptor of two values, which defined its category. The message descriptors were combinations of the three possible behaviour stages [1,2,3] and whether self-efficacy should be addressed [0,1].

Example: {"ID": 0, "message descriptor": [1,0]}, which refers to the first message category for the first behaviour stage (initiation) and other determinants than self-efficacy to be addressed.

At each step, the agent received a user observation, to which an action should be mapped. The user observations were generated within the environment at the start of each new episode and sent to the agent at each decision point. However, although the full user state/context consisted of the weekday, hour, time from last PA, behaviour stage and self-efficacy at the corresponding decision point, only the behaviour stage and self-efficacy variables were accessed by the agent, since this was the only relevant information for the agent to choose the right type of message category. The other information, such as time from last PA, was mainly used in the human simulator.

In order to learn the mapping between the user observations and the available actions, the algorithm made use of a neural network. The neural network was composed of three fully connected linear layers (see Figure 4.5) where the hidden layers had a ReLU activation function, and soft-max was employed on the last layer, which converts the values output by the network into a probability distribution (that sums up to 1). Thus, at each step, the neural network computes the probability of each action within the action space, given a user observation/state and returns the one with the highest probability (of being relevant to the user state). After each episode is over, the network performs a gradient step for optimising the policy parameter θ. The network made use of the Adam optimiser for calculating the gradient descent in training with a learning rate of 2^{e-3}.

Once the algorithm selects an action for the observed user state, this action is passed to the environment (and consequently the human simulator) where it is evaluated whether the action is relevant to the user state (i.e., whether the behaviour stage and self-efficacy values match with the ones given in the user state). If that is the case, the algorithm receives a reward of 1, otherwise a reward of 0.

Once an episode has finished, the entire list of raw rewards from that episode is sent back to the REINFORCE algorithm for updating its policy. This list is then used to calculate the discounted return, which is a sum of each raw reward with its future rewards, where each future reward is discounted with the parameter γ (the discount rate). Thus, the list of raw rewards is transformed into the discounted return G_t, and its value is calculated as:

$$G_t = R_t + \gamma R_{t+1} + ... + \gamma^n R_{t+n} \tag{4.4}$$

where γ has a value $0 \leq \gamma \leq 1$. In the current implementation, the value of $\gamma = 0.8$.

Once the discounted return has been calculated, a baseline, i.e., the average of of all G_t returns from all previous episodes, is subtracted from the newly calculated return, in order to reduce variance and allow for faster learning.

In order to then update the parameters of the neural network, backpropagation is used to update the weights of the network using gradient descent on negative gradient, which serves as a workaround for performing gradient ascent.

Feedback category

In addition to the model with the 6 total possible actions (message categories) described above, a second network was trained that also included a feedback message type.

In order to also be able to send feedback messages to the user after a PA has recently been performed, the feedback category was included as a possible action in the action space of the algorithm. Since feedback was given to the user after a recent activity, *time from last PA* was included as an input to the neural network and the final action space consisted of seven actions [0-6], see Figure 4.5, which displays the architecture of the used neural network.

However, the feedback category was not included in the reward system of the neural network during training. This was done because, since the feedback message should be delivered when *time from last PA* < 12 (a PA was performed in the last 12 hours), which is what the goal was for the final implementation, this would give too many plausible points for the category of feedback messages (as during training the network was asked to select a message at each time point), so the model would simply learn to deliver the feedback action most of the time, reaching high reward still. Therefore, the feedback category was 'ignored' in the reward system during training; even though it was contained in the action space, it was never evaluated as relevant. In this way, the model learned all the other categories, and then, during the evaluation phase (when the model is not learning anymore as not to influence the weights in the network), the feedback category was injected into the algorithm by forcibly selecting feedback when *time from last PA* < 12.

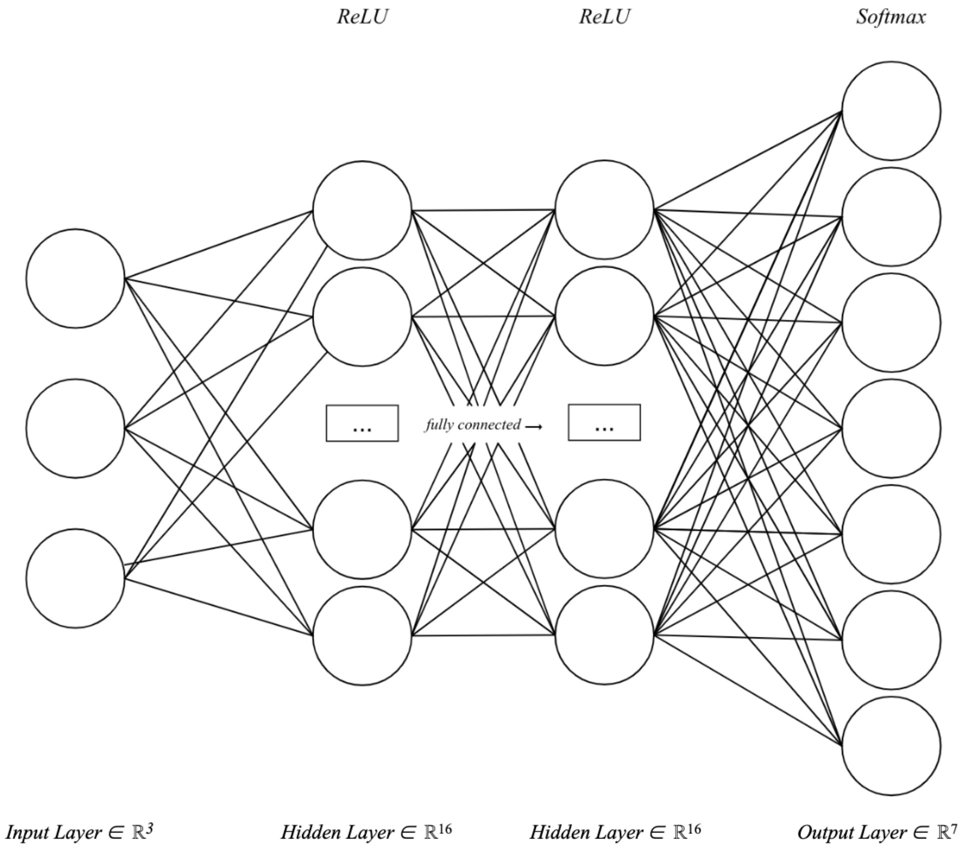


FIGURE 4.5: Diagram of the neural network within the REINFORCE algorithm. The input layer corresponds to the user states used as input for the neural network, corresponding to behaviour stage, self-efficacy and time from last PA. The output layer contains seven nodes, which corresponds to the seven possible actions, and each are given a probability according to the soft-max distribution, based on values computed within the hidden layers.

Timing of the message delivery

Once the content selection algorithm was trained, in order to create a more realistic simulation scenario, the message delivery was restricted to only certain times of the day, which can be selected according to the user's preference.

In the current implementation, a message would be sent at 10AM and at 2PM, as PA levels have been shown to be high during late morning and early afternoon [72].

However, this selection can be changed easily as needed (according to preferences of specific users). Limiting the frequency to two messages per day was done in order to avoid decrease in salience, and thus, effectiveness of the delivered messages in influencing the user to engage in PA, which may be caused by sending the messages too frequently.

During each episode, there are in total twelve possible decision points within the simulation, representing twelve hours, from 8AM to 7PM. At each decision point, the algorithm checks whether the hour corresponds to the preferred hour (so either 10AM or 2PM) and if yes, it prompts the content selection algorithm to select an appropriate message type to send to the user according to the user's current state. Otherwise, no message is sent.

To sum up, Algorithm 1 outlines the final, high-level outline of the content selection algorithm, together with the feedback message category and time restriction.

Algorithm 1 High-level workings of the content selection algorithm.

```
Require: user state
  for each episode do
  generate trajectory of $s, a$ using $\theta$
      for each step in episode do
          if time 10AM or 2PM then                                  ▷ Check time
              if time from last PA $< 12$ then
                  let $a_t$ = feedback
              end if
              $G_t \leftarrow$ sum of discounted returns
              $b \leftarrow$ mean of $G_t$ from past episodes
          end if
          if episode is done then
              $\theta \leftarrow \theta + \alpha(G_t - b)\nabla \ln \pi(a_t|s_t, \theta)$        ▷ Update agent policy
          end if
      end for
  end for                                                   ▷ Return learned policy $\pi$
```

For each episode, an action is matched to the current user state, using the current policy parameter θ within the neural network. If the current time within the simulation corresponds with the selected time that a message should be sent to the user (10AM or 2PM), the algorithm chooses the type of message to send. If PA has been performed recently, a feedback message category is selected. The algorithm then computes the discounted return G_t, based on the reward received from the selected action, as well as the baseline b from all previous returns. Once the episode is over, the baseline is subtracted from the current return, which is used to update the existing policy.

