The Future of E-Health is Mobile

Grua, Eoin Martino

2021

The Future of E-Health is Mobile

Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

Eoin Martino Grua

MSc.

Department of Computer Science
Faculty of Sciences, Vrije Universiteit Amsterdam

2021

SIKS Dissertation Series No. 2021-25
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

VRIJE UNIVERSITEIT AMSTERDAM

The Future of E-Health is Mobile

Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor of Philosophy
aan de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. C.M. van Praag,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Bètawetenschappen
op vrijdag 3 december 2021 om 9.45 uur
in een bijeenkomst van de universiteit,
De Boelelaan 1105

door

Eoin Martino Grua

geboren te Cork, Ierland

promotoren:	prof.dr. P. Lago prof.dr. A.E. Eiben
copromotoren:	prof.dr. M. Hoogendoorn dr. I. Malavolta
promotiecommissie:	prof.dr. L. Fuentes dr. R. Spalazzese prof.dr. B. Funk prof.dr. M.M. Riper prof.dr. K.V. Hindriks

Contents

List of Tables

List of Figures

Dedicated to my mum Gemma, my late father Claudio, and to my wife Kara.

Acknowledgements

Firstly I would like to thank my mum for her support over the years! I would also like to thank Kara, for all of the support and patience she has given me, especially in the final stretch! I want to thank Mark Hoogendoorn, Gusz Eiben, Ivano Malavolta, and Patricia Lago for their support and supervision during my doctorate years. Doing this kind of interdisciplinary project has been a challenge but thanks to your guidance and patience we got there in the end!

I want to thank all of the colleagues I have met and worked with in the Computational Intelligence and Qualitative Data Analytics groups at the Vrije Universiteit Amsterdam. We had some amazing times together sharing meals, watching movies and dancing! Thank you Julien, Milan, Jakub, Daan, Gongjin, Karine, Masoume, Diederik, Luca, Emile, Bart, Frank, Ward, Jacqueline, Vincent, Lucas, Eliseo, Matteo, Tarik, Fuda, Jan, Floris, David, Luis, Alessandro, and Ali!

I, of course, want to also thank all of my colleagues from the S2 Group at the Vrije Universiteit Amsterdam. I will always remember our talks in front of the coffee machine, mensa, and out eating a pizza! Thank you Fahimeh, Robert, Paolo, Giuseppe, Grace, Nelly, Ilias, Emitz, Jaap, Antony, Michel, Kousar, Remco, Fadime, Razieh, Anjana, Tanjina, Robin, Lara, and Roberto!

Summary

With the current digitisation of our world, we have witnessed a surge in the presence and use of mobile devices. Consequently, there has been a natural increase in the use of mobile applications (apps). A category of app that has been growing in popularity is e-Health apps. However, even though popular, e-Health apps have many shortcomings that need to be addressed. Most notably, the rules and mechanisms employed by current day e-Health apps do not use the full potential of context and features they have access to. Leading to apps that are too rigid and not well tailored to the users' needs and goals.

In this thesis we look at overcoming this rigidity and sub-optimal tailoring of e-Health apps. To reach this goal, we propose combining AI-based personalisation and software self-adaptation. For personalisation, we choose to use reinforcement learning (RL) as it is a good fit in providing personalisation in the e-Health domain. We explore the current state of the art by conducting a systematic literature review. With this review we identify two main weaknesses of RL: it requires a lot of data to reach an optimal policy and exploration can lead to user disengagement. To tackle the former, we propose cluster-based RL. We then further improve our proposed solution by developing an online clustering algorithm designed for e-Health. For the latter, we explore how machine learning can be used to predict user engagement. To better understand software self-adaptation in the domain of apps, we conduct a systematic literature review. In the review, we classify the current approaches and identify several shortcomings relevant to e-Health apps. Lastly, to tackle the identified shortcomings and combine personalisation and self-adaptation, we introduce

a reference architecture for personalised and self-adaptive e-Health apps. We explore the benefits that said architecture can have on social sustainability and empirically evaluate an app implemented following this architecture. For the empirical evaluation two experiments were performed: a user study and a measurement-based experiment. With the user study, we better understand the effects of the implemented app on the end users' perception and usability. With the measurement-based experiment, we investigate the effects that the app has on performance and energy consumption. Our results are promising, as the user study shows improved end users' usability and no significant drawback in end users' perception as well as no perceivable increase in energy consumption or decrease in performance.

Samenvatting

Met de huidige digitalisering van onze wereld zijn we getuige geweest van een toename van de aanwezigheid en het gebruik van mobiele apparaten. Hierdoor is er een natuurlijke stijging van het gebruik van mobiele applicaties (apps). Een categorie apps die steeds populairder wordt, zijn e-Health-apps. Hoewel populaire e-Health-apps veel tekortkomingen hebben die moeten worden aangepakt. Het meest opvallende is dat de regels en mechanismen die worden gebruikt door hedendaagse e-gezondheidsapps niet het volledige potentieel van context en functies gebruiken waartoe ze toegang hebben. Dit leidt tot apps die te rigide zijn en niet goed zijn afgestemd op de behoeften en doelen van de gebruikers.

In dit proefschrift kijken we naar manieren om deze rigiditeit te overwinnen en suboptimale afstemming van e-Health-apps te verebeteren. Om dit doel te bereiken, stellen we voor om de combinatie AI-gebaseerde personalisatie en software-zelfaanpassing te gebruiken.

Voor personalisatie kiezen we voor Reinforcement Learning (RL) omdat dit goed past bij het bieden van personalisatie in het e-Health domein. We verkennen de huidige stand van deze techniek door een systematische literatuurstudie uit te voeren. Met deze studie identificeren we twee belangrijke zwakke punten van RL: het vereist veel gegevens om tot een optimaal beleid te komen en verkenning kan leiden tot terugtrekking van gebruikers. Om het eerste punt aan te pakken, stellen we cluster-gebaseerde RL voor. Vervolgens verbeteren we onze voorgestelde oplossing door een online clusteringalgoritme te ontwikkelen dat is ontworpen voor e-Health. Voor dat laatste onderzoeken we hoe Machine Learning kan worden gebruikt om gebruikersbetrokkenheid te voorspellen. Om de zelfaanpassing van software in het domein van apps beter te begrijpen, voeren we een systematisch literatuuronderzoek uit. In de review classificeren we de huidige benaderingen en identificeren we een aantal tekortkomingen die relevant zijn voor e-Health apps. Om de geïdentificeerde tekortkomingen aan te pakken en personalisatie en zelfaanpassing te combineren, introduceren we ten slotte een referentiearchitectuur voor gepersonaliseerde en zelf-adaptieve e-Health-apps. We onderzoeken de voordelen die deze architectuur kan hebben op sociale duurzaamheid en evalueren empirisch een app die is geïmplementeerd volgens deze architectuur. Voor de empirische evaluatie zijn twee experimenten uitgevoerd: een gebruikersonderzoek en een meetexperiment. Met het gebruikersonderzoek begrijpen we beter wat de effecten zijn van de geïmplementeerde app op de beleving en bruikbaarheid van de eindgebruikers. Met het meetexperiment onderzoeken we welke effecten de app heeft op prestaties en energieverbruik. Onze resultaten zijn veelbelovend, aangezien het in het gebruikersonderzoek een verbeterde bruikbaarheid van eindgebruikers laat zien en geen significant nadeel in de perceptie van eindgebruikers, evenals geen waarneembare toename van het energieverbruik of afname van de prestaties.

1 Introduction

Preamble. *We live in a world with more data and technology than ever. Most of us these days, have an incredibly compact and powerful computer in their pockets in the form of a smartphone. By being almost always on our person, it has the possibility of collecting very precise data about our behaviours and context. This kind of information, when used in the best interest of the user, can have greatly positive consequences. E-Health mobile applications, have that potential. If we can use the full extent of the collected data to personalise and adapt e-Health mobile applications and their interventions, we could help improve the well being of any person who owns a mobile device. This thesis aims at contributing to the creation of personalised and adaptive e-Health mobile applications, for a better future.*

1.1 Motivation

With the digitisation of our world the use of mobile devices is becoming ubiquitous. Consequently comes an increase of collected and stored electronic data. No exception to this trend is the Health domain, where the use of mobile devices and software systems is becoming widespread. This new form of electronic Healthcare is called e-Health. Silber et al. define e-Health as "... the application of information and communications technologies (ICT) across the whole range of functions that affect health" [320].

A thriving area of e-Health is the domain of e-Health mobile applications, otherwise referred to as e-Health apps. In the past years the expansion of e-Health apps has been increasing with a projected market growth of US$102.3Billion by 2023 [134]. E-Health apps offer a wide range of medical services for their users, *e.g.,* lifestyle improvement, fitness, and mental health [269]. E-Health apps differ from other e-Health systems by being able to collect user data with the onboard sensors present on the smartphone on which the app is installed on. Furthermore, e-Health apps have a wide potential user base with a low infrastructure investment as many potential users already own a smartphone or another form of smart mobile device. Lastly, in contrast to other e-Health systems, apps are able to leverage all of the intrinsic characteristics of a mobile medium (such as, being always on, always carried by the user, and personal) to provide timely and in-context services [119]. Although e-Health apps have the potential to use context and their other inherent features to the user's advantage, the rules and mechanisms employed by current day e-Health apps are too rigid and not fully tailored to the individual. In this context, we propose **personalisation** together with software **self-adaptation** as effective tools to better the level of **engagement** and **tailoring** that e-Health apps can offer to the user.

We start by explaining the work done in the field of **personalisation**. For this thesis we built on the definition given by Fan and Poole and define **personalisation** as: "a process that changes a system to increase its personal relevance to an individual or a category of individuals". Researchers working in e-Health have been developing ways to help health workers deliver personalised health care, as it is shown to improve the effectiveness of the health interventions given [11; 29; 153; 253; 371; 294; 157]. When working in e-Health, especially with e-Health apps, reinforcement learning (RL) has been shown to be a good fit [160]. This is because RL is the algorithm of choice for solving sequential decision-making problems. To be able to improve a user's well-being, a e-Health app has to periodically give health interventions with the goal of changing the user's behaviour. The nature of this scenario makes it a sequential decision-making process as many interventions are sequentially given to a user over time. This and the difficulty that comes with observing the outcomes of the interventions only later in the future, play perfectly to RL's strengths. However, RL does have two major downsides: 1) it takes a lot of data to reach a good policy and 2) exploring user undesired or ineffective interventions can lead to their **disengagement**.

A possible solution to the first difficulty can be the use of clustering. By using clustering, the data of like minded people is grouped together. This provides us with a larger dataset from which we can create better **personalisation** strategies, effective for the entire cluster. However, a large amount of these clustering techniques are done in an off-line fashion (*i.e.,* the clusters are done on a fixed dataset and once created they can not be changed in real time). Whilst there are some clustering algorithms created for online clustering, they have limitations that make them a bad fit for the health domain.

To help tackle the second shortcoming of RL, *i.e.,* **disengagement**, machine learning can be used to predict the engagement of a user and when they are going to disengage from an app. There is, however, still work that must be done in understanding which models are most effective and which set of features have the highest predictive value.

Whilst **personalisation** will help in tackling the poor tailoring techniques in e-Health apps, it cannot tackle their current rigidity and minimal use of users' context. In order to adapt to the context of the user as they are utilising the app, developers can apply various software strategies. A common and well researched technique is the use of software **self-adaptation** [277]. **Self-**

adaptation techniques allow the software system to "...modify their behavior at run-time due to changes in the system, its requirements, or the environment in which it is deployed." [9]. This type of adaptation can allow apps to maintain or achieve certain set quality goals, such as adapting to minimise energy consumption or improving performance when required. There is, however, room for improvement in this field, particularly for achieving goals pertinent to the e-Health domain. Only a minimal amount of the current state of the art performs self-adaptation with the aim of achieving non-technical goals, like promoting user behavioural change and lifestyle improvements. Furthermore, whilst self-adaptation is ideal for adapting to the user context, only a minimal number of the current state of the art adapt due to changes detected from third-party apps and smart-objects (*e.g.,* a user's smart watch). The latter, in particular, can capture important information about the user context and has become more prevalent in sub-domains of e-Health, such as fitness.

Whilst some of the techniques described, both in AI and software engineering, have been separately used within the domain of e-Health, little work has been done to combine them together. Their combination, under one architecture designed for e-Health apps, could lead to the solving of many of the challenges and shortcomings identified within this field.

1.2 Thesis Research Questions

The goal of this thesis is to understand how AI-based personalisation and self-adaptation can be used together for designing and developing e-Health mobile apps. We propose RL and clustering techniques to improve the data efficiency and efficacy of e-Health interventions and machine learning models to predict user engagement. We also propose to combine these AI techniques with state of the art software self-adaptation methodologies to explore how e-Health apps can dynamically adapt to the user and their context. For the purpose of this thesis we define dynamic adaptation as the result of the combination of AI-based personalisation and software self-adaptation. As a result, we formulate the following thesis research questions noted as T.RQs, so to not be confused with chapter research questions (identified simply as RQs):

***T.RQ*1** How can RL-based personalisation for e-Health be improved?

***T.RQ*2** How can online-clustering be used to efficiently and effectively cluster e-Health data?

***T.RQ*3** How can we predict user engagement in apps?

***T.RQ*4** How can AI-based personalisation and self-adaptation be used to create e-Health apps that dynamically adapt to the user and their context?

***T.RQ*5** How do dynamically adaptive e-Health apps affect users and their mobile devices?

1.3 Scope

This thesis was created by collecting all of the papers written over the four years of my doctorate degree. Each paper is presented in a separate chapter. The included papers might have had minor changes done to them to better suit the layout of this document (*e.g.,* table or figure resizing). We also define three main parts that collectively create the scope of this thesis, these are: 1) reinforcement learning and machine learning for personalisation and engagement in e-Health,

2) self-adaptation in mobile applications, and 3) creating self-adaptive and personalised e-Health mobile applications. In this section we elaborate on each part and list each paper's contribution to their respective part. Lastly, an overview of this thesis is presented with Figure 1.1.

Part one answers ***T.RQ*1**, ***T.RQ*2**, ***T.RQ*3**, part two gives important background to part three which answers ***T.RQ*4** and ***T.RQ*5**.

1.3.1 Reinforcement learning and machine learning for personalisation and engagement in e-Health

This part answers ***T.RQ*1**, ***T.RQ*2**, and ***T.RQ*3** and aims to understand how RL and machine learning techniques can be used to better personalisation and engagement in the domain of e-Health. In paper [V] we perform a systematic literature review study. Within this review we explore the various settings, including e-Health, in which RL has been used for personalisation. We do so by proposing a framework of evaluation settings, as well as reviewing the solutions and evaluation strategies adopted. During our investigation, we observed that the majority of the RL models found were either used by applying one model to all of the data (*i.e.,* one-size-fits-all) or by using one model per user (*i.e.,* on an individual level). Little work was found on the idea of pooling data together of similar users (*i.e.,* clustering) and training RL models on that data, so that each pooled group has their own policy.