4.3 Overall Simulation process

Considering the components of the system outlined above (environment, user model, RL algorithm), the complete simulation scenario was executed according to the following steps:

- First, a placeholder calendar containing the user context for one episode (simulation week) was created in the simulation environment (see Chapter 4.2.1) and the baseline values of intention and SE for the human simulator were generated and adjusted within the user model, as described in Chapter 4.2.2.

- Starting with the first episode, for each step, the information from the calendar was retrieved and the algorithm checked if the current time corresponded to the time set for sending a message.

- If the current time was outside the hours set for message delivery, the rest of the system continued without action selection from the RL algorithm. Otherwise, the current user context was sent to the content selection (RL) algorithm from the environment.

- Using the received user context, the neural network within the content selection algorithm then output the action with the highest probability. Afterwards, the action was sent back to the environment.

- The selected action and calendar information were consequently sent to the human simulator, so that decision on PA engagement could be made.

- The human simulator calculated the probability of engaging in PA, given the current context and the action received from the algorithm. Using the calculated probability, a random choice was made on the decision to exercise.

- The choice of the human simulator was then sent to the environment.

- The environment updated the calendar information with the algorithm's action, as well as the human simulator's decision, and assigned a reward to algorithm if the chosen message was relevant to the user's context. The reward is then also stored in the environment.

- If there were still more steps left within the episode to go through, the entire process was repeated for each step.

- Once the episode was finished, the accumulated raw rewards were sent to the algorithm, so that they could be used by the algorithm update its policy.

- Once all episodes were done, the process was terminated, otherwise the process was repeated.

4.4 Evaluation of the algorithm

The performance of the RL-based algorithm was evaluated in comparison to the non-tailoring (Random) algorithm, in order to provide answers to RQ$_{s1}$ and RQ$_{s2}$. The comparison was carried out in several different steps.

First, since the content selection (RL) algorithm had a first version that did not include the feedback action category, and a second version that did; the learning curves of both of these versions were compared. This was done in terms of the raw rewards reached by the algorithms during training, in order to evaluate the capacity of each version to reach the maximum raw reward, i.e., the maximum possible number of relevant messages.

Next, the performance of the trained RL-based algorithm with feedback action category was contrasted against the Random algorithm. The performance was averaged over 4 runs (of 100 episodes each) and compared in terms of average raw reward per episode, i.e., how many messages were relevant, out of all the messages sent by each algorithm in each episode.

Lastly, the performance of the trained RL algorithm was also measured in terms of the total PA of the simulated users and the ratio of PA that resulted from receiving a message. Additionally, the behaviour stage that a user (or different users) could be in, varied during the learning of the RL algorithm, so that the content selection could work well for all combinations of behaviour stages and self-efficacy within the user state. Therefore, the performance of both (RL and Random) algorithms was compared across all behaviour stages and this evaluation further served as a judgement of validity of the user model.

For the evaluation of the effects of the RL algorithm on user engagement in PA, three hypotheses were formed to concretise the comparison methodology, as shown below. In each of the experiments, the RL algorithm and/or the Random algorithm were ran 4 times, where each run constituted the number of episodes indicated in the brackets below. The four runs were then averaged to provide the mean and standard deviation of the obtained values.

- *Hypothesis 1* (H_1): Users will, regardless of behaviour stage, engage in PA more often if they receive messages from the RL-based algorithm, in comparison to the Random algorithm.

  Comparison of the general performance of the two algorithms:

  1. in terms of total user PA (50 episodes)
  2. in terms of message to PA ratio (50 episodes)

  where the user states are not distinguished and involve varying behaviour stages and self-efficacy.

- *Hypothesis 2* (H_2)*:* Users in earlier behaviour stages engage in PA less, even when the messages are chosen by an RL-based algorithm.

  1. Total PA count comparison at each behaviour stage with the RL algorithm only (50 episodes)

- *Hypothesis 3* (H_3): Users engage in PA more in each behaviour stage when the messages are chosen by an RL-based algorithm, in comparison to the Random algorithm, i.e., a random selection.

  1. Comparison of total PA count at each behaviour stage when the RL-based vs. when the Random algorithm is used (50 episodes)
  2. Comparison of message to PA ratio at each behaviour stage when the RL-based vs. when the Random algorithm is used (50 episodes)

For each experiment under each hypothesis, the values were tested for statistically significant differences between the involved conditions, using the independent two-samples t-test. For testing H_1, within each of the two experiments, the RL and Random conditions were compared. For testing H_2, the differences between the values for each behaviour stage under the RL condition were compared. For testing H_3, the values obtained at each behaviour stage were compared between the RL and Random condition.

Chapter 5

Results

This section presents the results obtained according to the evaluation steps defined in Chapter 4.4.

Learning of the algorithm

In order to address RQ$_{s1}$ on how RL can be used in user-based tailoring, the two versions of the RL algorithm (with and without the feedback message category as a possible action) were compared in terms of performance during training to analyse the effect of rewarding the feedback category only after the network has been trained (see Chapter 4.2.4). In this way, the performance of the algorithm with the feedback category as a possible action can be compared during training (Figure 5.1) and during evaluation (Figure 5.2), and evaluated in terms of its capacity to learn the optimal policy.

Figure 5.1 shows the learning curves across 1000 episodes of the content selection algorithm with (green) and without (blue) the feedback category, as represented by averaged rewards over windows of 25 episodes.

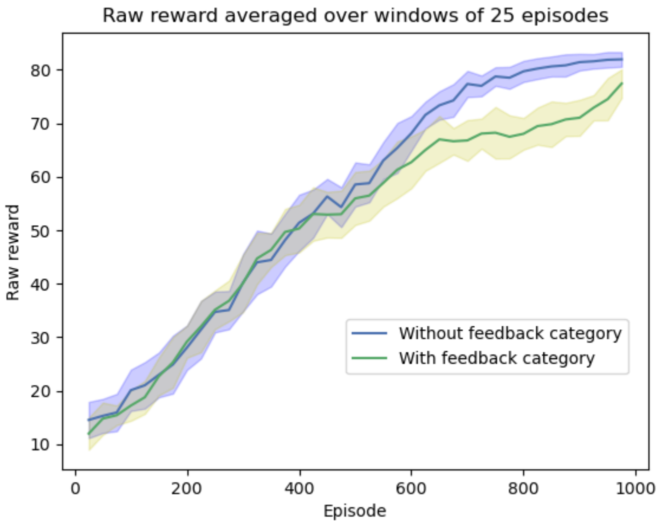


FIGURE 5.1: Comparison of the learning curves of the two versions of the RL-based algorithm, with and without the feedback category as a possible action over 1000 episodes.

The raw reward represents the sum of relevant messages delivered by the algorithm at each given decision point over an episode. The highest reward that the algorithm can reach is 84, meaning the algorithm chose a message relevant to the user context every time.

The algorithm version without the feedback message category was able to reach and stabilise at the maximum reward around episode 700. On the other hand, the algorithm version with the feedback message category contained in the action space was fluctuating close to 80, which was to be expected, since the feedback category itself was never rewarded during the learning process due to its expected prominence, as described in Chapter 4.2.4.

Besides these two algorithm versions, another experiment with the content selection algorithm was conducted where the user state included dayPart (morning/afternoon/evening) and the action state was expanded with combinations thereof, resulting in 18 possible actions to take. The learning, however, was rather slow and staggered at a low reward, and so this version was not used in the final implementation. The visualisation of this version's learning can be found in Appendix B.

In order to evaluate the performance of the content selection algorithm, it was run in an experiment together with an agent without content tailoring where the timing of the message delivery remains the same, but the message is selected randomly from the action space. To ensure a comparable environment (using the same randomisation parameters), the two algorithms were initialised at the same time for each included experiment. To account for varying initialisation parameters, each experiment was run four times and the resulting values were averaged across these runs. Figure 5.2 shows the comparison of the general performance of the two algorithms over 100 episodes where the RL algorithm is set to evaluation state, and thus, is not learning anymore.