In paper [I] we bring forth a contribution to RL for personalisation by systematically exploring how different clustering techniques and distance metrics can improve conventional RL. Within this work we use a simulation environment to empirically study how cluster-based RL differs from RL used on an individual level or a one-size-fits-all approach. Our results show that clustering configurations using high-level features significantly outperform the other two non cluster-based RL techniques. An example of a high-level feature, would be the average number of times a user works out in a week. To further improve cluster-based RL approaches, in paper [IV], we address limitations within the domain of online clustering by developing a novel algorithm called CluStream-GT.

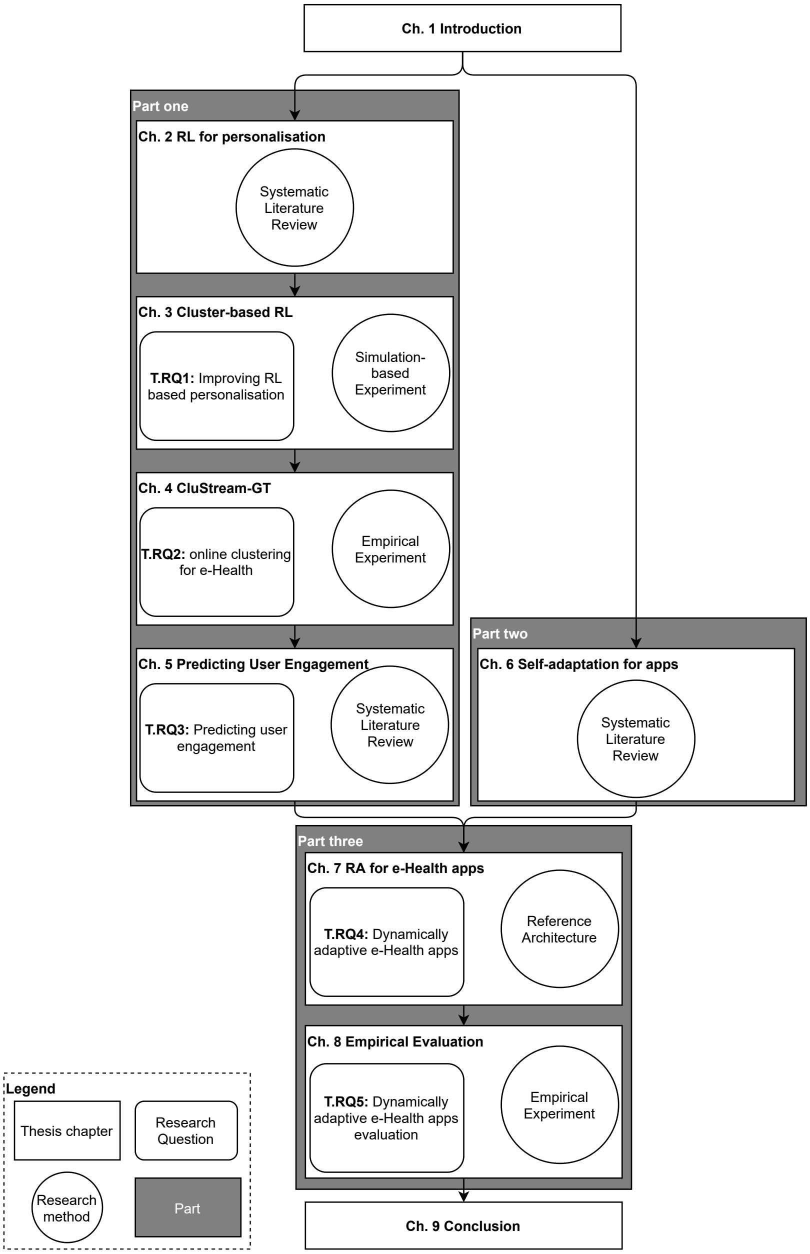


Figure 1.1: *The overview of the thesis, including the parts, chapters, research questions, and methods.*

Standing for CluStream for Growing Time-series, we designed CluStream-GT to overcome found limitations with the state of the art in online clustering and making an algorithm better suited for the e-Health domain. We do so by 1) allowing CluStream-GT to continuously cluster even when the number of total users changes and 2) when the length of the time-series of the existing patients changes. Whilst the second challenge is tackled by some state of the art approaches, none were found that could cope with the former. Our empirical results show that CluStream-GT is able to cluster time-series data more efficiently than other online clustering methods, whilst being comparatively effective.

An important target of personalisation is user engagement. Within the domain of e-Health apps the longer the user is engaged the more effective the health intervention will be. It then becomes a crucial aspect of app development to be able to predict user engagement and consequently their potential disengagement. In paper [III] we propose, apply, and evaluate a framework composed of several machine learning techniques used to predict user engagement. The framework is empirically evaluated using a year long observational dataset collected by the real world deployment of a waste-recycling app. We show that the non-domain specific features used in these models are successful in predicting user engagement. Whilst the app was not within the e-Health domain, the features used are generic enough that any app is able to collect them, including an e-Health app.

1.3.2 Self-adaptation in mobile applications

This part is related to ***T.RQ*4** and ***T.RQ*5**. It provides important background knowledge on the state of the art of self-adaption in the context of apps. In paper [II] we perform a systematic literature review study. In this review we examine the use of self-adaptation in the context of mobile applications, including e-Health apps. We propose a customised classification framework used to classify and compare self-adaptive approaches that were found by the review. From this review we also identified a number of shortcomings within the field. A notable finding was the lack of work done on self-adaptation with the objective of achieving non-technical goals. Examples of non-technical goals are promoting user behavioural change and lifestyle improvements. Furthermore, we identified a lack of self-adaptation techniques that adapted due to changes in smart-objects and third-party applications.

1.3.3 Creating self-adaptive and personalised e-Health mobile applications

This part focuses on answering ***T.RQ*4** and ***T.RQ*5**. Its aim is to find a solution to how to dynamically adapt e-Health apps and overcome the shortcomings identified in paper [II]. In paper [VI] we propose a reference architecture (RA) for enabling AI-based personalisation and self-adaptation of e-Health apps. The RA combines the previously described techniques into one software architecture. It does so by introducing multiple self-adaptive components across the architecture and having these work together with our cluster based RL approach and user engagement models. As a result of this combination of techniques, the RA is able to have these main characteristics: 1) guaranteeing the correct functioning of the given features with the use of runtime adaptation strategies and connected IoT devices, 2) personalising the given health interventions and provided services and adapting such services to the user's context (*e.g.,* environment, weather forecast), 3) allowing the RA to be applied to a singular e-Health app and by integrating the services of existing third-party e-Health apps, and 4) supporting the participation of domain experts into the system.

We then expand on this work with paper [VII]. Here we provide documentation of the methodology and viewpoint definition used to develop the RA, report on a scenario-based evaluation of it, include a goal model to be used with it, and overall report the RA within the broader context of social sustainability.

Lastly, in paper [VIII], we conduct two experiments to empirically evaluate our described RA. The first experiment is a user study, whilst the second experiment is a measurement-based experiment. We implement an e-Health app by using our RA as guide and use the app to empirically test end user usability and perception in the first experiment, and energy and performance impact in the second experiment. These experiments were conducted to explore how an app complying to our RA behaves in real world scenarios and how successful our proposed RA can be in guiding the development of dynamically adapting e-Health apps.

1.4 Contributions

The main contributions present in this thesis to the fields of AI and Software Engineering, applied to e-Health mobile applications are:

1. ***A rigorous map into the current state of the art use of RL for personalisation:*** An overview and categorisation of RL applications used for personalisation across application domains, including e-Health. This is presented together with a framework for classifying personalisation settings. This contribution relates to the part of reinforcement learning and machine learning for personalisation and engagement in e-Health and helps answer ***T.RQ*1**.

2. ***Data-efficient and effective techniques for personalisation:*** A systematic exploration of which clustering techniques and distance metrics perform best in aiding RL in delivering better policies for personalisation whilst increasing the initial learning speed and overcoming the cold start problem. Design, implementation and evaluation of a state of the art algorithm for online clustering of time series data for the e-Health domain. This contribution relates to reinforcement learning and machine learning for personalisation and engagement in e-Health and helps answer ***T.RQ*1** and ***T.RQ*2**.

3. ***Machine Learning models to predict user engagement in mobile apps:*** Design and implementation of a reusable framework for predicting user engagement in mobile apps. An empirical evaluation of various machine learning models used to characterise user engagement in mobile

apps. This contribution relates to reinforcement learning and machine learning for personalisation and engagement in e-Health and answers ***T.RQ*3**.

4. ***A rigorous map into the current state of the art use of self-adaptation for mobile apps:*** An up to date systematic review of the literature on self-adaptation in the context of mobile apps. Furthermore, a customised classification framework used for understanding, classifying and comparing self-adaptive approaches used in the context of mobile apps. Lastly, a discussion of the findings, research challenges, and the main application domains found in the literature. This contribution relates to part self-adaptation in mobile applications and helps answer ***T.RQ*4**.

5. ***An RA for personalised self-adaptive e-Health apps:*** Design and evaluation of a unique RA created for personalised self-adaptive e-Health apps. Frame the created RA in the context of social sustainability and how it can be used within the domain of e-Health to address this dimension of sustainability. Utilise the RA to guide the implementation of an app. Design and execution of a measurement-based experiment to test the performance and energy impact of the app on the end users' smartphone. Design and execution of a user study to research usability and end users' perception of the app. This contribution relates to all of the parts and answers ***T.RQ*4** and ***T.RQ*5**.

List of Papers

This thesis is the result of four years of research, and is constructed using the content of four conference papers, two journal papers and one book chapter. These papers are listed below, along with details of my contribution to each one.

part	**2018**	**2019**	**2020**
AI for personalisation and engagement	[I]	[III, IV]	[V]
Self-adaptation in apps		[II]	
Creating self-adaptive and personalised e-Health apps			[VI, VII, VIII]

[I] Grua, E. M., & Hoogendoorn, M. (2018, November). Exploring clustering techniques for effective reinforcement learning based personalization for health and wellbeing. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 813-820). IEEE.

I helped create an open-source simulation environment for e-Health. I developed the idea of using clustering with RL. I used the created environment to implement and execute experiments using clustering and RL techniques. Lastly, I conducted the analysis and wrote a significant amount of the paper.

[II] Grua, E. M., Malavolta, I., & Lago, P. (2019, May). Self-adaptation in mobile apps: a systematic literature study. In 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) (pp. 51-62). IEEE.

I identified the lack of an up-to-date systematic review on self-adaptation on the context of mobile apps. I equally contributed to the study design and creation of the customised classification framework presented. I conducted all of the paper gathering and most of the selection and analysis. Lastly, I wrote most of the paper.

[III] Barbaro, E., Grua, E. M., Malavolta, I., Stercevic, M., Weusthof, E., & van den Hoven, J. (2020). Modelling and predicting User Engagement in mobile applications. Data Science, (Preprint), 1-17.

The idea of using machine learning models to predict user engagement was of the first author. A few of the machine learning models were implemented by other authors. I replicated and validated these models. I then selected, implemented and ran all other machine learning models present in the paper. I conducted the data analysis and extracted the results. I contributed in the writing and revision of the whole paper.

[IV] Grua, E. M., Hoogendoorn, M., Malavolta, I., Lago, P., & Eiben, A. E. (2019, October). Clustream-GT: online clustering for personalization in the health domain. In IEEE/WIC/ACM International Conference on Web Intelligence (pp. 270-275).

I identified the lack of an online clustering algorithm fit for the health domain. I created and implemented a novel algorithm for online clustering for the health domain. I then designed and implemented the evaluation approach. Lastly, I conducted the analysis of the results and wrote most of the paper.

[V] den Hengst, F., Grua, E. M., el Hassouni, A., & Hoogendoorn, M. (2020). Reinforcement learning for personalization: A systematic literature review. Data Science (pp. 1-41).

The idea of conducting this systematic literature review (SLR) was of the first author. I, however, guided the first author in the process of conducting the SLR and helped him with understanding the PRISMA standard for reporting on the SLR and its components. I contributed equally to the other authors to the screening phases. I then contributed equally to the first author in the data collection phase. I helped with the data analysis and equally contributed in examining results and extrapolating the identified shortcomings. Lastly, I helped with writing and reviewing several sections of the paper.

[VI] Grua, E. M., De Sanctis, M., & Lago, P. (2020, September). A Reference Architecture for Personalized and Self-adaptive e-Health Apps. In European Conference on Software Architecture (pp. 195-209). Springer, Cham.

I proposed the idea of a RA for personalised and self-adaptive e-Health apps. I performed most of the design of the RA. Lastly, I wrote most of the paper.

[VII] Grua, E. M., De Sanctis, M., Malavolta, I., Hoogendoorn, M., & Lago, P. (2021). Social Sustainability in the e-Health Domain via Personalized and Self-adaptive Mobile Apps. Software Sustainability. Springer, Cham. To appear (book chapter).

As an extension to paper [VI], I proposed and conducted the scenario based evaluation as well as defining our RA in the scope of social sustainability. I contributed to the viewpoint definition and wrote most of the paper.

[VIII] Grua, E. M., De Sanctis, M., Malavolta, I., Hoogendoorn, M., & Lago, P. (2021). An Evaluation of the Effectiveness of Personalization and Self-Adaptation for e-Health Apps. Elsevier. Under review (journal).

As an extension to papers [VI, VII], I used our RA to guide the implementation of an app. I mostly designed two experiments to evaluate the implemented app. I conducted both experiments and performed the analyses on the collected results. Lastly, I wrote most of the paper.

1.5 Extra Publications

Schneider, A. F., Matinfar, S., Grua, E. M., Casado-Mansilla, D., & Cordewener, L. (2018, May). Towards a sustainable business model for smartphones: Combining product-service systems with modularity. In ICT4S (pp. 82-99).

I contributed equally to the forming of the idea proposed, as well as the writing of the paper.

Chan-Jong-Chu, K., Islam, T., Exposito, M. M., Sheombar, S., Valladares, C., Philippot, O., Grua, E.M., & Malavolta, I. (2020). Investigating the correlation between performance scores and energy consumption of mobile web apps. In Proceedings of the International Conference on Evaluation and Assessment on Software Engineering (EASE), pp. 190–199.

I contributed in the idea forming phase. Furthermore, I helped with operating the tool used for the experiments and with the methodology of the analysis.

Malavolta, I., Grua, E. M., Lam, C. Y., De Vries, R., Tan, F., Zielinski, E., Peters, & Kaandorp, L. (2020, September). A framework for the automatic execution of measurement-based experiments on android devices. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering Workshops (pp. 61-66).

I implemented parts of the framework. I helped with the design of other parts of the framework. Lastly, I tested and revised a significant amount of the framework. Furthermore, it is this framework that was used in the measurement-based experiment conducted in Chapter 8.

Part I

Reinforcement learning and machine learning for personalisation and engagement in e-Health

2

Reinforcement Learning for Personalisation

Chapter 2 was published as:

den Hengst, F., Grua, E. M., el Hassouni, A., & Hoogendoorn, M. (2020). Reinforcement learning for personalization: A systematic literature review. Data Science (pp. 1-41).

Abstract - This chapter provides important background knowledge which will be used to answer ***T.RQ*1**. The major application areas of reinforcement learning (RL) have traditionally been game playing and continuous control. In recent years, however, RL has been increasingly applied in systems that interact with humans. RL can personalise digital systems to make them more relevant to individual users. Challenges in personalisation settings may be different from challenges found in traditional application areas of RL. An overview of work that uses RL for personalisation, however, is lacking. In this work, we introduce a framework of personalisation settings and use it in a systematic literature review. Besides setting, we review solutions and evaluation strategies. Results show that RL has been increasingly applied to personalisation problems and realistic evaluations have become more prevalent. RL has become sufficiently robust to apply in contexts that involve humans and the field as a whole is growing. However, it seems not to be maturing: the ratios of studies that include a comparison or a realistic evaluation are not showing upward trends and the vast majority of algorithms are used only once. This review can be used to find related work across domains, provides insights into the state of the field and identifies opportunities for future work.

2.1 Introduction

For several decades, both academia and commerce have sought to develop tailored products and services at low cost in various application domains. These reach far and wide, including medicine [132; 16], human-computer interaction [218; 118], product, news, music and video recommendations [293; 295; 368] and even manufacturing [278; 85]. When products and services are adapted to individual tastes, they become more appealing, desirable, informative, e.g. *relevant* to the intended user than one-size-fits all alternatives. Such adaptation is referred to as *personalisation* [109].

Digital systems enable personalisation on a grand scale. The key enabler is data. While the software on these systems is identical for all users, the behaviour of these systems can be tailored based on experiences with individual users. For example, Netflix's[1] digital video delivery mechanism includes tracking of views and ratings. These ease the gratification of diverse entertainment needs as they enable Netflix to offer instantaneous personalised content recommendations. The ability to adapt system behaviour to individual tastes is becoming increasingly valuable as digital systems permeate our society.

Recently, reinforcement learning (RL) has been attracting substantial attention as an elegant paradigm for personalisation based on data. For any particular environment or user state, this technique strives to determine the sequence of actions to maximise a reward. These actions are not necessarily selected to yield the highest reward *now*, but are typically selected to achieve a high reward in the long term. Returning to the Netflix example, the company may not be interested in having a user watch a single recommended video instantly, but rather aim for users to prolong their subscription after having enjoyed many recommended videos. Besides the focus on long-term goals in RL, rewards can be formulated in terms of user feedback so that no explicit definition of desired behaviour is required [26; 149].

RL has seen successful applications to personalisation in a wide variety of domains. Some of the earliest work, such as [314], [312] and [394] focused on web services. More recently, [198] showed that adding personalisation to an existing online news recommendation engine increased click-through rates by 12.5%. Applications are not limited to web services, however. As an example from the health domain, [398] achieve optimal per-patient treatment plans to address advanced metastatic stage IIIB/IV non-small cell lung cancer in simulation. They state that 'there is significant potential of the proposed methodology for developing personalised treatment strategies in other cancers, in cystic fibrosis, and in other life-threatening diseases'. An early example of tailoring intelligent tutor behaviour using RL can be found in [227]. A more recent example in this domain, [137], compared the effect of personalised and non-personalised affective feedback in language learning with a social robot for children and found that personalisation significantly impacts psychological valence.

Although the aforementioned applications span various domains, they are

[1] https://www.netflix.com

similar in solution: they all use traits of users to achieve personalisation, and all rely on implicit feedback from users. Furthermore, the use of RL in contexts that involve humans poses challenges unique to this setting. In traditional RL subfields such as game-playing and robotics, for example, simulators can be used for rapid prototyping and *in-silico* benchmarks are well established [181; 103; 32; 44]. Contexts with humans, however, may be much harder to simulate and the deployment of autonomous agents in these contexts may come with different concerns regarding for example safety. When using RL for a personalisation problem, similar issues may arise across different application domains. An overview of RL for personalisation across domains, however, is lacking. We believe this is not to be attributed to fundamental differences in setting, solution or methodology, but stems from application domains working in isolation for cultural and historical reasons.

This chapter provides an overview and categorisation of RL applications for personalisation across a variety of application domains. It thus aids researchers and practitioners in identifying related work relevant to a specific personalisation setting, promotes the understanding of how RL is used for personalisation and identifies challenges across domains. We first provide a brief introduction of the RL framework and formally introduce how it can be used for personalisation. We then present a framework to classify personalisation settings by. The purpose of this framework is for researchers with a specific setting to identify relevant related work across domains. We then use this framework in a systematic literature review (SLR). We investigate in which settings RL is used, which solutions are common and how they are evaluated: Section 2.5 details the SLR protocol, results and analysis are described in Section 8.3. All data collected has been made available digitally [96]. Finally, we conclude with current trends challenges in Section 4.5.

2.2 Reinforcement learning for personalisation

RL considers problems in the framework of *Markov decision processes* or MDPs. In this framework, an agent collects rewards over time by performing actions in

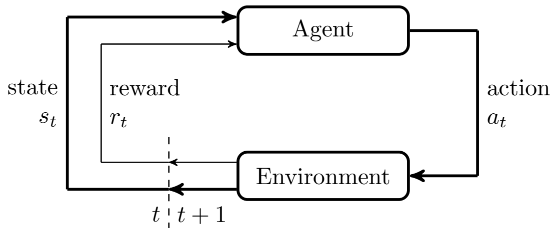


Figure 2.1: *The agent-environment in RL for personalisation from [334].*

an environment as depicted in Figure 2.1. The goal of the agent is to maximise the total amount of collected rewards over time. In this section, we formally introduce the core concepts of MDPs and RL and include some strategies to personalisation without aiming to provide an in depth introduction to RL. Following [334], we consider the related *multi-armed* and *contextual bandit* problems as special cases of the full RL problem where actions do not affect the environment and where observations of the environment are absent or present respectively. We refer the reader to [334], [376] and [336] for a full introduction.

An MDP is defined as a tuple $\langle \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{R}, \gamma \rangle$ where $\boldsymbol{S} \in \{\boldsymbol{s_1}, \dots, \boldsymbol{s_n}\}$ is a finite set of states, $\boldsymbol{A} \in \{\boldsymbol{a_1}, \dots, \boldsymbol{a_m}\}$ a finite set of system actions, $\boldsymbol{T} : \boldsymbol{S} \times \boldsymbol{A} \times \boldsymbol{S} \rightarrow [\boldsymbol{0}, \boldsymbol{1}]$ a probabilistic transition function, $\boldsymbol{R} : \boldsymbol{S} \times \boldsymbol{A} \rightarrow \mathbb{R}$ a reward function and $\boldsymbol{\gamma} \in [\boldsymbol{0}, \boldsymbol{1}]$ a factor to discount future rewards. At each time step $\boldsymbol{t}$, the system is confronted with some state $\boldsymbol{s_t}$, performs some action $\boldsymbol{a_t}$ which yields a reward $\boldsymbol{r_{t+1}} : \boldsymbol{R}(\boldsymbol{s_t}, \boldsymbol{a_t})$ and some state $\boldsymbol{s_{t+1}}$ following the probability distribution $\boldsymbol{T}(\boldsymbol{s_t}, \boldsymbol{a_t})$. A series of these states, actions and rewards from the onset to some terminal state $\boldsymbol{T}$ is called a trajectory $\boldsymbol{tr} : \langle \boldsymbol{s_{t_0}}, \boldsymbol{a_{t_0}}, \boldsymbol{r_{t_1}}, \boldsymbol{s_{t_1}}, \dots, \boldsymbol{s_{T-1}}, \boldsymbol{a_{T-1}}, \boldsymbol{r_T}, \boldsymbol{s_T} \rangle$. These trajectories typically contain the interaction histories for users with the system. A single trajectory can describe a single session of the user interacting with the system or can contain many different separate sessions. Multiple trajectories may be available in a data set $\boldsymbol{D} \in \{\boldsymbol{tr_1}, \dots, \boldsymbol{tr_\ell}\}$. The goal is to find a policy $\boldsymbol{\pi^*}$ out of all $\boldsymbol{\Pi} : \boldsymbol{S} \times \boldsymbol{A} \rightarrow [\boldsymbol{0}, \boldsymbol{1}]$ that maximises the sum of future rewards at any $\boldsymbol{t}$, given an end time $\boldsymbol{T}$:

$$\boldsymbol{G_t : \sum_{k=t}^{T-1} \gamma^{k-t} r_{k+1}} \tag{2.1}$$

If some expectation $\mathbb{E}_{\boldsymbol{\pi}}$ over the future reward for some policy $\boldsymbol{\pi}$ can be formulated, a value can be assigned to some state $\boldsymbol{s}$ given that policy:

$$\boldsymbol{V_\pi(s) = \mathbb{E}_\pi[G_t | s_t = s]} \tag{2.2}$$

Similarly, a value can be assigned to an action $\boldsymbol{a}$ in a state $\boldsymbol{s}$:

$$\boldsymbol{Q_\pi(s, a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a]} \tag{2.3}$$

Now the optimal policy $\boldsymbol{\pi^*}$ should satisfy $\boldsymbol{\forall s \in S, \forall \pi \in \Pi : V_{\pi^*}(s) \geq V_\pi(s)}$ and $\boldsymbol{\forall s \in S, a \in A, \forall \pi \in \Pi : Q_{\pi^*}(s, a) \geq Q_\pi(s, a)}$. Assuming a suitable $\mathbb{E}_{\boldsymbol{\pi^*}}[\boldsymbol{G}]$, $\boldsymbol{\pi^*}$ consists of selecting the action that is expected to yield the highest sum of rewards:

$$\boldsymbol{\pi^*(s) = \arg\max_a Q_{\pi^*}(s, a), \forall s \in S, a \in A} \tag{2.4}$$

With these definitions in place, we now turn to methods of finding $\boldsymbol{\pi^*}$. Such methods can be categorised by considering which elements of the MDP are known. Generally, $\boldsymbol{S}$, $\boldsymbol{A}$ and $\boldsymbol{\gamma}$ are determined upfront and known. $\boldsymbol{T}$ and $\boldsymbol{R}$, on the other hand, may or may not be known. If they are both known, the expectation $\mathbb{E}_{\boldsymbol{\pi}}[\boldsymbol{G}]$ is directly available and a corresponding $\boldsymbol{\pi^*}$ can be found analytically. In some settings, however, $\boldsymbol{T}$ and $\boldsymbol{R}$ may be unknown and $\boldsymbol{\pi^*}$ must be found empirically. This can be done by estimating $\boldsymbol{T}$, $\boldsymbol{R}$, $\boldsymbol{V}$, $\boldsymbol{Q}$ and finally $\boldsymbol{\pi^*}$ or a combination thereof using data set $\boldsymbol{D}$. Thus, if we include approximations in Eq. (2.4), we get:

$$\boldsymbol{\hat{\pi^*}(s) | D = \arg\max_a \hat{Q}_{\hat{\pi^*}}(s, a) | D, \forall s \in S, a \in A} \tag{2.5}$$

As $\boldsymbol{D}$ may lack the required trajectories for a reasonable $\mathbb{E}_{\boldsymbol{\hat{\pi^*}}}[\boldsymbol{G}]$ and may even be empty initially, *exploratory* actions can be selected to enrich $\boldsymbol{D}$. Such actions need not follow $\hat{\boldsymbol{\pi}^*}$ as in Eq. (2.5) but may be selected through some other mechanism such as sampling from the full action set $\boldsymbol{A}$ randomly.

Having introduced RL briefly, we continue by exploring some strategies in applying this framework to the problem of personalising systems. We return to our earlier example of a video recommendation task and consider a set of $\boldsymbol{n}$ users $\boldsymbol{U} \in \{\boldsymbol{u_1}, \ldots, \boldsymbol{u_n}\}$. A first way to adapt software systems to an individual users' needs is to define a separate environment, corresponding MDP and RL agent for each user. The overall goal becomes to find a set of optimal policies $\{\boldsymbol{\pi_1^*}, \ldots, \boldsymbol{\pi_n^*}\}$ for a set of environments formalised as MDPs $\boldsymbol{M} : \{\boldsymbol{M_1} : \langle \boldsymbol{S_1}, \boldsymbol{A_1}, \boldsymbol{T_1}, \boldsymbol{R_1}, \gamma_1 \rangle, \ldots, \boldsymbol{M_n} : \langle \boldsymbol{S_n}, \boldsymbol{A_n}, \boldsymbol{T_n}, \boldsymbol{R_n}, \gamma_n \rangle\}$. In the case of approximations as in Eq. (2.5), these are made per MDP based on data set $\boldsymbol{D_i}$ with trajectories only involving that environment. In the running example, videos would be recommended to a user based on previous video recommendations and selections of that particular user. The benefit of isolated MDPs is that differences between $\boldsymbol{T_i}$ and $\boldsymbol{T_j}$ or between $\boldsymbol{R_i}$ and $\boldsymbol{R_j}$ for MDPs $\boldsymbol{M_i} \neq \boldsymbol{M_j}$ are handled naturally, e.g. such differences do not make $\mathbb{E}_{\boldsymbol{\pi_i}}[\boldsymbol{G}]$ incorrect. On the other hand, similarities between $\boldsymbol{T_i}, \boldsymbol{T_j}$ and $\boldsymbol{R_i}, \boldsymbol{R_j}$ cannot be used. For example, consider a video recommendation task with $\boldsymbol{S_{ij}} = \{\boldsymbol{morning}, \boldsymbol{afternoon}, \boldsymbol{night}\}$. If two users $\boldsymbol{u_i} \neq \boldsymbol{u_j}$ are both using a video service in the $\boldsymbol{morning}$ state, they may both like to watch a breakfast news broadcast whereas in the $\boldsymbol{night}$ state they may both prefer a talk show. Learning such patterns for each environment individually may require a substantial number of trajectories and may be infeasible in some settings, such as those where users cannot be identified across trajectories or those where each user is expected to contribute only one trajectory to $\boldsymbol{D_i}$.

An alternative approach is to define is a single agent and MDP with user-specific information in the state space $\boldsymbol{S}$ and learn a single $\boldsymbol{\pi^*}$ for all users [97]. In some settings, users can be described using a function that returns a vector representation of the $\boldsymbol{l}$ features that characterise a user $\boldsymbol{\phi} : \boldsymbol{U} \rightarrow \langle \boldsymbol{\phi_1}(\boldsymbol{U}), \ldots, \boldsymbol{\phi_l}(\boldsymbol{U}) \rangle$. Such a vector could for example contain age, favourite genre and viewing history. If two users $\boldsymbol{u_j} \neq \boldsymbol{u_i}$ have both enjoyed the first "Lord of the Rings" movie and viewer $\boldsymbol{u_j}$ has followed up on a recommendation of its sequel by the system then this sequel may be a suitable recommendation for the other viewer $\boldsymbol{u_i}$ as well. Generally, this approach can be valuable when it is unclear which elements of trajectories of users $\boldsymbol{u_j}$ should be used in determining $\boldsymbol{\pi_i^*}$. Conceptually, finding $\boldsymbol{\pi^*}$ now includes determining $\boldsymbol{u_i}$'s preference for actions given a state and determining the relationship between user preferences. This approach should therefore be able to overcome the negative transfer problem described below when enough trajectories are available. The growth in state space size, on the other hand, may require an exorbitant number of trajectories in $\boldsymbol{D}$ due to the curse of dimensionality [34]. Thus, $\boldsymbol{\phi}$ is to be carefully designed or dimensionality reduction techniques are to be used in approaches following this strategy. As a closing remark on this approach to personalisation, we note that the distinction between task-related and user-specific information is somewhat artificial as $\boldsymbol{S}$ may already contain $\boldsymbol{\phi(U)}$ in many practical settings and we stress that the distinction is made for illustrative purposes here.