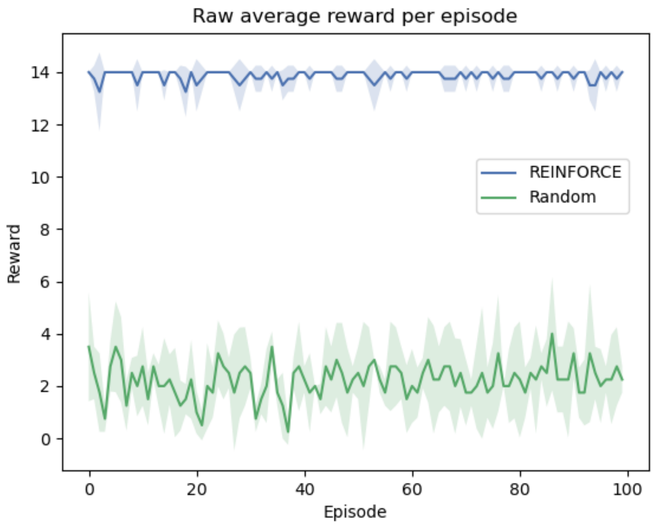


FIGURE 5.2: Comparison of sum of raw rewards per episode obtained by the content selection algorithm with REINFORCE (in blue) vs. a random selection algorithm (in green) across 100 test episodes (post training). The algorithms could obtain a maximum score of 14, corresponding to the maximum number of messages that could be sent during an episode (a week), due to the frequency restriction of two messages per day. The higher the score, the higher the number of relevant selected messages.

The Random algorithm fluctuated around lower rewards, delivering 222 relevant messages out of a total 1400 notifications across 100 episodes, which amounts to 16% accuracy.

On the other hand, the content selection algorithm performed at maximum reward 99% of the time with and average of 1387 relevant messages across four runs of the algorithm. Therefore, the RL-based algorithm was able to outperform the Random algorithm by 83%.

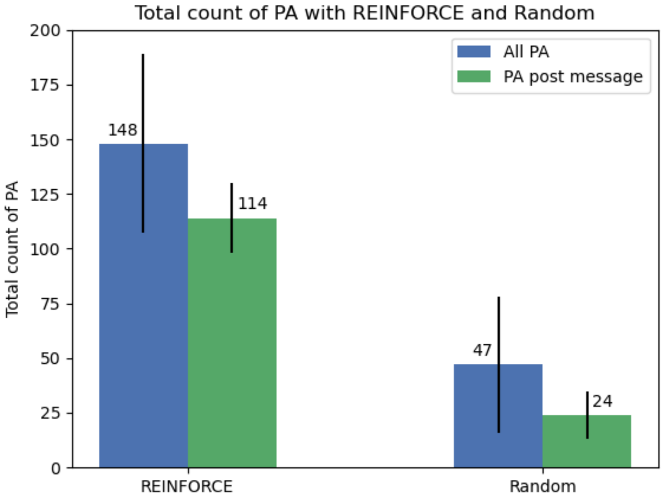


FIGURE 5.3: Comparison of the total PA and the ratio of message to PA (i.e., PA following a message), between the RL-based and the Random algorithm conditions.

PA engagement comparison

To test H_1, the total count of performed PA when RL-based vs. when random messages are sent was compared over 4 runs of 50 episodes each. Figure 5.3 displays this comparison. Next to the total count of PA performed, the number of PA performed after a message was delivered, was also counted.

In total, 700 notifications were sent within the 50 episodes, of which 350 could have resulted in the user performing PA within the simulation, due to the restriction of only exercising every 12 hours. However, the user model was set in a way that the simulated user was not restricted to these intervals of when a message was delivered. This means that the simulated user could have also exercised several hours before or after receiving the message. This is why, next to the total PA count, the *PA post message* measure was defined. *PA post message* refers to PA performed right after (within the same hour) a message has been delivered to the user, which is when the effect of the notification would be at the highest level.

In general, within the content selection algorithm (RL) condition, the simulated user engaged in PA more often, with the total PA count being 29% higher than in the random selection algorithm condition. This difference in total PA between the RL condition (M=148; SD=41) and the Random condition (M=47; SD=31) was statistically significant, as shown by the two-samples t-test ($t(6)$=3.39; p=0.02), supporting H_1.

Additionally, the effect of the message relevance on the simulated user's PA was also higher in the RL condition, with 77% of the total PA engagement following after a message, while this ratio amounted to 51% in the Random condition. This difference in the ratio of message to PA between the RL (M=114; SD=16) and the Random (M=24; SD=11) conditions was also found to be significant ($t(5)$=7.52; p<0.001), providing further support to H_1.

PA per behaviour stage

To test H_2, the content selection algorithm was run in three experiments where the user was in a different behaviour stage each time; in the initiation, action and maintenance phases respectively. The experiments involved four runs of 50 episodes each again that were averaged, and the mean count of PA was analysed at each stage.

Figure 5.4 shows the average number of times the simulated used engaged in PA at each behaviour stage when the content selection algorithm using REINFORCE was employed for selecting messages. Since within the simulation, the user can exercise up to once per day, the maximum possible PA count during one episode equals to 7. Therefore, over 50 episodes, there are 350 possible exercise moments. Considering this number as the maximum threshold, the total count of PA increases by 34% from the initiation phase to the maintenance phase (by 118 on average). In comparison, there is a 2% increase between the first two phases (8) and a 31% increase between the action and maintenance phases (110), signifying a gradual increase towards more consistent PA across the behavioural stages.

The differences between the initiation (M=94; SD=17) and maintenance (M=212; SD=2) stages were found to be statistically significant ($t(3)=-13.61$; $p<0.001$), as well as the differences between the action (M=102; SD=27) and maintenance stages ($t(3)=-8.13$; p=0.003). Together, these results lend support to H_2.

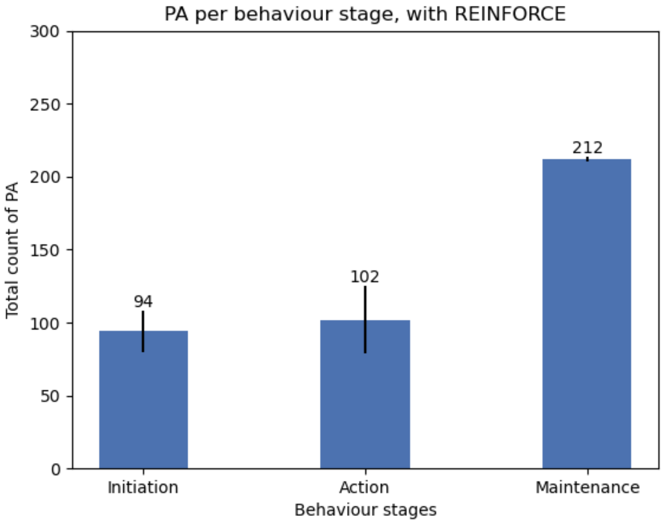


Figure 5.4: Comparison of PA at each behaviour stage when messages are delivered by the content selection (RL) algorithm.

To further analyse the effect of the messages on PA compliance, the count of PA after receiving a message from the RL-based algorithm was also plotted per behaviour stage in Figure 5.5. This plot shows a similar pattern of gradual increase across the behaviour stages also when it comes to the total count of PA performed within the hour that a message was delivered. More interestingly, however, there is a decrease in the total number of times that PA resulted from receiving a message when the later stages are considered, suggesting a decrease in reliance on being prompted by the messages to perform PA. This analysis only considers PA right after a message is sent. A further analysis that looks at PA up to two hours after message can be found in the Appendix B.

The same variables were then also analysed from when the Random algorithm was used

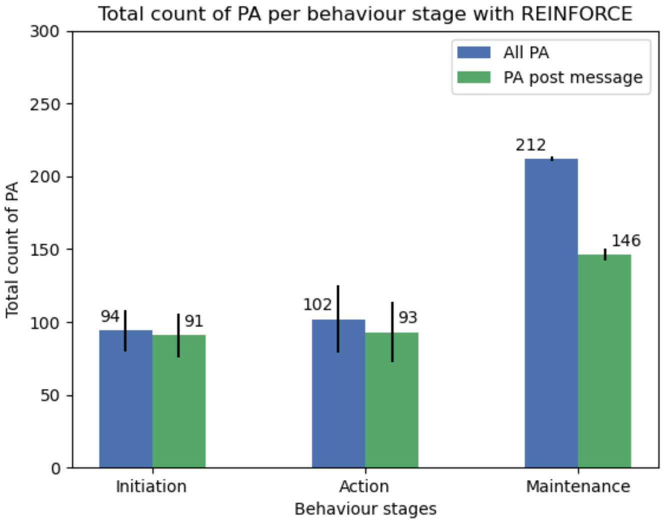


FIGURE 5.5: Comparison of total PA and total PA after receiving a message at each behaviour stage when messages are delivered by the content selection / REINFORCE algorithm

for message selection in order to compare PA counts between the two algorithms and to test H_3 on the effect of the messages chosen by the content selection algorithm on the total count of PA.

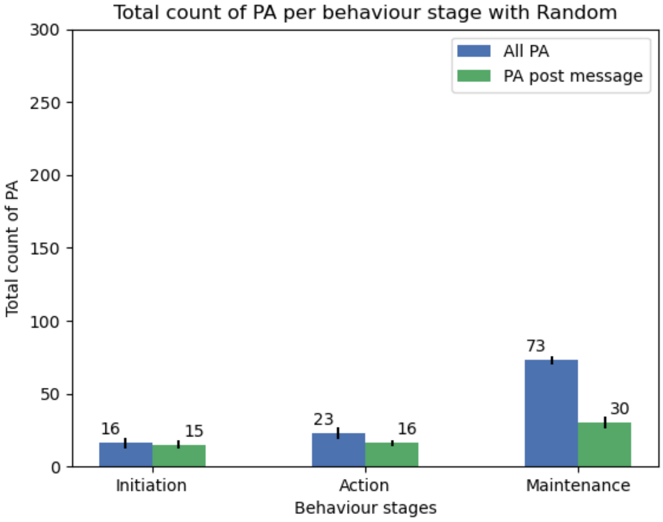


FIGURE 5.6: Count of total PA and total PA after receiving a message at each behaviour stage when the Random algorithm is used for message selection.

Figure 5.6 shows the total count of PA, as well as the count of PA that was performed when a message was delivered under the Random condition. Although a similar pattern as in Figure 5.5 is present, the amount of PA in general is much lower in comparison to when the content selection algorithm with REINFORCE was used.

In the final, maintenance stage, the ratio of PA post message in the Random condition

	RL		Random			
Total PA per stage	*M*	*SD*	*M*	*SD*	*t(df)*	*p*-value
Initiation	94	17	16	4	8.78 (3.39)	0.002
Action	102	27	23	4	5.74 (3.18)	0.008
Maintenance	212	2	73	3	78.51 (5.43)	<0.001
PA post message per stage						
Initiation	91	16	15	3	−8.89 (3.33)	0.002
Action	93	24	16	3	−6.24 (3.09)	0.007
Maintenance	146	4	30	5	−37.93 (5.84)	<0.001

TABLE 5.1: Statistical comparison of difference in means between the RL and Random conditions for each behaviour stage in terms of total PA per behaviour stage, as well as in terms of PA after receiving a message. The difference in each comparison has been found to be statistically significant.

only amounts to 41%, while in the RL condition it reaches 69%, highlighting the greater effect of relevant messages. Testing for statistically significant differences in the mean values of each behaviour stage between the two (RL and Random) conditions, the analysis has found them all to be significant, as shown in Table 5.1, supporting H_3.

Chapter 6

Discussion

The main aim of the present study was to create an adaptive intervention that could be utilised to aid users, especially users with COPD, to increase their physical activity, which is often insufficient despite its numerous benefits on disease management and treatment (e.g., [4, 5, 6, 7, 8]).

The results have shown that, in comparison to an algorithm that selects messages at random, an RL-based system is able to more accurately, and in a higher proportion select messages that are relevant to the current user state. Although not implemented, if a rule-based algorithm's performance was compared to the RL algorithm, it is possible that the performance would be similar, if not better for the rule-based system, as the RL algorithm is still limited in its capabilities. However, this could demonstrate that the RL algorithm was able to infer and learn the decision rules well and such a system is eventually more advantageous, as will be discussed later, in the Section 6.1.

Testing the hypotheses defined in Chapter 4.4, all hypotheses turned out to be true. Starting with H_1, due to the content selection algorithm's ability to frequently choose messages relevant to the current user state, higher PA engagement of the simulated user was observed for this algorithm, in comparison to the Random one. When the message delivered to the user is relevant to/matches their state, the user's intention and self-efficacy to conduct PA are positively affected. Therefore, it was expected that, since the Random algorithm should not be able to match the same amount of messages to the user's state as the content selection algorithm, the PA resulting from receiving a message, and thus, the general PA would be lower in the random condition. The simulation showed this indeed to be true.

Another expectation that was confirmed via testing of H_2, was that the user's engagement in PA would be dependent on the behaviour stage that they are currently in. Previous research has shown that intention and self-efficacy levels are lowest at the beginning stages of behaviour change and highest when the users have reached the stage of maintaining it [73]. The likelihood of engaging in PA at each stage was also adjusted to better fit with the expectations of activity levels in individuals with COPD, and therefore, although the general levels of PA might be lower than they would be for a healthy individual, there is still a clear pattern of increase produced by the simulation across the three stages, with the highest amount of activity in the maintenance stage, which is in accordance with the literature.

The further analysis of how much of this general activity was a product of a relevant message also revealed an interesting pattern. Although the amount of PA engagement after receiving a message still increased at each following stage, its ratio to the total amount of PA decreased. This was an interesting pattern, which suggests a decline in the amount of reliance that the use has on the messages to engage in PA. While in the earlier behaviour stages the user might be less likely to engage in PA by themselves (lower intent), most of the PA is a result of being prompted by a relevant message. On the other hand, in the last, maintenance stage, the user is much more likely to engage in PA on their own, and thus, the amount of PA that is a result of a message prompt is decreased.

This pattern is in accordance with the underlying literature on habit formation, which stipulates that consistently repeating a behaviour, which happens more as the user progresses to the later behaviour stages, leads to a more automatic performance of the behaviour [10], and so users in the later stages would exhibit less reliance on external motivators (i.e., the messages) to engage in PA. Similar results have also been achieved in Gönül et al. [21] where the RL-based intervention delivered less number of reminders as the personas' commitment intensity increased, which can be likened to the personas advancing towards the maintenance stage.

Lastly, when testing H_3, the expectation that the content selection algorithm with REINFORCE would produce higher levels of PA at each behaviour was also confirmed, as within the Random condition, a smaller number of relevant messages was produced. At the last stage in the Random condition, the gap between total PA and the PA after a message is larger, since the user is more likely to exercise on their own and the number of times the exercise results from a message is smaller, producing a smaller ratio in comparison to the RL condition.

All in all, these results have contributed to answering RQ_{s2} relating to the effectiveness of the user-based tailoring in comparison to no tailoring when it comes to increasing the user's engagement in PA.

Even though the RL-based model does not employ all the possibilities that could be contained within user-based tailoring (such as, weather-based preferences or adaptation of the timing of the message according to a learned model), it takes into account relevant variables within the user's current context, to which it adapts the selection of the intervention content. This personalisation then increases the relevance of the intervention to individual users, boosting the effectiveness of the intervention in increasing the user's engagement in PA, which is demonstrated by the obtained results.

The obtained findings further provide suggestions on how RL-based algorithms can be implemented within user-based tailoring interventions for increasing users' engagement, addressing the research question RQ_{s1}.

The main advantage of using an RL-based algorithm within an mHealth intervention is that it can be used to learn to address the user's current needs in terms of both content and timing of the delivery (e.g. [21]). This adaptation can additionally be individualised, rather than remaining group-based, which produces better results in e.g., intervention adherence, as relevance is increased [38]. The main objective of the RL algorithm incorporated within the current intervention design is to select appropriate content/message that is personalised to the user's current state (e.g., their behaviour stage and levels of self-efficacy). The agent, via numerous interactions with a simulated user, learns what constitutes a suitable message in each state and the findings of the simulation demonstrate that it can then frequently deliver relevant intervention content, which increases the likelihood that a user will engage in PA.

What is more, although currently the timing of delivery of the message is fixed to two specified hours of the day, a similar approach of RL application can be utilised here as well. An additional RL-based algorithm can be exploited, so that the selection of an appropriate timeslot can too be automated and adapted to the daily activities of each user.