A third category of approaches can be considered as a middle ground between learning a single $\boldsymbol{\pi^*}$ and learning a $\boldsymbol{\pi_i^*}$ per user. It is motivated by the idea that users and corresponding environments may be similar. If this is the case, then trajectories $\boldsymbol{D_j}$ from some similar environment $\boldsymbol{M_j} \neq \boldsymbol{M_i}$ may prove useful in estimating $\mathbb{E}_{\boldsymbol{\pi_i}}[\boldsymbol{G}]$. One such an approach is based on clustering [227; 337; 145; 107]. Formally, it requires $\boldsymbol{q} \leq \boldsymbol{n}$ groups $\boldsymbol{G} \in \{\boldsymbol{g_1}, \ldots, \boldsymbol{g_q}\}$ and a mapping function $\boldsymbol{\Phi : M \rightarrow G}$. In practice, this mapping function is typically defined on the level of users $\boldsymbol{U}$ or the feature representation $\boldsymbol{\phi(U)}$. An RL agent is defined for every $\boldsymbol{g_p}$ and interacts with all environments $\boldsymbol{M_i, M_j, \Phi(M_i) = \Phi(M_j) = g_p}$. Trajectories in $\boldsymbol{D_i}$ and $\boldsymbol{D_j}$ are concatenated or *pooled* to form a single $\boldsymbol{D_p}$ which is used to approximate $\mathbb{E}_{\hat{\boldsymbol{\pi_p}}}[\boldsymbol{G}]$ for all $\boldsymbol{M_i, M_j}$. A combined $\boldsymbol{D_p}$ may be orders of magnitude bigger than an isolated $\boldsymbol{D_i}$, which may result in a much better approximation $\mathbb{E}_{\hat{\boldsymbol{\pi_p}}}[\boldsymbol{G}]|\boldsymbol{D_p}$ and a resulting $\hat{\boldsymbol{\pi_p^*}}(\boldsymbol{s})|\boldsymbol{D_p}$ that yields a higher reward in all environments. For example, users of the video recommendation service may be clustered by age and users in the 'infant' cluster may generally prefer children's movies over history documentaries. A related approach similarly uses trajectories $\boldsymbol{D_j}$ of other environments $\boldsymbol{M_j} \neq \boldsymbol{M_i}$ but still aims to find environment-specific $\boldsymbol{\pi_i^*}$. Trajectories in $\boldsymbol{D_j}$ are weighted during estimation of $\mathbb{E}_{\pi_i}[G]$ using some weighting scheme. This can be understood as a generalisation of the pooling approach. First, recall that $\boldsymbol{\Phi : M \rightarrow G}$ for the pooling approach and note that it can be rewritten to $\boldsymbol{\Phi : M \times M \rightarrow \{0, 1\}}$. The weighting scheme, now, is a generalisation where $\boldsymbol{\Phi : M \times M} \rightarrow \mathbb{R}$. Finding a suitable $\boldsymbol{\Phi}$ can be challenging in itself and depends on the availability of user features, trajectories and the task at hand. Typical strategies are to define $\boldsymbol{\Phi}$ in terms of similarity of feature representations of users $[\boldsymbol{\phi(u_i), \phi(u_j)}]$ or similarity of $\boldsymbol{D_i, D_j}$. The two previous approaches work under the assumption that $\boldsymbol{T_i, T_j}$ and $\boldsymbol{R_i, R_j}$ are similar and that $\boldsymbol{\Phi}$ is suitable. If either of these assumptions is not met, pooling data may result in a policy that is suboptimal for both $\boldsymbol{M_i}$ and $\boldsymbol{M_j}$. This phenomenon is typically referred to as the *negative transfer problem* [267].

2.3 Algorithms

In this section we provide an overview of specific RL techniques and algorithms used for personalisation. This overview is the result of our systematic literature review as can be seen in Table 2.4. Figure 2.2 contains a diagram of the discussed techniques. We start with a subset of the full RL problem known as $\boldsymbol{k}$-armed bandits. We bridge the gap towards the full RL setting with contextual bandits approaches. Then, value-based and policy-gradient RL methods are discussed.

2.3.1 Multi-armed bandits

Multi-armed bandits is a simplified setting of RL. As a result, it is often used to introduce basic learning methods that can be extended to full RL algorithms [334]. In the non-associative setting, the objective is to learn how to act optimally in a single situation. Formally, this setting is equivalent to an MDP with a single state. In the associative or *contextual* version of this setting, actions are taken in more than one situation. This setting is closer to the full RL problem yet it lacks an important trait of full RL, namely that the selected action affects the situation. Both associative and non-associative multi-armed

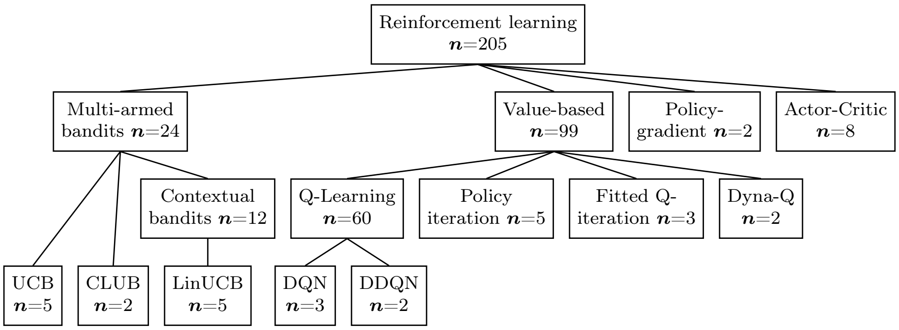


Figure 2.2: *Overview of types of RL algorithms discussed in this section and the number of uses in publications included in this survey. See Table 2.4 for a list of all (families of) algorithms used by more than one publication.*

bandit approaches do not take into account temporal separation of actions and related rewards.

In general, multi-armed bandit solutions are not suitable when success is achieved by sequences of actions. Non-associative k-armed bandits solutions are only applicable when context is not important. This makes them generally unsuitable for personalisation as it typically utilises different personal contexts for different users by offering a different functionality. In some niche areas, however, k-armed bandits are applicable and can be very attractive due to formal guarantees on their performance. If context is of importance, contextual bandit approaches provide a good starting point for personalising an application. These approaches hold a middle ground between non-associative multi-armed bandits and full RL solutions in terms of modelling power and ease of implementation. Their theoretical guarantees on optimality are less strong than their k-armed counterparts but they are easier to implement, evaluate and maintain than full RL solutions.

2.3.1.1 k-Armed bandits

In a k-armed bandit setting, one is constantly faced with the choice between k different actions [334]. Depending on the selected action, a scalar reward is obtained. This reward is drawn from a stationary probability distribution. It is assumed that an independent probability distribution exists for every action. The goal is to maximise the expected total reward over a certain period of time. Still considering the k-armed bandit setting, we assign a value $Q(a)$ to each of the k actions and define this value as the expected reward given that the action was selected. The expected reward given that an action a is selected is defined as follows:

$$Q(a) = \mathbb{E}[r_t | a_t = a]. \tag{2.6}$$

In a trivial problem setting, one knows the exact value of each action and selecting the action with the highest value would constitute the optimal policy. In more realistic problems, it is fair to assume that one cannot know the values of the actions exactly. In this case, one can estimate the value of an action. We denote this estimated value with $\hat{Q}(a)$ and our goal is to have estimate $\hat{Q}(a)$ as close to the true $Q(a)$ as possible.

At each time step t, estimates of the values of actions are obtained. Always selecting the actions with the highest estimated value is called greedy action selection. In this case we are exploiting the knowledge we have built about the values of the actions. When we select actions with a lower expected value, we say we are exploring. In this case we are improving the estimates of values for these actions. In the balancing act of exploration and exploitation, we opt for exploitation to maximise the expected total reward for the next step, while opting for exploration could results in higher expected total reward in the long run.

2.3.1.2 Action-value methods for multi-armed bandits

Action-value methods [334] denote a collections of methods used for estimating the values of actions. The most natural way of estimating the action-values is to average the rewards that were observed. This method is called the sample-average method. The value estimate $\hat{Q}_\pi(a)$ is then defined as:

$$\hat{Q}(a) = \frac{\sum_{i=1}^{t-1} r_i \cdot \mathbb{1}_{a_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{a_i=a}} \tag{2.7}$$

where $\mathbb{1}_{a_i=a}$ is 1 when $a_i = a$ is true and 0 otherwise. A default value is assigned to $\hat{Q}(a)$ when the denominator is zero. As the denominator approaches infinity, the estimate $\hat{Q}(a)$ converges to the true $Q(a)$. Again, the most basic way of selecting actions is the greedy action selection method. Here the action with the highest value is selected. In the case of a tie, one action is selected using tie-breaking methods such as random selection. Greedy action selection is defined as follows for any time point t:

$$a_t = \arg\max_a \hat{Q}(a). \tag{2.8}$$

Greedy action selection only exploits knowledge built up using the action-value method and only maximises the immediate reward. This can lead to incorrect action-value approximations because actions with e.g. low *estimated* but high *actual* values are not sampled. An improvement over this greedy action selection is to randomly explore with a small probability ϵ. This method is named the ϵ-greedy action selection. A benefit of this method is that, while it is relatively simple, in the limit $\hat{Q}(a)$ will converge to $Q(a)$ [334]. This indicates that the probability of selecting the optimal action is then greater than $1 - \epsilon$ which is near certainty.

2.3.1.3 Incremental Implementation

In Section 2.3.1.2 we discussed a method to estimate action-values using sample-averaging. To ensure the usability of these method in real-world applications, we need to be able to compute these values in an efficient way. Assume a setting with one action. At each iteration j a reward r_{t_j} is obtained after selecting an action. Let $\hat{Q}_n(a)$ denote the estimate value of the action after $n-1$ iterations.

We can then define:

$$\hat{Q}_n(a) = \frac{r_{t_1} + r_{t_2} + r_{t_3} + ... + r_{t_{n-1}}}{n-1}. \tag{2.9}$$

Using this approach would mean storing the values of all the rewards to recalculate $\hat{Q}_n(a)$ from scratch at every iteration. There is however a more efficient way for calculating $\hat{Q}_n(a)$ that is constant in memory and computation time. Rewriting it yields the following update rule:

$$\hat{Q}_{n+1}(a) = \hat{Q}_n(a) + \frac{1}{n}[r_{t_n} - \hat{Q}_n(a)], \tag{2.10}$$

where the term $\hat{Q}_n(a)$ represents the old estimate, $[r_n - \hat{Q}_n(a)]$ the error in the estimate we made of the reward and $\frac{1}{n}$ the learning rate.

2.3.1.4 UCB: Upper-Confidence Bound

The greedy and ϵ-greedy action selection methods were discussed in Section 2.3.1.2 and it was introduced that exploration is required to establish good action-value estimates. Although ϵ-greedy explores all actions eventually, it does so randomly. A better way of exploration would take into account the action-value's proximity to the optimal value and the uncertainty in the value estimations. Intuitively, we want a selected action a to either provide a good immediate reward or else some very useful information in updating $\hat{Q}(a)$. An approach that uses this idea is the upper confidence bound action selection (UCB) method [334; 19; 125]. UCB is defined as follows at time step t:

$$a_t = \arg\max_a \left[\hat{Q}_n(a) + c \cdot \sqrt{\frac{\ln t}{N_t(a)}} \right] \tag{2.11}$$

where $N_t(a)$ is how often action a was chosen up to time t and $c > 0$ is a parameter to control the rate of exploration. The square root term denotes the level of uncertainty in the approximation of the value of action a. Hence, UCB provides an upper bound for the true value of the action a. Here, c is used to define the confidence level. When the action a is selected often, $N_t(a)$ will become larger which leads the uncertainty term to decrease. On the other hand, if the action $\boldsymbol{a}$ is not selected very often, $\boldsymbol{t}$ increases and so does the uncertainty term.

$\boldsymbol{k}$-Armed bandit approaches address the trade-off between exploitation and exploration directly. It has been shown that the difference between the obtained rewards and optimal rewards, or the *regret*, is at best logarithmic in the number of iterations $\boldsymbol{n}$ in the absence of prior knowledge of the action value distributions and in the absence of context [190]. UCB algorithms with a regret logarithmic in and uniformly distributed over $\boldsymbol{n}$ exist [19]. This makes them a very interesting choice when strong theoretical guarantees on performance are required.

Whether these algorithms are suitable, however, depends on the setting at hand. If there is a large number of actions to choose from or when the task is not stationary $\boldsymbol{k}$-armed bandits are typically too simplistic. In a news recommendation task, for example, exploration may take longer than an item stays relevant. Additionally, $\boldsymbol{k}$-armed bandits are not suitable when action values are conditioned on the situation at hand, that is: when a single action results in a different reward based on e.g. time-of-day or user-specific information such as in Section 2.2. In these scenarios, the problem formalisation of contextual bandits and the use of function approximation are of interest.

2.3.1.5 Contextual bandits

In the previous sections, action-values where not associated with different situations. In this section we extend the non-associative bandit setting to the associative setting of contextual bandits. Assume a setting with $\boldsymbol{n}$ $\boldsymbol{k}$-armed bandits problems. At each time step $\boldsymbol{t}$ one encounters a situation with a randomly selected $\boldsymbol{k}$-armed bandits problem. We can use some of the approaches that were discussed to estimate the action values. However, this is only possible if the true action-values change slowly between the different $\boldsymbol{n}$ problems [334]. Add to this setting the fact that now at each time $\boldsymbol{t}$ a distinctive piece of information is provided about the underlying $\boldsymbol{k}$-armed bandit which is not the actual action value. Using this information we can now learn a policy that uses the distinctive information to associate the $\boldsymbol{k}$-armed bandit with the best action to take. This approach is called contextual bandits and uses trial-and-error to search for the optimal actions and associates these actions with situation in which they perform optimally. This type of algorithm is positioned between $\boldsymbol{k}$-armed bandits and full RL. The similarity with RL lies in the fact that a policy is learned while the association with $\boldsymbol{k}$-armed bandits stems from the fact that actions only affect immediate rewards. When actions are allowed to affect the next situation as well then we are dealing with RL.

2.3.1.6 Function approximation: LinUCB and CLUB

Despite the good theoretical characteristics of the UCB algorithm, it is not often used in the contextual setting in practice. The reason is that in practice, state and action spaces may be very large and although UCB is optimal in the uninformed case, we may do better if we use obtained information across actions and situations. Instead of maintaining isolated sample-average estimates per action or per state-action pair such as in Sections 2.3.1.2 and 2.3.1.5, we can estimate a parametric payoff function approximated from data. The parametric function takes some feature description of actions for $\boldsymbol{k}$-armed bandit settings and state-action pairs for the contextual bandit setting and output some estimated $\hat{\boldsymbol{Q_\theta}}(\boldsymbol{a})$. Here, we focus on the contextual-bandit algorithms LinUCB and CLUB.