Thus, an RL-based system can be put together to adapt certain elements of a given intervention to suit each user's needs and improve intervention adherence by delivering relevant content at relevant times.

The present findings cannot be directly compared to the existing literature on research of RL-based algorithms for PA promotion, however, certain parallels can be found.

As in previous work, although a different approach was taken towards implementing RL-based tailoring algorithms, the RL condition outperformed non-optimised/non-tailoring algorithms [64, 66, 67, 68], which is in accordance with the findings of the present study. Therefore, an RL-based algorithm can overall be considered as a potentially useful approach to make use of when designing tailored or just-in-time adaptive interventions.

Although the study of Wang et al. [68], of which the code for the intervention has been made use of, has primarily focused on opportune moment identification, there can be certain comparisons made in terms of the learning of the algorithm. In the implementation of the current study, the RL algorithm is able to reach high rewards, however, it considers a rather small input and a less complex user state. In [68], the user state is considerably more complex, which is also reflected on the extent to which the algorithm is able to learn appropriate actions. Although it was able to optimise the intervention delivery better than the comparison (fixed) agents, the amount of reward that the RL algorithm reached was quite small, which was also observed in the present study when the part of the day was added as a tailoring variable to the content selection algorithm, which significantly increased the action space (see Appendix B). Therefore, improvement to the configuration of the algorithm in its current state might be necessary for it to be able to deal with complex state and/or action spaces.

In the implementation of Gönül et al. [21], the RL algorithm learned to deliver less reminders to the users when commitment intensity towards performing the target behaviour was higher. Although the message frequency in the present study was not restricted, the results demonstrate that in the later behaviour stages (where a higher commitment intensity is assumed), there is lower reliance of the users on the messages to engage in PA. Therefore, similarly to [21], lesser amount of messages was necessary to prompt behaviour in the later stages and the current implementation could be improved by incorporating a frequency restriction to reduce burden on the user.

However, it is important to retain caution when interpreting the obtained results, as the user model employed within the simulation operates under numerous assumptions.

For example, there is always an element of randomness to the simulated user's actions, since the baseline values for the user's intent and self-efficacy are generated at random within a certain range. Additionally, whether the user actually chooses to perform PA or not is also a random choice that makes use of the probability of PA computed within the user model. Therefore, although the probability of engaging in PA could be rather high, the user could still choose not to do it. In a way, this resembles real-world circumstances where a person might have a high desire/intention to exercise but does not actually do so, either due to unsuitable circumstances or other barriers [26]. What is more, the adherence to the intervention as observed in the obtained results is rather high, which may not be fully in line with reality, as adherence to PA-based intervention within the COPD group may vary [9].

Coming back to the generation of values for intent and self-efficacy, although these were inspired by psychological insights from previous literature (see Chapter 4.2.2), it is not entirely possible to estimate fully accurate values that would match a real-world scenario without validating them with data/a user study, especially when it comes to a "non-normative" user group. The same goes for the estimation of the effect size of the relevant messages on the intent and self-efficacy, which might also vary across individual users. Therefore, although the results show a potential resemblance to a real-world scenario, some reservations should be kept about their ecological validity. In the end, rather than the ground truth, the user model used within this study should serve as a basis to, in future work, compare to results obtained from a user study to create a more ecologically valid framework.

Lastly, the obtained findings also contribute to answering the central research question RQ_m, by providing further insight into whether the applied approach could be a suitable method for implementing user-based tailoring within an mHealth application, and whether such a method could be utilised or further expanded within the context of the E-Manager and/or RE-SAMPLE projects. The present findings demonstrate that the content selection algorithm is capable of learning the appropriate message category based on the variables of behaviour stage and self-efficacy. Although that is still a rather limited implementation, it shows the potential of using an RL-based algorithm in an mHealth intervention for increasing PA, and thus, when expanded or improved, it could amount to a rather powerful system. Such an implementation could also be a part of an intervention where face-to-face/telephone and a digital (mHealth) approaches are combined. This combination could provide good results [4], for example by starting with coaching guided by a medical professional, which is then followed up by digital coaching to increase adherence [12]. The digital intervention could then offer a more long-term coaching to provide longer lasting effects on behaviour change [74]. In this way, the user is supported to continue implementing behaviours instructed during the interpersonal coaching and prevent relapse [30].

6.1 Strengths & Limitations

The main strength/advantage of the current approach of using an RL-based algorithm for the intervention content selection as opposed to e.g., rule-based approach lies in the adaptability of RL. Although the system's capabilities as contained in the current implementation might not exceed those of elaborate rule-based systems that might be able to e.g., select messages based on many different variables, the main strength of the current system lies in its potential. While it might take longer for the algorithm to learn what makes an appropriate message (category) and it might require a more sophisticated structure (e.g., a larger neural network) to train with more elaborate user states and large action spaces, it eventually eradicates the need for constructing complex rules and offers higher flexibility in terms of updating the intervention as a whole, since updating the learned policy might be easier than rewriting the defined rules. Most importantly, however, the system can adapt to individual users based on observations and interactions, removing the need to pre-define individual user models or preferences.

On the other hand, one of the limitations of an RL-based approach is the cold-start problem (see Chapter 2.7.1) where, in order to learn the optimal policy, many interactions with the environment are necessary at the beginning. This problem is emphasised even further when learning with more complex user states and larger action spaces, as the time it can take to find and learn the most optimal policy increases. By a rule of thumb, the larger the user state space and/or action space, the more interactions are necessary for the model to learn the appropriate mapping between states and actions. What is more, this learning can also be dependent on the data available to the model, as well as on the way that the environment (that the model is interacting with) is constructed, with both having an influence on the learning ability and speed. However, there exist different techniques to address both of these issues and the length of the training time or the number of interactions is of less importance when utilising a simulation. The policy learned within the simulation could then be used as a starting point for a real-world scenario, circumventing the cold-start problem. In the end, while not perfect, RL-based algorithms still create the potential for a better, smoother user experience and increased intervention adherence.

Another strength of the current system is that it makes use of a user model within the simulation that takes into account factors related to COPD, which might be useful for future implementations within the RE-SAMPLE project in terms of a tailored intervention targeted at this user group. For example, the likelihood of the user engaging in PA at each behaviour stage was derived from activity data of COPD user clusters defined within the RE-SAMPLE project (inactive, active, medium active). Additionally, self-efficacy has an influence to the simulated user's final decision on whether to engage in PA and, although not directly implemented, there is an assumption that self-efficacy is, in the COPD group, related to the presence of symptoms, which may act as a barrier towards PA. However, there remain constraints on the scope of the user model, as well as the values that are used for modelling the influence of the included factors on the final probability of the user engaging in PA.

Since the RL algorithm has only been trained within a simulation, the learning of the algorithm may differ when implemented in a real-world scenario where the states of the environment may be noisy or costly to obtain, while within the simulation, the agent has complete access to the state at each time step [75]. On the other hand, using a simulation for training the RL model also has the advantage that, when employed in real-world, the model does not have to start learning the mappings between the user states and the available actions from scratch, but can rather exploit what it already knows from the simulation/simulated data. Consequently, after interacting with the users, it can then further keep adjusting this knowledge of optimal actions.

6.2 Contributions

The main contribution of the present study is the feasibility demonstration for an intervention targeting PA increase that employs user-based tailoring through the use of RL.

The use of an RL-based algorithm within the intervention design was among the main objectives, as previous work has shown this approach to be effective means for learning to deliver appropriate intervention content and/or to deliver the intervention at appropriate times. Therefore, the goal was to develop an RL-based algorithm that could be combined with the content previously developed for the E-Manager project and potentially further expanded upon in future research. In the end, a working system with RL at the forefront for selecting a fitting intervention content was created, and although limited in its capabilities in its present version, it demonstrates the potential of such an approach to eventually be useful in practice. On top of content tailoring, the system also contains adaptation in terms of timing of the delivery, which can be based off of the user's preferences on when it would be a good time to send a message, which adds to the JITAI nature of the design.

Another contribution is the user model that was created to test together with the RL algorithm. It is based on behavioural theories, which strengthens its validity and it considers different types of users (in varying behaviour stages and with differing self-efficacy). It considers COPD-specific factors and it can be used to create different profiles for this user group to test within simulations for further development of different interventions. In this way, simulation personas can be useful to analyse the potential effects of an intervention on the representation of the user group (and the different types of users therein) before testing with real users. This can aid improving the workings of an intervention prior to a user study and potentially reducing burden on the users. What is more, using the model in a simulation, especially when RL-based algorithms are employed, can help create a basis for the agent to learn appropriate actions, so that the number of random actions, as well as the time it takes to adapt to real users is reduced, helping to overcome the cold-start problem.

6.3 Future Work

There are several ways in which the present work could be improved and/or expanded upon. Firstly, it should be studied further how a more complex user state could be considered as an input to the neural network of the RL algorithm. While further tests would need to be conducted, preliminary experiments show that the algorithm in its current configuration is much slower to find the optimal policy with an extended user state space that significantly increases the action space (see Appendix B). One approach could be to study different configurations of the neural network that the algorithm utilises to learn the probabilities of each action, e.g., in the number of nodes within the hidden layers, a different number of hidden layers, various optimisers or a different neural network overall. However, it should also be considered that an improvement may be reached when a more complex user state is used, while the action space does not increase by a large amount, as this reduces the number of possible actions that the algorithm needs to explore for each state. While many possible ways of adapting and/or extending the algorithm exist, the way to achieve success when using RL is not so straightforward, as the algorithm is composed of many changing (sometimes random) elements that might be difficult to troubleshoot if any errors arise, and may require ample fine-tuning in order to obtain promising outcomes [76].

However, a more considerable alteration could be the usage of a different RL algorithm. For example, there exist algorithms that are successors to the REINFORCE with baseline algorithm that provide improvements to certain elements, such as stability or speed of the learning. One such algorithm is the Proximal Policy Optimisation (PPO) algorithm, a popularly applied algorithm that is more efficient to train in terms of wall-clock time and has a good overall performance on various benchmarks [77].

When it comes to the user model that has been used for the simulation, future work should evaluate this model within a user study to validate the values used for determining, e.g., the effect size of the message on the user's state. What is more, additions can be made to the user model (and consequently the user state) in the form of additional behavioural determinants that may, on one hand add to the model's ability to account for the users' behaviour and be more representative, and on the other hand, extend the number of categories of the messages that the algorithm is choosing from, so that more detailed choices can be made. For example, other factors could be added, such as social support, which, when received, can have a positive effect on self-efficacy, although age can play a role in how much social support is necessary [28]. However, such an extension will need to go hand-in-hand with improving the performance/power of the network or the algorithm, so that an extended user model can be handled.

The intervention as a whole should also be tested within a user study to evaluate its performance and efficacy in a real-world scenario. Ideally, the developed system should be tested within a long-term evaluation study with the target group, as shorter evaluation studies tend to not be long enough for the algorithm to make significant individual-based adaptations [69], or the participants experience increased stimulation towards performing PA due to being involved in the study, which can skew the effects of the intervention within the initial weeks [65].

Chapter 7

Conclusion

In conclusion, an RL-based algorithm was tested in comparison to an algorithm with random selection, which showed that the RL-based algorithm can, with the use of the provided user model, outperform the comparison algorithm without tailoring in terms of efficacy in increasing PA in the simulated users. Overall, the RL-based algorithm was almost 30% more effective than the comparison algorithm, answering RQ$_{s2}$.

In this way the results contributed to answering RQ$_{s1}$, as it has been shown that RL can be used for adapting the selection of the intervention content by learning to choose messages relevant to the current user's state. Although in the current version the timing of the messages is set up manually, a similar RL approach can be tested for opportune moment identification.

With regards to the RQ$_m$, user-based tailoring for interventions that aim to increase PA in the COPD user group can be achieved with the developed content selection algorithm, together with a user model based on behavioural theories that takes the concepts of and the barriers in translating the intention to act into actual action and eventually maintenance of said action into account. Since the used intervention content (i.e., the messages) can be rather easily adapted to suit also other lifestyle goals and/or other chronic diseases [20], it is possible that eventually, with a robust enough model, such user groups or goals could also be incorporated into the RL algorithm and the user model, making for a widely adaptable, user-focused health intervention.

Bibliography

[1] B. K. Pedersen and B. Saltin, “Evidence for prescribing exercise as therapy in chronic disease,” *Scandinavian journal of medicine & science in sports*, vol. 16, no. S1, pp. 3–63, 2006.

[2] R. voor Volksgezondheid en Milieu, “Beweegrichtlijnen,” 2020. Accessed February 28, 2023. https://www.rijksoverheid.nl/documenten/publicaties/2020/07/07/beweegrichtlijnen.

[3] P. Adami, A. Negro, N. Lala, and P. Martelletti, “The role of physical activity in the prevention and treatment of chronic diseases,” *Clin Ter*, vol. 161, no. 6, pp. 537–41, 2010.

[4] L. Nici, C. Donner, E. Wouters, R. Zuwallack, N. Ambrosino, J. Bourbeau, M. Carone, B. Celli, M. Engelen, B. Fahy, *et al.*, “American thoracic society/european respiratory society statement on pulmonary rehabilitation,” *American journal of respiratory and critical care medicine*, vol. 173, no. 12, pp. 1390–1413, 2006.

[5] C. F. Emery, N. E. Leatherman, E. J. Burker, and N. R. MacIntyre, “Psychological outcomes of a pulmonary rehabilitation program,” *Chest*, vol. 100, no. 3, pp. 613–617, 1991.

[6] C. Egan, B. M. Deering, C. Blake, B. M. Fullen, N. M. McCormack, M. A. Spruit, and R. W. Costello, “Short term and long term effects of pulmonary rehabilitation on physical activity in copd,” *Respiratory medicine*, vol. 106, no. 12, pp. 1671–1679, 2012.

[7] M. A. Spruit, C. Burtin, P. De Boever, D. Langer, I. Vogiatzis, E. F. Wouters, and F. M. Franssen, “Copd and exercise: does it make a difference?,” *Breathe*, vol. 12, no. 2, pp. e38–e49, 2016.

[8] R. V. Milani, J. Myers, and A. L. Ries, “The impact of exercise reconditioning on breathlessness in severe chronic airflow limitation,” *Journal of Cardiopulmonary Rehabilitation and Prevention*, vol. 16, no. 4, p. 260, 1996.

[9] C. F. Emery, R. L. Schein, E. R. Hauck, and N. R. MacIntyre, “Psychological and cognitive outcomes of a randomized trial of exercise among patients with chronic obstructive pulmonary disease.,” *Health Psychology*, vol. 17, no. 3, p. 232, 1998.

[10] P. Lally, C. H. Van Jaarsveld, H. W. Potts, and J. Wardle, “How are habits formed: Modelling habit formation in the real world,” *European journal of social psychology*, vol. 40, no. 6, pp. 998–1009, 2010.

[11] C. J. Armitage, “Can the theory of planned behavior predict the maintenance of physical activity?,” *Health psychology*, vol. 24, no. 3, p. 235, 2005.

[12] K. Kivelä, S. Elo, H. Kyngäs, and M. Kääriäinen, "The effects of health coaching on adult patients with chronic diseases: a systematic review," *Patient education and counseling*, vol. 97, no. 2, pp. 147–157, 2014.

[13] P. Krebs, J. O. Prochaska, and J. S. Rossi, "A meta-analysis of computer-tailored interventions for health behavior change," *Preventive medicine*, vol. 51, no. 3-4, pp. 214–221, 2010.

[14] R. H. Bonczek, C. W. Holsapple, and A. B. Whinston, *Foundations of decision support systems*. Academic Press, 2014.

[15] M. B. Sesen, A. E. Nicholson, R. Banares-Alcantara, T. Kadir, and M. Brady, "Bayesian networks for clinical decision support in lung cancer care," *PloS one*, vol. 8, no. 12, p. e82349, 2013.

[16] ZonMw, "E-manager chronic diseases," 2018. Accessed March 4, 2023. https://www.zonmw.nl/nl/over-zonmw/e-health-en-ict-in-de-zorg/programmas/project-detail/imdi/e-manager-chronic-diseases/.

[17] "RE-SAMPLE - real-time data monitoring for shared, adaptive, multi-domain and personalised prediction and decision making for long-term pulmonary care ecosystems," 2021-2025. Accessed March 4, 2023. https://www.re-sample.eu/.

[18] S. Mohan, "Exploring the role of common model of cognition in designing adaptive coaching interactions for health behavior change," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, no. 1, pp. 1–30, 2021.