LinUCB (Linear Upper-Confidence Bound) uses linear function approximation to calculate the confidence interval efficiently in closed form [198]. Define the expected payoff for action $\boldsymbol{a}$ with the $\boldsymbol{d}$-dimensional featurised state $\boldsymbol{s_{t,a}}$ and $\boldsymbol{\Theta^*_a}$ a vector of unknown parameters as follows:

$$\mathbb{E}[\boldsymbol{r_a}|\boldsymbol{s_a}] = \boldsymbol{s_a^T \Theta_a^*}. \tag{2.12}$$

Using ridge regression, an estimate of $\boldsymbol{\hat{\Theta}_a}$ can be obtained [198]. Consequently, it can be shows that for any $\boldsymbol{\sigma > 0}$ and $\boldsymbol{s_a} \in \mathbb{R}^{\boldsymbol{d}}$ with $\boldsymbol{\alpha = 1 + \sqrt{ln(\frac{2}{\sigma})/2}}$ a reasonably tight estimate for the expected payoff of arm $\boldsymbol{a}$ can be obtained as follows:

$$\boldsymbol{a_t} = \mathbf{arg}\max_{\boldsymbol{a}} \left[\boldsymbol{s_a^T \Theta_a^*} + \boldsymbol{\alpha}\sqrt{\boldsymbol{s_a^T A_a^{-1} s_a}} \right], \tag{2.13}$$

where $\boldsymbol{A_a^{-1}} = \boldsymbol{D_a^T D_a} + \boldsymbol{I_d}$ and $\boldsymbol{D_a}$ a design matrix of dimension $\boldsymbol{m}$ x $\boldsymbol{d}$ whose rows are the $\boldsymbol{m}$ contexts that are observed, $\boldsymbol{b_a} \in \mathbb{R}^{\boldsymbol{m}}$ the corresponding response vector and $\boldsymbol{I_d}$ the $\boldsymbol{d}$ x $\boldsymbol{d}$ identity matrix [198].

Similar to LinUCB, CLUB (Clustering of bandits) utilises the linear bandit algorithm for payoff estimation [129]. In contrast to LinUCB, CLUB uses adaptive clustering in order to speed up the learning process. The main idea is to use confidence balls of user models estimate user similarity and share feedback across similar users. CLUB can thus be understood as a cluster-based alternative (see Section 2.2) to LinUCB algorithm.

2.3.2 Value-based RL

In value based RL, we learn an estimate $\boldsymbol{V}$ of the optimal value function $\boldsymbol{V_{\pi^*}}$ for a given policy $\boldsymbol{\pi}$. We do this with the aim of finding $\boldsymbol{\pi^*}$. Temporal-difference (TD) prediction is a method that learns from raw experiences without having to build a model of the environment the policy is interacting with [334]. In this section, we discuss various RL algorithms based on TD prediction.

2.3.2.1 Sarsa: on-policy temporal-difference RL

Sarsa is an on-policy temporal-difference method that learns an action-value function [334; 325]. Given the current behaviour policy $\boldsymbol{\pi}$, we estimate $\boldsymbol{\hat{Q_\pi}(a)}$ $\forall$ $\boldsymbol{s}$, and $\boldsymbol{a}$. This is done using transitions from state-action pair to state-action pair. Events of the form $\langle \boldsymbol{s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}} \rangle$ are used in the following update rule to estimate the state-action values:

$$\boldsymbol{\hat{Q}_\pi(s_t, a_t) = \hat{Q}_\pi(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \hat{Q}_\pi(s_{t+1}, a_{t+1}) - \hat{Q}_\pi(s_t, a_t) \right].} \tag{2.14}$$

This update rule is applied after every transition from $\boldsymbol{s_t}$ to $\boldsymbol{s_{t+1}}$. In case $\boldsymbol{s_{t+1}}$ is a terminal state, a value of zero is assigned. By doing this we are ensuring that the estimate $\boldsymbol{\hat{Q_\pi}}$ for a behaviour policy $\boldsymbol{\pi}$ while resulting in changes in $\boldsymbol{\pi}$ given $\boldsymbol{Q_\pi}$. Sarsa will converge to an optimal action-value function $\boldsymbol{Q_{\pi^*}}$ and hence an optimal policy $\boldsymbol{\pi^*}$ in the limit given that all possible state-action pairs

Algorithm 1 Sarsa - An on-policy temporal-difference RL algorithm

Parameters: learning rate $\boldsymbol{\alpha} \in \boldsymbol{(0, 1]}$ and $\boldsymbol{\epsilon} > \mathbf{0.0}$;
Initialise $\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}\ \forall\ \boldsymbol{s} \in \boldsymbol{S}$, $\boldsymbol{a} \in \boldsymbol{A}$. For terminal states initialise the value with 0.
for *each episode* **do**
  Initialise $\boldsymbol{s}$
  Choose action $\boldsymbol{a}$ in $\boldsymbol{s}$ using $\boldsymbol{\pi}$ derived from $\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}$ (e.g. $\boldsymbol{\epsilon - greedy}$)
  for *each step in episode* **do**
    Select action $\boldsymbol{a}$ and obtain reward $\boldsymbol{r}$ and next state $\boldsymbol{s'}$
    Take next action $\boldsymbol{a'}$ from $\boldsymbol{s'}$ following $\boldsymbol{\pi}$ derived from $\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}$ (e.g. $\boldsymbol{\epsilon -}$ $\boldsymbol{greedy}$)
    $\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) = \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) + \boldsymbol{\alpha} \left[\boldsymbol{r} + \gamma \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s'}, \boldsymbol{a'}) - \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a})\right]$
    Set $\boldsymbol{s} = \boldsymbol{s'}$ and $\boldsymbol{a} = \boldsymbol{a'}$
    Stop loop if $\boldsymbol{s}$ is terminal
  end
end

are visited an infinite amount of time [334]. Consequently, Sarsa converges to the greedy policy in the limit. Algorithm 1 shows Sarsa in more detail.

2.3.2.2 Q-learning: off-policy temporal-difference RL

Q-learning was one of the breakthroughs in the field of RL [334; 370]. Q-learning is classified as an off-policy temporal-difference algorithm for control. Similar to Sarsa, Q-learning approximates the optimal action-value function $\boldsymbol{Q}_{\boldsymbol{\pi^*}}$ by learning $\hat{\boldsymbol{Q}_{\boldsymbol{\pi^*}}}$. Differently from Sarsa, Q-learning learns $\hat{\boldsymbol{Q}_{\boldsymbol{\pi^*}}}$ independently of the policy being followed. The policy being followed still has an effect on the learning process, but only by determining which state-action pairs are visited and consequently updated. Algorithm 2 shows Q-learning in more detail. The update rule for Q-learning is defined as follows:

$$\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s_t}, \boldsymbol{a_t}) = \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s_t}, \boldsymbol{a_t}) + \boldsymbol{\alpha} \left[\boldsymbol{r_{t+1}} + \gamma \boldsymbol{max_a} \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s_{t+1}}, \boldsymbol{a}) - \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s_t}, \boldsymbol{a_t})\right]. \tag{2.15}$$

Algorithm 2 Q-Learning - An off-policy Temporal-Difference RL algorithm

Parameters: learning rate $\boldsymbol{\alpha} \in (\mathbf{0}, \mathbf{1}]$ and $\boldsymbol{\epsilon} > \mathbf{0}$.
Initialise $\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}\ \forall\ \boldsymbol{s} \in \boldsymbol{S},\ \boldsymbol{a} \in \boldsymbol{A}$. For terminal states initialise the value with 0.
for *each episode* **do**
    Initialise $\boldsymbol{s}$

    for *each step in episode* **do**
        Choose action $\boldsymbol{a}$ in $\boldsymbol{s}$ using $\boldsymbol{\pi}$ derived from $\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}$ (e.g. $\boldsymbol{\epsilon - greedy}$)
        Take action $\boldsymbol{a}$ and obtain reward $\boldsymbol{r}$ and next state $\boldsymbol{s'}$
        $\hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) = \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) + \boldsymbol{\alpha} \left[\boldsymbol{r} + \boldsymbol{\gamma} \cdot \arg\max_{\boldsymbol{a}} \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s'}, \boldsymbol{a}) - \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) \right]$
        Set $\boldsymbol{s} = \boldsymbol{s'}$
        Stop loop if $\boldsymbol{s}$ is terminal
    end
end

2.3.2.3 Value-function approximation

In sections 2.3.2.2 and 2.3.2.1 we discussed tabular algorithms for value-based RL. In this section we discuss function approximation in RL for estimating state-value functions from a known policy $\boldsymbol{\pi}$ (i.e. on-policy RL). The difference with the tabular approach is that we represent $\boldsymbol{v}_{\boldsymbol{\pi}}$ as a parameterised function with a weight vector $\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}}$ where $\hat{\boldsymbol{v}}(\boldsymbol{s}, \boldsymbol{w}) \approx \boldsymbol{v}_{\boldsymbol{\pi}}(\boldsymbol{s})$ is the approximated value of state $\boldsymbol{s}$ given the learned weights $\boldsymbol{w}$. Different function approximators can be used to estimate $\hat{\boldsymbol{v}}$. For instance, $\hat{\boldsymbol{v}}$ can be a deep neural network with $\boldsymbol{w}$ representing the weights of the network. In the tabular version of value-based RL, states and their estimated values are isolated from each other while in function approximation adjusting one weight in the network can lead to changes in the estimated values of many states. This form of learning is powerful due its ability to generalise across different states, but at the same time may lead to more complex models that are harder to understand and to tune. An example of value-function approximation is the deep Q-network (DQN) algorithm [241]. This algorithm combines deep (convolutional) neural network and Q-learning. Using DQN, it was shown that RL agents can achieve state-of-the-art performances on many problems without relying on engineered features. DNQ learns directly from raw (pixel) data instead. The following update rule is an alteration of the Q-learning (semi-gradient of Q-learning [334]) update rule for estimating the weights of the network:

$$\mathrm{w}_{t+1} = \boldsymbol{w_t} + \boldsymbol{\alpha}\left[\boldsymbol{r_{t+1}} + \boldsymbol{\gamma} \cdot \mathbf{max}_{\boldsymbol{a}}\, \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s_{t+1}}, \boldsymbol{a}, \boldsymbol{w_t}) - \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{w_t})\right] \boldsymbol{\nabla}_{\boldsymbol{wt}} \hat{\boldsymbol{Q}}_{\boldsymbol{\pi}}(\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{w_t}). \tag{2.16}$$

2.3.3 Policy-gradient RL

In value-based RL values of actions are approximated and then a policy is derived by selecting actions using a certain selection strategy. In policy-gradient RL we learn a parameterised policy directly [334; 335]. Consequently, we can select actions without the need for an explicit value function. Let $\boldsymbol{\Theta} \in \mathbb{R}^{\boldsymbol{d}}$ where $\boldsymbol{d}$ is the dimension of the parameter vector $\boldsymbol{\Theta}$. For policy-based methods that also rely on a value function, we denote the function's weight vector denoted by $\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d'}}$ as $\hat{\boldsymbol{v}}(\boldsymbol{s}, \boldsymbol{w})$. Define the probability of selecting action $\boldsymbol{a}$ at time step $\boldsymbol{t}$ given state $\boldsymbol{s}$ with policy parameters $\boldsymbol{\Theta}$ as:

$$\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{\Theta}) = \boldsymbol{P}[\boldsymbol{a_t} = \boldsymbol{a}|\boldsymbol{s_t} = \boldsymbol{s}, \boldsymbol{\Theta_t} = \boldsymbol{\Theta}] \tag{2.17}$$

Consider a function $\boldsymbol{J}(\boldsymbol{\Theta})$ that quantifies the performance of the policy $\boldsymbol{\pi}$ with respect to parameter vector $\boldsymbol{\Theta}$. The goal is to optimise $\boldsymbol{\Theta}$ such that $\boldsymbol{J}(\boldsymbol{\Theta})$ is maximised. We use the following update rule to approximate gradient ascent in $\boldsymbol{J}$ where the term $\boldsymbol{\nabla}\widehat{\boldsymbol{J}(\boldsymbol{\Theta_t})} \in \mathbb{R}^{\boldsymbol{d}}$ approximates the gradient of $\boldsymbol{J}(\boldsymbol{\Theta})$ at $\boldsymbol{t}$:

$$\boldsymbol{\Theta_{t+1}} = \boldsymbol{\Theta_t} + \boldsymbol{\alpha}\boldsymbol{\nabla}\widehat{\boldsymbol{J}(\boldsymbol{\Theta_t})}. \tag{2.18}$$

2.3.4 Actor-critic

In actor-critic methods [334; 182] both the value and policy functions are approximated. The actor in actor-critic is the learned policy while the critic approximates the value function. Algorithm 3 shows the one-step episodic actor-critic algorithm in more detail. The update rule for the parameter vector

Algorithm 3 One-step episodic actor-critic

Input: a differentiable policy $\boldsymbol{\pi(a|s,\Theta)}$
Input: a differentiable state-value function $\boldsymbol{\hat{v}(s,w)}$
Parameters: $\boldsymbol{\alpha(\Theta) > 0}$ and $\boldsymbol{\alpha(w) > 0}$ Initialise $\boldsymbol{\Theta} \in \mathbb{R}^{\boldsymbol{d}}$ and $\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d'}}$
for *each episode* **do**
  Initialise $\boldsymbol{S}$
  I = 1
  for *each step in episode* **do**
    Choose action $\boldsymbol{a}$ in $\boldsymbol{s}$ using $\boldsymbol{\pi}$: $\boldsymbol{a} \quad \boldsymbol{\pi(.|s,\Theta)}$
    Take action $\boldsymbol{a}$ and obtain reward $\boldsymbol{r}$ and next state $\boldsymbol{s'}$
    $\boldsymbol{\delta = r + \gamma\hat{v}(s',w) - \hat{v}(s,w)}$
    $\boldsymbol{w = w + \alpha(w)\delta\nabla\hat{v}(s,w)}$
    $\boldsymbol{\Theta = \Theta + \alpha(\Theta)I\delta\nabla \ln \pi(a|s,\Theta)}$
    $\boldsymbol{I = \gamma I}$
    $\boldsymbol{s = s'}$
  end
end

$\boldsymbol{\Theta}$ is defined as follows:

$$\boldsymbol{\Theta_{t+1} = \Theta_t + \alpha\delta_t \frac{\nabla\pi(a|s_t,\Theta_t)}{\pi(a|s_t,\Theta_t)}} \tag{2.19}$$

where $\boldsymbol{\delta_t}$ is defined as follows:

$$\boldsymbol{\delta_t = r_{t+1} + \gamma\hat{v}(s_{t+1},w) - \hat{v}(s_t,w).} \tag{2.20}$$

2.4 A classification of personalisation settings

Penalisation has many different definitions [296; 65; 109]. We adopt the definition proposed in [109] as it is based on 21 existing definitions found in literature and suits a variety of application domains: "personalization is a process that changes the functionality, interface, information access and content, or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals". This definition identifies personalisation as a process and mentions an existing system subject to that process. We include aspects of both the desired process of change and existing system in our framework. Section 2.5.4 further details how this framework was used in a SLR.