[19] M. Menictas, M. Rabbi, P. Klasnja, and S. Murphy, "Artificial intelligence decision-making in mobile health," *The biochemist*, vol. 41, no. 5, pp. 20–24, 2019.

[20] E. A. Hietbrink, A. Middelweerd, P. van Empelen, K. Preuhs, A. A. Konijnendijk, W. O. Nijeweme-d'Hollosy, L. K. Schrijver, G. D. Laverman, and M. M. Vollenbroek-Hutten, "A digital lifestyle coach (e-supporter 1.0) to support people with type 2 diabetes: Participatory development study," *JMIR Human Factors*, vol. 10, no. 1, p. e40017, 2023.

[21] S. Gönül, T. Namlı, A. Coşar, and İ. H. Toroslu, "A reinforcement learning based algorithm for personalization of digital, just-in-time, adaptive interventions," *Artificial Intelligence in Medicine*, vol. 115, p. 102062, 2021.

[22] S. Muellmann, S. Forberger, T. Möllers, E. Bröring, H. Zeeb, and C. R. Pischke, "Effectiveness of ehealth interventions for the promotion of physical activity in older adults: a systematic review," *Preventive medicine*, vol. 108, pp. 93–110, 2018.

[23] R. A. J. de Vries, "Theory-based and tailor-made: Motivational messages for behavior change technology," 2018.

[24] C. R. Nigg, K. S. Geller, R. W. Motl, C. C. Horwath, K. K. Wertin, and R. K. Dishman, "A research agenda to examine the efficacy and relevance of the transtheoretical model for physical activity behavior," *Psychology of sport and exercise*, vol. 12, no. 1, pp. 7–12, 2011.

[25] R. Schwarzer, "Health action process approach (hapa) as a theoretical framework to understand behavior change," *Actualidades en Psicología*, vol. 30, no. 121, pp. 119–130, 2016.

[26] R. Schwarzer, “Modeling health behavior change: How to predict and modify the adoption and maintenance of health behaviors,” *Applied psychology*, vol. 57, no. 1, pp. 1–29, 2008.

[27] R. Schwarzer, S. Lippke, and A. Luszczynska, “Mechanisms of health behavior change in persons with chronic illness or disability: the health action process approach (hapa).,” *Rehabilitation psychology*, vol. 56, no. 3, p. 161, 2011.

[28] T. Bonsaksen, A. Lerdal, and M. S. Fagermoen, “Factors associated with self-efficacy in persons with chronic illness,” *scandinavian Journal of psychology*, vol. 53, no. 4, pp. 333–339, 2012.

[29] S. B. Bentsen, T. Wentzel-Larsen, A. H. Henriksen, B. Rokne, and A. K. Wahl, “Self-efficacy as a predictor of improvement in health status and overall quality of life in pulmonary rehabilitation—an exploratory study,” *Patient education and counseling*, vol. 81, no. 1, pp. 5–13, 2010.

[30] R. Arnold, A. V. Ranchor, G. H. Koëter, M. J. de Jongste, J. B. Wempe, N. H. ten Hacken, V. Otten, and R. Sanderman, “Changes in personal control as a predictor of quality of life after pulmonary rehabilitation,” *Patient education and counseling*, vol. 61, no. 1, pp. 99–108, 2006.

[31] P. Pirolli, “A computational cognitive model of self-efficacy and daily adherence in mhealth,” *Translational behavioral medicine*, vol. 6, no. 4, pp. 496–508, 2016.

[32] R. Neff, J. Fry, *et al.*, “Periodic prompts and reminders in health promotion and health behavior interventions: systematic review,” *Journal of medical Internet research*, vol. 11, no. 2, p. e1138, 2009.

[33] M. Fiordelli, N. Diviani, and P. J. Schulz, “Mapping mhealth research: a decade of evolution,” *Journal of medical Internet research*, vol. 15, no. 5, p. e95, 2013.

[34] C. Free, G. Phillips, L. Felix, L. Galli, V. Patel, and P. Edwards, “The effectiveness of m-health technologies for improving health and health services: a systematic review protocol,” *BMC research notes*, vol. 3, no. 1, pp. 1–7, 2010.

[35] M. L. A. Lustria, S. M. Noar, J. Cortese, S. K. Van Stee, R. L. Glueckauf, and J. Lee, “A meta-analysis of web-delivered tailored health behavior change interventions,” *Journal of health communication*, vol. 18, no. 9, pp. 1039–1069, 2013.

[36] G. Valerio, V. Gallarato, O. D’Amico, M. Sticco, P. Tortorelli, E. Zito, R. Nugnes, E. Mozzillo, and A. Franzese, “Perceived difficulty with physical tasks, lifestyle, and physical performance in obese children,” *BioMed research international*, vol. 2014, 2014.

[37] S. C. M. W. Tummers, A. Hommersom, L. Lechner, R. Bemelmans, and C. A. W. Bolman, “Determinants of physical activity behaviour change in (online) interventions, and gender-specific differences: a bayesian network model,” *International Journal of Behavioral Nutrition and Physical Activity*, vol. 19, no. 1, pp. 1–18, 2022.

[38] I. Nahum-Shani, S. N. Smith, B. J. Spring, L. M. Collins, K. Witkiewitz, A. Tewari, and S. 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*, vol. 52, no. 6, pp. 446–462, 2018.

[39] S. Hojjatinia, S. Hojjatinia, C. M. Lagoa, D. Brunke-Reese, and D. E. Conroy, "Person-specific dose-finding for a digital messaging intervention to promote physical activity.," *Health Psychology*, vol. 40, no. 8, p. 502, 2021.

[40] S. P. Goldstein, F. Zhang, J. G. Thomas, M. L. Butryn, J. D. Herbert, and E. M. Forman, "Application of machine learning to predict dietary lapses during weight loss," *Journal of diabetes science and technology*, vol. 12, no. 5, pp. 1045–1052, 2018.

[41] A. Kankanhalli, M. Saxena, and B. Wadhwa, "Combined interventions for physical activity, sleep, and diet using smartphone apps: A scoping literature review," *International journal of medical informatics*, vol. 123, pp. 54–67, 2019.

[42] I. Nahum-Shani, E. B. Hekler, and D. Spruijt-Metz, "Building health behavior models to guide the development of just-in-time adaptive interventions: A pragmatic framework.," *Health psychology*, vol. 34, no. S, p. 1209, 2015.

[43] L. Wang and L. C. Miller, "Just-in-the-moment adaptive interventions (jitai): A meta-analytical review," *Health Communication*, vol. 35, no. 12, pp. 1531–1544, 2020.

[44] W. Hardeman, J. Houghton, K. Lane, A. Jones, and F. Naughton, "A systematic review of just-in-time adaptive interventions (jitais) to promote physical activity," *International Journal of Behavioral Nutrition and Physical Activity*, vol. 16, no. 1, pp. 1–21, 2019.

[45] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[46] M. L. Puterman, "Markov decision processes," *Handbooks in operations research and management science*, vol. 2, pp. 331–434, 1990.

[47] C. Yu, J. Liu, S. Nemati, and G. Yin, "Reinforcement learning in healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–36, 2021.

[48] Y. Li, "Reinforcement learning in practice: Opportunities and challenges," *arXiv preprint arXiv:2202.11296*, 2022.

[49] H. Zhang and T. Yu, "Taxonomy of reinforcement learning algorithms," *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pp. 125–133, 2020.

[50] OpenAI, "Introduction to rl, part 2: Kinds of algorithms," 2018. Accessed May 15, 2023. https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html?highlight=taxonomy#a-taxonomy-of-rl-algorithms.

[51] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[52] D. Silver, "Policy gradient methods," 2015. Retrieved from https://www.youtube.com/watch?v=KHZVXao4qXs.

[53] S. Kapoor, "Policy gradients in a nutshell," 2018. Retrieved from https://towardsdatascience.com/policy-gradients-in-a-nutshell-8b72f9743c5d.

[54] T. Simonini, "Policy gradient with pytorch," 2018. Retrieved from https://huggingface.co/learn/deep-rl-course/unit4/advantages-disadvantages?fw=pt.

[55] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Reinforcement learning*, pp. 5–32, 1992.

[56] C. X. Tsou, “Reinforcement learning: Introduction to policy gradients,” 2021. Retrieved from https://medium.com/nerd-for-tech/reinforcement-learning-introduction-to-policy-gradients-aa2ff134c1b.