Table 2.1 provides an overview of the framework. On a high level, we distinguish three categories. The first category contains aspects of suitability of system behaviour. We differentiate settings in which suitability of system behaviour is determined explicitly by users and settings in which it is inferred by the system after observing user behaviour [309]. For example, a user can explicitly rate suitability of a video recommendation; a system can also infer suitability by observing whether the user decides to watch the video. Whether implicit or explicit feedback is preferable depends on availability and quality of feedback signals [309; 168]. Besides suitability, we consider safety of system behaviour. Unaltered RL algorithms use trial-and-error style exploration to optimise their behaviour yet this may not suit a particular domain. For example, tailoring the insulin delivery policy of an artificial pancreas to the metabolism of an individual requires trial insulin delivery action but these should only be sampled when their outcome is within safe certainty bounds [93]. If safety is a significant concern in the systems' application domain, specifically designed safety-aware RL techniques may be required, see [271] and [124] for overviews of such techniques.

Aspects in the second category deal with the availability of upfront knowledge. Firstly, knowledge of how users respond to system actions may be captured in user models. Such models open up a range of RL solutions that require less or no sampling of new interactions with users [154]. As an example, user pain models are used to predict suitability of exercises in an adaptive physical rehabilitation curriculum manager a priori[354]. Models can also be used to interact with the RL agent in simulation. For example, dialogue agent modules may be trained by interacting with a simulated chatbot user [97]. Secondly, upfront knowledge may be available in the form of data on human responses to system behaviour. This data can be used to derive user models and can be used to optimise policies directly and provide high-confidence evaluations of such policies [204; 349].

Table 2.1: *Framework to categorise personalisation settings by.*

Category	A#	Aspect	Description	Range
Suitability outcome	A1	Control	The extent to which the user defines the suitability of behaviour explicitly.	Explicit - implicit
	A2	Safety	The extent to which safety is of importance.	Trivial - critical
Upfront knowledge	A3	User models	The a priori availability of models that describe user responses to system behaviour.	Unavailable - unlimited
	A4	Data availability	The a priori availability of human responses to system behaviour.	Unavailable - unlimited
New Experiences	A5	Interaction availability	The availability of new samples of interactions with individuals.	Unavailable - unlimited
	A6	Privacy sensitivity	The degree to which privacy is a concern.	Trivial - critical
	A7	State observability	The degree to which all information to base personalisation can be measured.	Partial - full

The third category details new experiences. Empirical RL approaches have proven capable of modelling extremely complex dynamics, however, this typically requires complex estimators that in turn need substantial amounts of training data. The availability of users to interact with is therefore a major consideration when designing an RL solution. A second aspect that relates to the use of new experiences is privacy sensitivity of the setting. Privacy sensitivity is of importance as it may restrict sharing, pooling or any other specific usage of data [21]. Finally, we identify the state observability as a relevant aspect. In some settings, the true environment state cannot be observed directly but must be estimated using available observations. This may be common as personalisation exploits differences in mental [46; 368; 175] and physical state [128; 228]. For example, recommending appropriate music during running involves matching songs to the user emotional state and e.g. running pace. Both mental and physical state may be hard to measure accurately [37; 4; 274].

Although aspects in Table 2.1 are presented separately, we explicitly note that they are not mutually independent. Settings where privacy is a major concern, for example, are expected to typically have less existing and new interactions available. Similarly, safety requirements will impact new interaction availability. Presence of upfront knowledge is mostly of interest in settings where control lies with the system as it may ease the control task. In contrast, user models may be marginally important if desired behaviour is specified by the user in full. Finally, a lack of upfront knowledge and partial observability complicates adhering to safety requirements.

2.5 A systematic literature review

A SLR is 'a form of secondary study that uses a well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable' [48]. PRISMA is a standard for reporting on SLRs and details eligibility criteria, article collection, screening process, data extraction and data synthesis [246]. This section contains a report on this SLR according to the PRISMA statement.

This SLR was a collaborative work to which all authors contributed. We denote authors by abbreviation of their names, e.g. FDH, EG, AEH and MH.

2.5.1 Inclusion criteria

Studies in this SLR were included on the basis of three eligibility criteria. To be included, articles had to be published in a peer-reviewed journal or conference proceedings in English. Secondly, the study had to address a problem fitting to our definition of personalisation as described in Section 2.4. Finally, the study had to use a RL algorithm to address such a personalisation problem. Here, we view contextual bandit algorithms as a subset of RL algorithms and thus included them in our analysis. Additionally, we excluded studies in which a RL algorithm was used for purposes other than personalisation.

2.5.2 Search strategy

Figure 2.3 contains an overview of the SLR process. The first step is to run a query on a set of databases. For this SLR, a query was run on Scopus, IEEE Xplore, ACM's full-text collection, DBLP and Google Scholar on June 6, 2018. These databases were selected as their combined index spans a wide range, and their combined result set was sufficiently large for this study. Scopus and IEEE Xplore support queries on title, keywords and abstract. ACM's full-text collection, DBLP and Google scholar do not support queries on keywords and abstract content. We therefore ran two kinds of queries: we queried on title only for ACM's full-text collection, DBLP and Google Scholar and we extended this query to keywords and abstract content for Scopus and IEEE Xplore. The query was constructed by combining techniques of interest and keywords for the personalisation problem. For techniques of interest the terms 'reinforcement learning' and 'contextual bandits' were used. For the personalisation problem, variations on the words 'personalized', 'customized', 'individualized' and 'tailored' were included in British and American spelling. All queries are listed in Appendix 2.8. Query results were de-duplicated and stored in a spreadsheet.

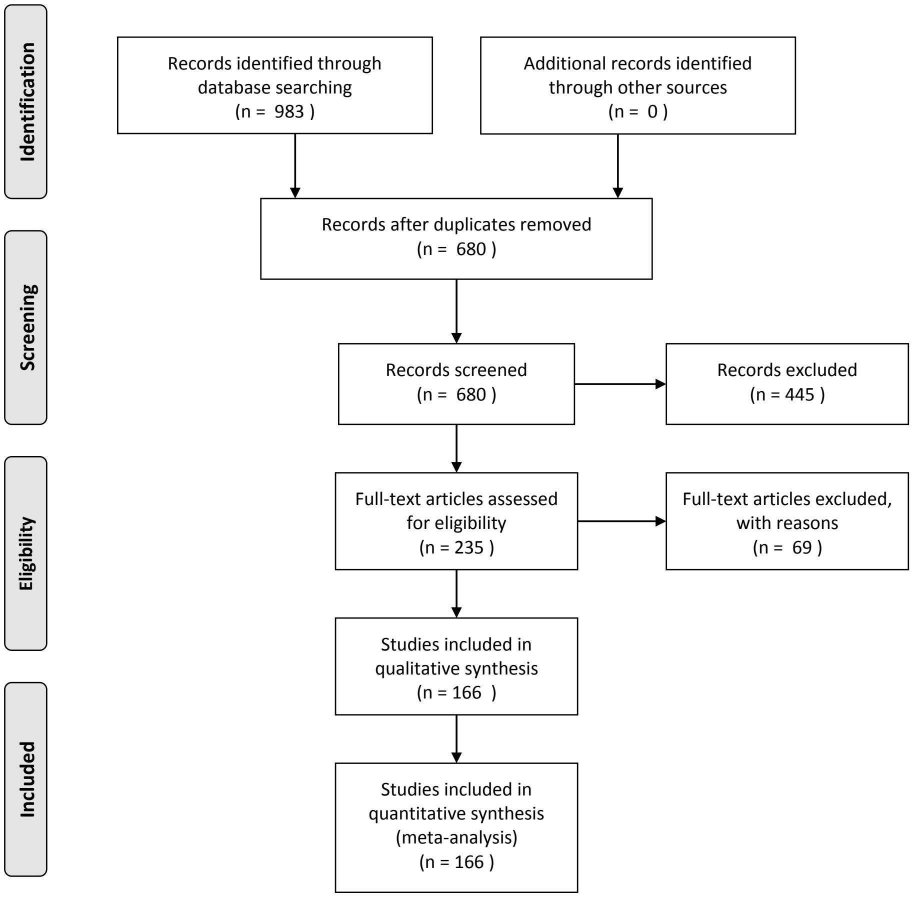


Figure 2.3: *Overview of the SLR process.*

2.5.3 Screening process

In the screening process, all query results are tested against the inclusion criteria from Section 2.5.1 in two phases. We used all criteria in both phases. In the first phase, we assessed eligibility based on keywords, abstract and title whereas we used full text of the article in the second phase. In the first phase, a spreadsheet with de-duplicated results was shared with all authors via Google Drive. Studies were assigned randomly to authors who scored each study by the eligibility criteria. The results of this screening were verified by one of the other authors, assigned randomly. Disagreements were settled in meetings involving those in disagreement and FDH if necessary. In addition to eligibility results, author preferences for full-text screening were recorded on a three-point scale. Studies that were not considered eligible were not taken into account beyond this point, all other studies were included in the second phase.

In the second phase, data on eligible studies was copied to a new spreadsheet. This sheet was again shared via Google Drive. Full texts were retrieved and evenly divided amongst authors according to preference. For each study, the assigned author then assessed eligibility based on full text and extracted the data items detailed below.

2.5.4 Data items

Data on setting, solution and methodology were collected. Table 2.2 contains all data items for this SLR. For data on setting, we operationalised our framework from Table 2.1 in Section 2.4. To assess trends in solution, algorithms used, number of MDP models (see Section 2.2) and training regime were recorded. Specifically, we noted whether training was performed by interacting with actual users ('live'), using existing data and a simulator of user behaviour. For the algorithms, we recorded the name as used by the authors. To gauge maturity of the proposed solutions and the field as a whole, data on the evaluation strategy and baselines used were extracted. Again, we listed whether evaluation included 'live' interaction with users, existing interactions between systems and users or using a simulator. Finally, publication year and application domain were registered to enable identification of trends over time and across domains. The list of domains was composed as follows: during phase one of the screening process, all authors recorded a domain for each included paper, yielding a highly inconsistent initial set of domains. This set was simplified into a more consistent set of domains which was used during full-text screening. For papers that did not fall into this consistent set of domains, two categories were added: a 'Domain Independent' and an 'Other' category. The actual domain was recorded for the five papers in the 'Other' category. These domains were not further consolidated as all five papers were assigned to unique domains not encountered before.

2.5.5 Synthesis and analysis

To facilitate analysis, reported algorithms were normalised using simple text normalisation and key-collision methods. The resulting mappings are available in the dataset release [96]. Data was summarised using descriptive statistics and figures with an accompanying narrative to gain insight into trends with respect to settings, solutions and evaluation over time and across domains.

2.6 Results

The quantitative synthesis and analyses introduced in Section 2.5.5 were applied to the collected data. In this section, we present insights obtained. We focus on the major insights and encourage the reader to explore the tabular view in Appendix 2.9 or the collected data for further analysis [96].

Before diving into the details of the study in light of the classification scheme we have proposed, let us first study some general trends. Figure 2.4 shows the number of publications addressing personalisation using RL techniques over time. A clear increase can be seen. With over forty entries, the health domain contains by far the most articles, followed by entertainment, education and commerce with all approximately just over twenty five entries. Other domains contain less than twelve papers in total. Figure 2.5a shows the popularity of

Table 2.2: *Data items in SLR. The last column relates data items to aspects of setting from Table 2.1 where applicable.*

Category	#	Data item	Values	A#
Setting	1	User defines suitability of system behaviour explicitly	Yes, No	A1
	2	Suitability of system behaviour is derived	Yes, No	A1
	3	Safety is mentioned as a concern in the article	Yes, No	A2
	4	Privacy is mentioned as a concern in the article	Yes, No	A6
	5	Models of user responses to system behaviour are available	Yes, No	A3
	6	Data on user responses to system behaviour are available	Yes. No	A4
	7	New interactions with users can be sampled with ease	Yes, No	A5
	8	All information to base personalisation on can be measured	Yes, No	A7
Solution	9	Algorithms	N/A	–
	10	Number of learners	1, 1/user, 1/group, multiple	–
	11	Usage of traits of the user	state, other, not used	–
	12	Training mode	online, batch, other, unknown	–
	13	Training in simulation	Yes, No	A3
	14	Training on a real-life dataset	Yes, No	A4
	15	Training in 'live' setting	Yes, No	A5
Evaluation	16	Evaluation in simulation	Yes, No	A3
	17	Evaluation on a real-life dataset	Yes, No	A4
	18	Evaluation in 'live' setting	Yes, No	A5
	19	Comparison with 'no personalisation'	Yes, No	–
	20	Comparison with non-RL methods	Yes, No	–

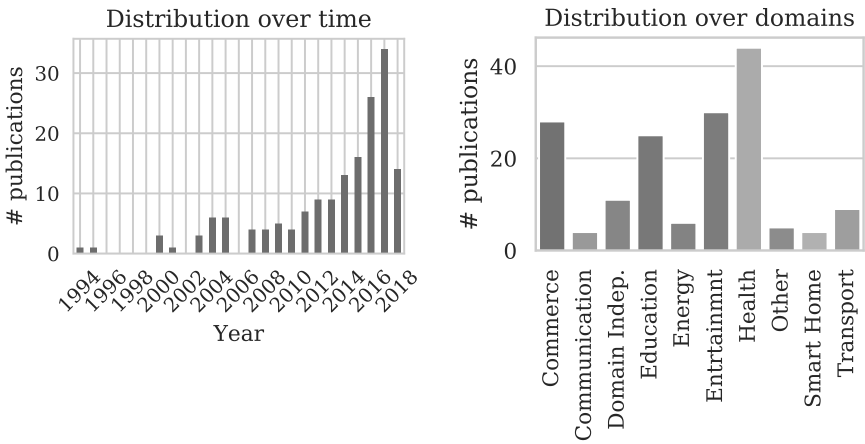


Figure 2.4: *Distribution of included papers over time and over domains. Note that only studies published prior to the query date of June 6, 2018 were included.*

domains for the five most recent years and seems to indicate that the number of articles in the health domain is steadily growing, in contrast with the other domains. Of course, these graphs are based on a limited number of publications, so drawing strong conclusions from these results is difficult. We do need to take into account that the popularity of RL for personalisation is increasing in general. Therefore Figure 2.5b shows the relative distribution of studies over domains for the five most recent years. Now we see that the health domain is just following the overall trend, and is not becoming more popular within studies that use RL for personalisation. We fail to identify clear trends for other domains from these figures.

2.6.1 Setting

Table 2.3 provides an overview of the data related to setting in which the studies were conducted. The table shows that user responses to system behaviour are present in a minority of cases (66/166). Additionally, models of user behaviour are only used in around one quarter of all publications. The suitability of system

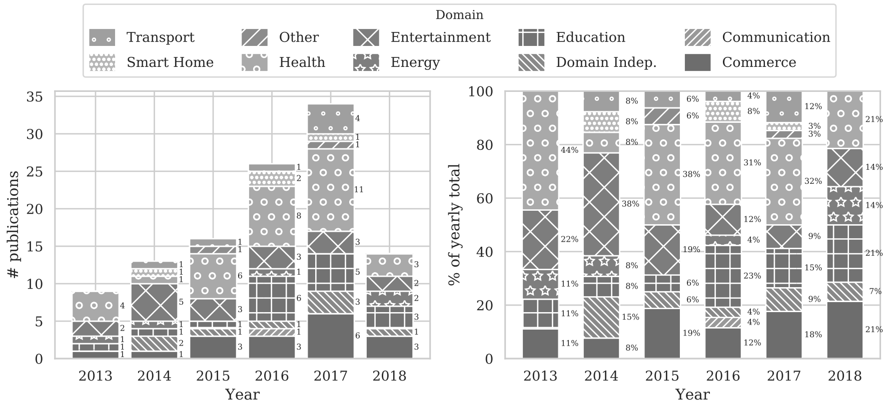


Figure 2.5: *Popularity of domains for the five most recent years.*

behaviour is much more frequently derived from data (130/166) rather than explicitly collected by users (39/166). Privacy is clearly not within the scope of most articles, only in 9 out of 166 cases do we see this issue explicitly mentioned. Safety concerns, however, are mentioned in a reasonable proportion of studies (30/166). Interactions can generally be sampled with ease and the resulting information is frequently sufficient to base personalisation of the system at hand on.