[57] N. Botteghi, “Robotics deep reinforcement learning with loose prior knowledge,” *University of Twente*, 2021.

[58] P. Lippe, R. Halm, N. Holla, and L. Meijerink, “Reinforcement learning: Introduction to policy gradients,” 2019. Retrieved from https://medium.com/@fork.tree.ai/understanding-baseline-techniques-for-reinforce-53a1e2279b57.

[59] F. Zhu, J. Guo, Z. Xu, P. Liao, L. Yang, and J. Huang, “Group-driven reinforcement learning for personalized mhealth intervention,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pp. 590–598, Springer, 2018.

[60] S. A. Tabatabaei, M. Hoogendoorn, and A. van Halteren, “Narrowing reinforcement learning: Overcoming the cold start problem for personalized health interventions,” in *PRIMA 2018: Principles and Practice of Multi-Agent Systems: 21st International Conference, Tokyo, Japan, October 29-November 2, 2018, Proceedings 21*, pp. 312–327, Springer, 2018.

[61] M. C. Klein, A. Manzoor, and J. S. Mollee, “Active2gether: A personalized m-health intervention to encourage physical activity,” *Sensors*, vol. 17, no. 6, p. 1436, 2017.

[62] A. Middelweerd, J. Mollee, M. M. Klein, A. Manzoor, J. Brug, S. J. Te Velde, *et al.*, “The use and effects of an app-based physical activity intervention “active2gether” in young adults: quasi-experimental trial,” *JMIR Formative Research*, vol. 4, no. 1, p. e12538, 2020.

[63] J. S. Mollee, A. Middelweerd, S. J. t. Velde, and M. C. Klein, “Evaluation of a personalized coaching system for physical activity: User appreciation and adherence,” in *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pp. 315–324, 2017.

[64] E. M. Forman, S. G. Kerrigan, M. L. Butryn, A. S. Juarascio, S. M. Manasse, S. Ontañón, D. H. Dallal, R. J. Crochiere, and D. Moskow, “Can the artificial intelligence technique of reinforcement learning use continuously-monitored digital data to optimize treatment for weight loss?,” *Journal of behavioral medicine*, vol. 42, pp. 276–290, 2019.

[65] M. Zhou, Y. Mintz, Y. Fukuoka, K. Goldberg, E. Flowers, P. Kaminsky, A. Castillejo, and A. Aswani, “Personalizing mobile fitness apps using reinforcement learning,” in *CEUR workshop proceedings*, vol. 2068, NIH Public Access, 2018.

[66] D. Martinho, J. Carneiro, J. Neves, P. Novais, J. Corchado, and G. Marreiros, “A reinforcement learning approach to improve user achievement of health-related goals,” in *Progress in Artificial Intelligence: 20th EPIA Conference on Artificial Intelligence, EPIA 2021, Virtual Event, September 7–9, 2021, Proceedings 20*, pp. 266–277, Springer, 2021.

[67] P. Liao, K. Greenewald, P. Klasnja, and S. Murphy, “Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–22, 2020.

[68] S. Wang, C. Zhang, B. Kr"ose, and H. van Hoof, “Optimizing adaptive notifications in mobile health interventions systems: Reinforcement learning from a data-driven behavioral simulator,” *Journal of medical systems*, vol. 45, no. 12, pp. 1–8, 2021.

[69] S. Wang, K. Sporrel, H. van Hoof, M. Simons, R. D. de Boer, D. Ettema, N. Nibbeling, M. Deutekom, and B. Kröse, “Reinforcement learning to send reminders at right moments in smartphone exercise application: A feasibility study,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 11, p. 6059, 2021.

[70] S. Gonul, T. Namli, S. Huisman, G. B. Laleci Erturkmen, I. H. Toroslu, and A. Cosar, “An expandable approach for design and personalization of digital, just-in-time adaptive interventions,” *Journal of the American Medical Informatics Association*, vol. 26, no. 3, pp. 198–210, 2019.

[71] K. N. G. voor Fysiotherapie (KNGF), “Kngf-richtlijn copd,” 2020. Retrieved from https://www.kngf.nl/binaries/content/assets/kennisplatform/onbeveiligd/richtlijnen/copd-2020/product-remake/copd-2020-praktijkrichtlijn.pdf.

[72] A. Hecht, S. Ma, J. Porszasz, R. Casaburi, C. C. R. Network, *et al.*, “Methodology for using long-term accelerometry monitoring to describe daily activity patterns in copd,” *COPD: Journal of Chronic Obstructive Pulmonary Disease*, vol. 6, no. 2, pp. 121–129, 2009.

[73] S. Lippke, J. P. Ziegelmann, and R. Schwarzer, “Stage-specific adoption and maintenance of physical activity: Testing a three-stage model,” *Psychology of Sport and Exercise*, vol. 6, no. 5, pp. 585–603, 2005.

[74] C. Engstrom, L.-O. Persson, S. Larsson, and M. Sullivan, “Long-term effects of a pulmonary rehabilitation programme in outpatients with chronic obstructive pulmonary disease: a randomized controlled study,” *Scandinavian journal of rehabilitation medicine*, vol. 31, no. 4, pp. 207–213, 1999.

[75] H. A. Nam, S. Fleming, and E. Brunskill, “Reinforcement learning with state observation costs in action-contingent noiselessly observable markov decision processes,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 15650–15666, 2021.

[76] J. Schulman, O. Klimov, F. Wolski, and A. Dhariwal, Prafulla Radford, “Proximal policy optimization,” 2017. Retrieved from https://openai.com/research/openai-baselines-ppo.

[77] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

Appendix A

Technical Details

The whole system was coded with Pytorch[1] and Python (3.9.13)[2] in an Anaconda environment and is available at a Github repository [3].

The REINFORCE algorithm was coded to run on a GPU (RTX 3060 Laptop GPU), using the NVIDIA CUDA[4] toolkit (12.1). Thus, for replication purposes where GPU is available, Pytorch has to be downloaded with CUDA enabled. However, the code also has the option to run with a CPU configuration. The code is divided into multiple scripts and processes, e.g., loading the environment and the REINFORCE algorithm, updating the environment information and running the human simulator that are triggered by one main script.

The necessary packages to run the code are listed below:

- Pandas (1.5.3)
- Numpy (1.24.3)
- scikit-learn (1.2.0)

[1] https://pytorch.org/

[2] https://python.org/downloads/

[3] https://github.com/SandraStrakova/Thesis

[4] https://developer.nvidia.com/cuda-toolkit

Appendix B

Additional results

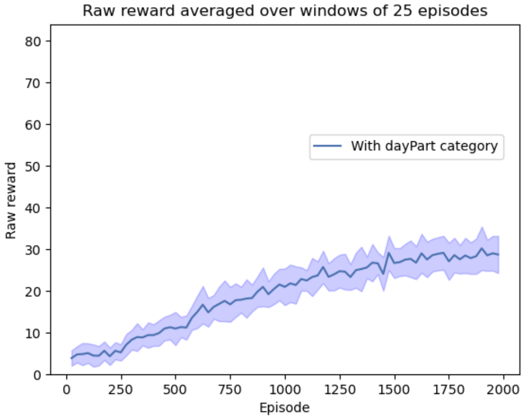


FIGURE B.1: Learning of the algorithm with a "day-part" variable included, meaning the algorithm had to consider 18 actions in total.

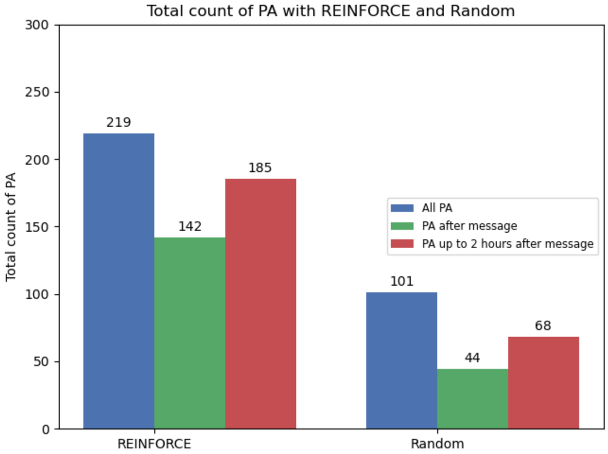


FIGURE B.2: Count of total PA up to 2hrs after receiving a message