Table 2.3: *Number of Publications by aspects of setting.*

Aspect	#
User defines suitability of system behaviour explicitly	39
Suitability of system behaviour is derived	130
Safety is mentioned as a concern in the article	30
Privacy is mentioned as a concern in the article	9
Models of user responses to system behaviour are available	41
Data on user responses to system behaviour are available	66
New interactions with users can be sampled with ease	97
All information to base personalisation on can be measured	132

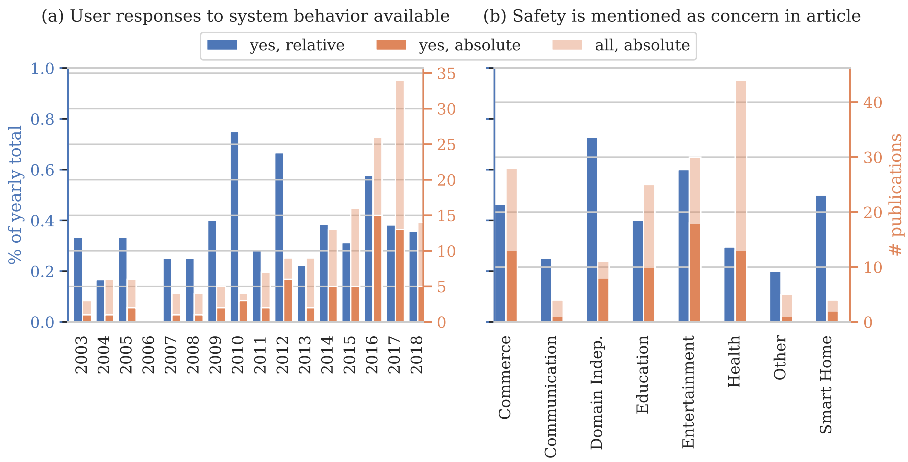


Figure 2.6: *Availability of user responses over time (a), and mentions of safety as a concern over domains (b).*

Let us dive into some aspects in a bit more detail. A first trend we anticipate is an increase of the fraction of studies working with real data on human responses over the years, considering the digitisation trend and associated data collection. Figure 2.6a shows the fraction of papers for which data on user responses to system behaviour is available over time. Surprisingly, we see that this fraction does not show any clear trend over time. Another aspect of interest relates to safety issues in particular domains. We hypothesise that in certain domains, such as health, safety is more frequently mentioned as a concern. Figure 2.6b shows the fraction of papers of the different domains in which safety is mentioned. Indeed, we clearly see that certain domains mention safety much more frequently than other domains. Third, we explore the ease with which interactions with users can be sampled. Again, we expect to see substantial differences between domains. Figure 2.7 confirms our intuition. Interactions can be sampled with ease more frequently in studies in the commerce, entertainment, energy, and smart homes domains when compared to communication and health domains.

Finally, we investigate whether upfront knowledge is available. In our analysis, we explore both real data as as well user models being available upfront. One would expect papers to have at least one of these two prior to starting experiments. User models and not real data were reported in 41 studies, while 53 articles used real data but no user model and 12 use both. We see that for 71 studies neither is available. In roughly half of these, simulators were used for both training (38/71) and evaluation (37/71). In a minority, training (15/71) and evaluation (17/71) were performed in a live setting, e.g. while collecting data.

2.6.2 Solution

In our investigation into solutions, we first explore the algorithms that were used. Figure 2.8 shows the distribution of usage frequency. A vast majority of the algorithms are used only once, some techniques are used a couple of times and one algorithm is used 60 times. Note again that we use the name of the algorithms used by the authors as a basis for this analysis. Table 2.4 lists the algorithms that were used more than once. A significant number of studies (60/166) use the Q-learning algorithm. At the same time, a substantial number of articles (18/166) reports the use of RL as the underlying algorithmic framework without specifying an actual algorithm. The contextual bandits, Sarsa, actor-critic and inverse RL (IRL) algorithms are used in respectively (18/166), (12/166), (8/166), (8/166) and (7/166) papers. We also observe some additional algorithms from the contextual bandits family, such as UCB and LinUCB. Furthermore, we find various mentions that indicate the usage of deep neural networks: deep reinforcement learning, DQN and DDQN. In general, we find that some publications refer to a specific algorithm whereas others only report generic techniques or families thereof.

Figure 2.9a lists the number of models used in the included publications. The majority of solutions relies on a single-model architecture. On the other end of the spectrum lies the architecture of using one model per person. This architecture comes second in usage frequency. The architecture that uses one model per group can be considered a middle ground between these former two.

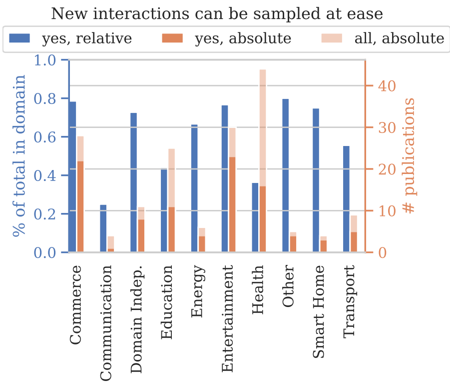


Figure 2.7: *New interactions with users can be sampled with ease.*

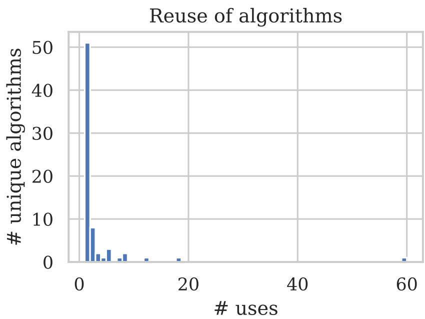


Figure 2.8: *Distribution of algorithm usage frequencies.*

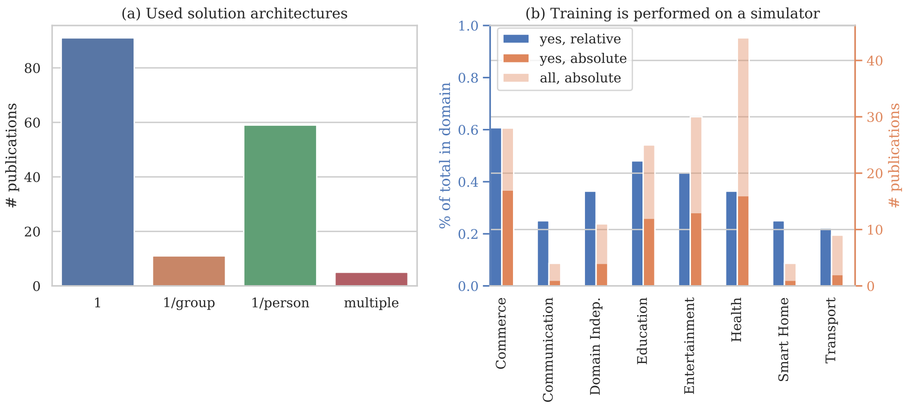


Figure 2.9: *Occurrence of different solution architectures (a) and usage of simulators in training (b). For (a), publications that compare architectures are represented in the 'multiple' category.*

Table 2.4: *Algorithm usage for all algorithms that were used in more than one publication.*

Algorithm	# of uses
Q-learning [370]	60
RL, not further specified	18
Contextual bandits	12
Sarsa [332]	8
Actor-critic	8
Inverse reinforcement learning	7
UCB [19]	5
Policy iteration	5
LinUCB [77]	5
Deep reinforcement learning	4
Fitted Q-iteration [297]	3
DQN [241]	3
Interactive reinforcement learning	2
TD-learning	2
DYNA-Q [331]	2
Policy gradient	2
CLUB [129]	2
Monte carlo	2
Thompson sampling	2
DDQN [359]	2

In this architecture, only experiences with relevant individuals can be shared. Comparisons between architectures are rare. We continue by investigating whether and where traits of the individual were used in relation to these architectures. Table 2.5 provides an overview. Out of all papers that use one model, 52.7% did not use the traits of the individuals and 41.7 % included traits in the state space. 47.5% of the papers include the traits of the individuals in the state representation while in 37.3% of the papers the traits were not included. In 15.3% of the cases this was not known.

Figure 2.9b shows the popularity of using a simulator for training per domain. We see that a substantial percentage of publications use a simulator and that simulators are used in all domains. Simulators are used in the majority

Table 2.5: *Number of models and the inclusion of user traits.*

	Number of models			
Traits of users were used	1	1/group	1/person	multiple
In state representation	38	8	28	2
Other	5	0	9	3
Not used	48	3	22	0
Total	91	11	59	5

of publications for the energy, transport, communication and entertainment domains. In publications in the first three out of these domains, we typically find applications that require large-scale implementation and have a big impact on infrastructure, e.g. control of the entire energy grid or a fleet of taxis in a large city. This complicates the collection of useful realistic dataset and training in a live setting. This is not the case for the entertainment domain with 17 works using a simulator for training. Further investigation shows that nine out of these 17 also include training on real data or in a 'live' setting. It seems that training on a simulator is part of the validation of the algorithm rather than the prime contribution of the paper in the entertainment domain.

2.6.3 Evaluation

In investigating evaluation rigour, we first turn to the data on which evaluations are based. Figure 2.10 shows how many studies include an evaluation in a 'live' setting or using existing interactions with users. In the years up to 2007 few studies were done and most of these included realistic evaluations. In more recent years, the absolute number of studies shows a marked upward trend to which the relative number of articles that include a realistic evaluation fails to keep pace. Figure 2.10 also shows the number of realistic evaluations per domain. Disregarding the smart home domain, as it contains only four studies, the highest ratio of real evaluations can be found in the commerce and entertainment domains, followed by the health domain.

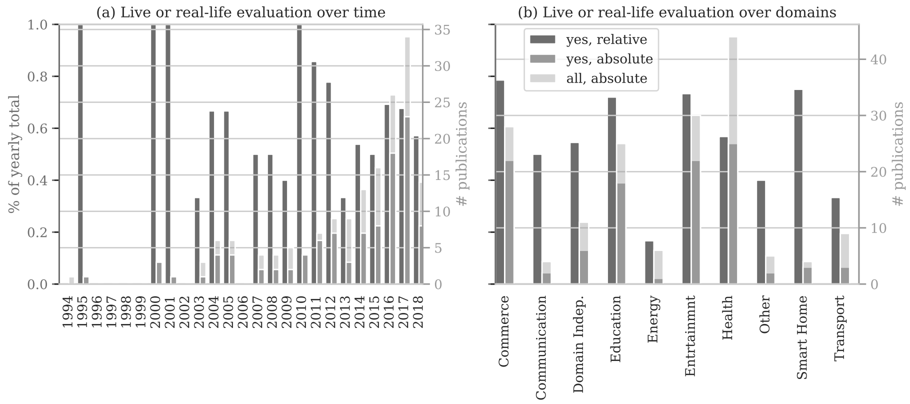


Figure 2.10: *Number of papers with a 'live' evaluation or evaluation using data on user responses to system behaviour.*

We look at possible reasons for a lack of realistic evaluation using our categorisation of settings from Section 2.4. Indeed, there are 63 studies with no realistic evaluation versus 104 with a realistic evaluation. Because these group sizes differ, we include ratios with respect to these totals in Table 2.6. The biggest difference between ratios of studies with and without a realistic evaluation is in the upfront availability of data on interactions with users. This is not surprising, as it is natural to use existing interactions for evaluation when they are available already. The second biggest difference between the groups is whether safety is mentioned as a concern. Relatively, studies that refrain from a realistic evaluation mention safety concerns almost twice as often as studies that do a realistic evaluation. The third biggest difference can be found in availability of user models. If a model is available, user responses can be simulated more easily. Privacy concerns are not mentioned frequently, so little can be said on its contribution to a lacking realistic evaluation. Finally and surprisingly, the ease of sampling interactions is comparable between studies with a realistic and without realistic evaluation.

Figure 2.11 describes how many studies include any of the comparisons in scope in this survey, that is: comparisons between solutions with and without

Table 2.6: *Comparison of settings with realistic and other evaluation.*

	Real-world evaluation		Other evaluation	
	Count	% of column total	Count	% of column total
Total	104	100.0%	63	100.0%
Data on user responses to system behaviour are available	57	54.8%	9	14.5%
Safety is mentioned as a concern in the article	14	13.5%	16	25.8%
Models of user responses to system behaviour are available	21	20.2%	20	32.3%
Privacy is mentioned as a concern in the article	7	6.7%	2	3.2%
New interactions with users can be sampled with ease	60	57.7%	37	59.7%

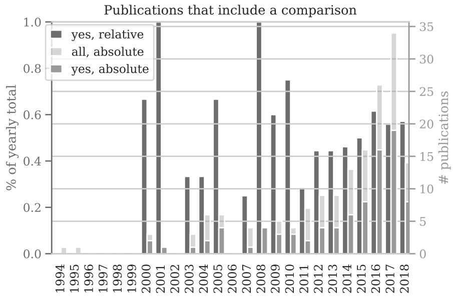


Figure 2.11: *Number of papers that include any comparison between solutions over time.*

personalisation, comparisons between RL approaches and other approaches to personalisation and comparisons between different RL algorithms. In the first years, no papers includes such a comparison. The period 2000-2010 contains relatively little studies in general and the absolute and relative numbers of studies with a comparison vary. From 2011 to 2018, the absolute number maintains it upward trend. The relative number follows this trend but flattens after 2016.

2.7 Discussion

The goal of this study was to give an overview and categorisation of RL applications for personalisation in different application domains which we addressed using a SLR on settings, solution architectures and evaluation strategies. The main result is the marked increase in studies that use RL for personalisation problems over time. Additionally, techniques are increasingly evaluated on real-life data. RL has proven a suitable paradigm for adaptation of systems to individual preferences using data.

Results further indicate that this development is driven by various techniques, which we list in no particular order. Firstly, techniques have been developed to estimate the performance of deploying a particular RL model prior to deployment. This helps in communicating risks and benefits of RL solutions with stakeholders and moves RL further into the realm of feasible technologies for high-impact application domains [348]. For single-step decision making problems, contextual bandit algorithms with theoretical bounds on decision-theoretic regret have become available. For multi-step decision making problems, methods that can estimate the performance of some policy based on data generated by another policy have been developed [77; 349; 169]. Secondly, advances in the field of deep learning have wholly or partly removed the need for feature engineering [106]. This may be especially challenging for sequential decision-making problems as different features may be of importance in different states encountered over time. Finally, research on safe exploration in RL has developed means to avoid harmful actions during exploratory phases of learning [124]. How any these techniques are best applied depends on setting. The collected data can be used to find suitable related work for any particular setting [96].

Since the field of RL for personalisation is growing in size, we investigated whether methodological maturity is keeping pace. Results show that the growth in the *number* of studies with a real-life evaluation is not mirrored by growth of the *ratio* of studies with such an evaluation. Similarly, results show no increase in the relative number of studies with a comparison of approaches over time. These may be signs that the maturity of the field fails to keep pace with its growth. This is worrisome, since the advantages of RL over other approaches or between RL algorithms cannot be understood properly without such comparisons. Such comparisons benefit from standardised tasks. Developing standardised personalisation datasets and simulation environments is an excellent opportunity for future research [206; 164].

We found that algorithms presented in literature are reused infrequently. Although this phenomenon may be driven by various different underlying dynamics that cannot be untangled using our data, we propose some possible explanations here without particular order. Firstly, it might be the case that separate applications require tailored algorithms to the extend that these can only be used once. This raises the question on the scientific contribution of such a tailored algorithm and does not fit with the reuse of some well-established algorithms. Another explanation is that top-ranked venues prefer contributions that are theoretical or technical in nature, resulting in minor variations to well-known algorithms being presented as novel. Whether this is the case is out of scope for this research and forms an excellent avenue for future work. A final explanation for us to propose, is the myriad axes along which any RL algorithm can be identified, such as whether and where estimation is involved, which estimation technique is used and how domain knowledge is encoded in the algorithm. This may yield a large number of unique algorithms, constructed out of a relatively small set of core ideas in RL. An overview of these core ideas would be useful in understanding how individual algorithms relate to each other.

On top of algorithm reuse, we analysed which RL algorithms were used most frequently. Generic and well-established (families of) algorithms such as Q-learning are the most popular. A notable entry in the top six most-used techniques is inverse reinforcement learning (IRL). Its frequent usage is surprising, as the only viable application area of IRL under a decade ago was robotics [181]. Personalisation may be one of the other useful application areas of this branch of RL and many existing personalisation challenges may still benefit from an IRL approach. Finally, we investigated how many RL models were included in the proposed solutions and found that the majority of studies resorts to using either one RL model in total or one RL model per user. Inspired by common practice of clustering in the related fields such as e.g. recommender systems, we believe that there exists opportunities in pooling data of similar users and training RL models on the pooled data. We are going to be exploring this idea further in the next chapter of this thesis, where we answer ***T.RQ*1**.

Besides these findings, we contribute a categorisation of personalisation settings in RL. This framework can be used to find related work based on the setting of a problem at hand. In designing such a framework, one has to balance specificity and usefulness of aspects in the framework. We take the aspect of 'safety' as an example: any application of RL will imply safety concerns at some level, but they are more prominent in some application areas. The framework intentionally includes a single ambiguous aspect to describe a broad range 'safety sensitivity levels' in order for it to suit its purpose of navigating literature. A possibility for future work is to extend the framework with other, more formal, aspects of problem setting such as those identified in [304].

2.8 Appendix A. Queries

Listing 2.1: *Query for Scopus Database*

```
TITLE-ABS-KEY(
("reinforcement learning" OR "contextual bandit") AND
("personalization" OR "personalized" OR "personal" OR "
    personalisation" OR "personalised" OR
"customization" OR "customized" OR "customised" OR "
    customised" OR
"individualized" OR "individualised" OR "tailored"))
```

Listing 2.2: *Query for IEEE Xplore Database Command Search*

```
(((reinforcement learning) OR contextual bandit) AND
(personalization OR personalized OR personal OR
    personalisation OR personalised OR
customization OR customized OR customised OR customised
    OR
individualized OR individualised OR tailored))
```

Listing 2.3: *Query for ACM DL Database*

```
("reinforcement learning" OR "contextual bandit") AND
(personalization OR personalized OR personal OR
    personalisation OR personalised OR
customization OR customized OR customised OR customised
    OR
individualized OR individualised OR tailored)
```

Listing 2.4: *First Query for DBLP Database*

```
reinforcement learning
(personalization|personalized|personal|personalisation|
    personalised|
customization|customized|customised|customised|
individualized|individualised|tailored)
```

Listing 2.5: *Second Query for DBLP Database*

```
contextual bandit
(personalization|personalized|personal|personalisation|
    personalised|
customization|customized|customised|customised|
individualized|individualised|tailored)
```

Listing 2.6: *First Query for Google Scholar Database*

```
allintitle: "reinforcement learning"
personalization OR personalized OR personal OR
    personalisation OR personalised OR
customization OR customized OR customised OR customised
    OR
individualized OR individualised OR tailored
```

Listing 2.7: *Second Query for Google Scholar Database*

```
allintitle: "contextual bandit"
personalization OR personalized OR personal OR
    personalisation OR personalised OR
customization OR customized OR customised OR customised
    OR
individualized OR individualised OR tailored
```

2.9 Appendix B. Tabular view of data

Table 2.7: *Table containing all included publications. The first column refers to the data items in Table 2.2.*

#	Value	Publications
1	n	[3; 10; 22; 24; 31; 36; 39; 41; 42; 49; 58; 60; 61; 62; 63; 66; 69; 74; 75; 78; 87; 90; 89; 88; 93; 94; 98; 99; 105; 112; 123; 126; 127; 130; 137; 155; 161; 162; 167; 171; 172; 173; 183; 184; 193; 197; 199; 200; 203; 207; 212; 213; 217; 221; 225; 226; 227; 229; 230; 231; 232; 233; 235; 236; 238; 242; 247; 248; 256; 258; 259; 264; 265; 268; 270; 279; 280; 282; 283; 285; 287; 292; 298; 303; 305; 310; 313; 314; 317; 326; 327; 330; 329; 338; 339; 341; 342; 343; 344; 346; 348; 347; 351; 352; 353; 357; 366; 365; 369; 382; 383; 384; 385; 386; 389; 392; 395; 396; 401; 399; 398; 400; 402; 403; 404; 405; 407]
	y	[6; 17; 20; 72; 73; 104; 113; 116; 114; 117; 115; 121; 135; 136; 151; 156; 191; 202; 209; 220; 260; 272; 273; 289; 299; 316; 324; 327; 354; 355; 356; 360; 364; 367; 388; 390; 391; 397]
2	n	[6; 17; 22; 31; 49; 62; 75; 78; 87; 94; 99; 104; 113; 115; 135; 136; 191; 202; 209; 230; 233; 236; 260; 268; 270; 272; 285; 292; 327; 360; 364; 367; 388; 389; 390; 397]
	y	[3; 10; 20; 24; 36; 39; 41; 42; 58; 60; 61; 63; 66; 69; 72; 73; 74; 90; 89; 88; 93; 98; 105; 112; 116; 114; 117; 121; 123; 126; 127; 130; 137; 151; 155; 156; 161; 162; 167; 171; 172; 173; 183; 184; 193; 197; 199; 200; 203; 207; 212; 213; 217; 220; 221; 225; 226; 227; 229; 231; 232; 235; 238; 242; 247; 248; 256; 258; 259; 264; 265; 273; 279; 280; 282; 283; 287; 289; 298; 299; 303; 305; 310; 313; 314; 316; 317; 324; 326; 327; 330; 329; 338; 339; 341; 342; 343; 344; 346; 348; 347; 351; 352; 353; 354; 355; 356; 357; 366; 365; 369; 382; 383; 384; 385; 386; 391; 392; 395; 396; 401; 399; 398; 400; 402; 403; 404; 405; 407]

#	Value	Publications
3	n	[3; 6; 10; 17; 20; 22; 24; 31; 36; 39; 41; 58; 60; 61; 62; 63; 66; 72; 73; 74; 75; 78; 87; 88; 98; 99; 104; 105; 112; 113; 116; 114; 117; 115; 123; 127; 130; 135; 136; 137; 155; 156; 162; 167; 171; 172; 173; 183; 191; 193; 199; 200; 202; 203; 209; 212; 217; 221; 225; 226; 227; 229; 231; 232; 233; 235; 236; 242; 247; 248; 256; 259; 260; 264; 265; 268; 272; 273; 279; 282; 283; 285; 287; 289; 292; 298; 305; 310; 313; 314; 316; 317; 324; 326; 327; 330; 329; 338; 339; 341; 342; 343; 344; 346; 347; 351; 353; 357; 364; 366; 365; 367; 369; 382; 383; 384; 385; 386; 388; 389; 390; 391; 392; 395; 397; 401; 399; 398; 400; 402; 403; 404; 405; 407]
	y	[42; 49; 69; 90; 89; 93; 94; 121; 126; 151; 161; 184; 197; 207; 213; 220; 230; 238; 258; 270; 280; 299; 303; 348; 352; 354; 355; 356; 360; 396]
4	n	[6; 10; 17; 22; 24; 31; 36; 39; 41; 42; 49; 58; 60; 61; 62; 63; 66; 69; 72; 73; 74; 75; 78; 87; 90; 89; 88; 93; 94; 98; 99; 104; 105; 112; 113; 116; 114; 117; 115; 121; 123; 126; 127; 130; 135; 137; 151; 155; 156; 161; 162; 171; 172; 173; 183; 184; 191; 193; 197; 200; 202; 203; 207; 209; 212; 213; 217; 220; 221; 225; 226; 227; 229; 230; 231; 232; 233; 235; 236; 238; 242; 247; 248; 256; 258; 259; 260; 264; 265; 268; 270; 272; 273; 279; 280; 282; 283; 285; 287; 289; 292; 298; 299; 303; 310; 313; 314; 316; 317; 324; 326; 327; 330; 329; 338; 339; 341; 342; 343; 344; 346; 348; 347; 351; 352; 353; 354; 356; 357; 360; 364; 366; 365; 369; 382; 383; 384; 385; 386; 388; 389; 390; 391; 392; 395; 396; 397; 401; 399; 398; 400; 402; 403; 405; 407; 20]
	y	[3; 136; 167; 199; 305; 327; 355; 367; 404]

#	Value	Publications
5	n	[3; 6; 17; 22; 24; 31; 41; 49; 58; 60; 61; 62; 63; 66; 72; 73; 74; 78; 87; 88; 99; 104; 105; 112; 113; 121; 123; 130; 135; 137; 151; 155; 161; 162; 167; 171; 172; 173; 183; 191; 193; 197; 199; 200; 202; 203; 207; 209; 212; 213; 217; 220; 225; 226; 231; 232; 233; 236; 238; 242; 247; 248; 256; 260; 264; 265; 268; 270; 273; 279; 280; 283; 285; 287; 289; 292; 298; 299; 303; 305; 310; 313; 314; 317; 324; 326; 327; 329; 338; 339; 342; 343; 344; 348; 347; 351; 352; 353; 355; 356; 360; 364; 366; 365; 367; 369; 382; 383; 384; 389; 390; 392; 395; 397; 401; 399; 398; 400; 402; 403; 404; 405; 407]
	y	[10; 20; 36; 39; 42; 69; 75; 90; 89; 93; 94; 98; 116; 114; 117; 115; 126; 127; 136; 156; 184; 221; 227; 229; 230; 235; 258; 259; 272; 282; 316; 330; 341; 346; 354; 357; 385; 386; 388; 391; 396]
6	n	[3; 6; 10; 20; 22; 31; 36; 39; 49; 60; 61; 62; 66; 69; 72; 78; 87; 89; 88; 93; 94; 98; 99; 104; 105; 112; 113; 115; 121; 126; 127; 130; 151; 155; 156; 161; 162; 171; 172; 173; 183; 184; 193; 197; 202; 213; 217; 220; 221; 225; 226; 227; 229; 230; 232; 235; 236; 238; 248; 256; 258; 259; 264; 265; 268; 272; 273; 280; 282; 285; 292; 298; 303; 305; 313; 316; 317; 326; 327; 330; 329; 339; 351; 352; 353; 354; 356; 357; 364; 369; 382; 383; 392; 396; 401; 399; 398; 400; 402; 407]
	y	[17; 24; 41; 42; 58; 63; 73; 74; 75; 90; 116; 114; 117; 123; 135; 136; 137; 167; 191; 199; 200; 203; 207; 209; 212; 231; 233; 242; 247; 260; 270; 273; 279; 283; 287; 289; 299; 310; 314; 324; 327; 338; 341; 342; 343; 344; 346; 348; 347; 355; 360; 366; 365; 367; 384; 385; 386; 388; 389; 390; 391; 395; 397; 403; 404; 405]
7	n	[3; 24; 31; 36; 61; 62; 63; 66; 69; 74; 75; 78; 87; 90; 98; 99; 104; 105; 113; 123; 126; 130; 137; 151; 173; 184; 191; 193; 197; 202; 207; 213; 225; 226; 229; 230; 236; 238; 247; 256; 259; 260; 268; 272; 279; 280; 282; 285; 292; 299; 305; 310; 317; 324; 326; 327; 351; 352; 353; 355; 356; 360; 369; 390; 392; 396; 399; 398; 400]

#	Value	Publications
	y	[6; 10; 17; 20; 22; 39; 41; 42; 49; 58; 60; 72; 73; 89; 88; 93; 94; 112; 116; 114; 117; 115; 121; 127; 135; 136; 155; 156; 161; 162; 167; 171; 172; 183; 199; 200; 203; 209; 212; 217; 220; 221; 227; 231; 232; 233; 235; 242; 248; 258; 264; 265; 270; 273; 283; 287; 289; 298; 303; 313; 314; 316; 327; 330; 329; 338; 339; 341; 342; 343; 344; 346; 348; 347; 354; 357; 364; 366; 365; 367; 382; 383; 384; 385; 386; 388; 389; 391; 395; 397; 401; 402; 403; 404; 405; 407]
8	n	[42; 58; 62; 66; 73; 75; 112; 121; 126; 135; 167; 199; 209; 212; 220; 225; 242; 264; 270; 285; 298; 338; 342; 347; 356; 366; 365; 367; 369; 383; 391; 395; 402; 404]
	y	[3; 6; 10; 17; 20; 22; 24; 31; 36; 39; 41; 49; 60; 61; 63; 69; 72; 74; 78; 87; 90; 89; 88; 93; 94; 98; 99; 104; 105; 113; 116; 114; 117; 115; 123; 127; 130; 136; 137; 151; 155; 156; 161; 162; 171; 172; 173; 183; 184; 191; 193; 197; 200; 202; 203; 207; 213; 217; 221; 226; 227; 229; 230; 231; 232; 233; 235; 236; 238; 247; 248; 256; 258; 259; 260; 265; 268; 272; 273; 279; 280; 282; 283; 287; 289; 292; 299; 303; 305; 310; 313; 314; 316; 317; 324; 326; 327; 330; 329; 339; 341; 343; 344; 346; 348; 351; 352; 353; 354; 355; 357; 360; 364; 382; 384; 385; 386; 388; 389; 390; 392; 396; 397; 401; 399; 398; 400; 403; 405; 407]
10	1	[3; 10; 22; 24; 39; 60; 61; 62; 63; 66; 73; 74; 78; 87; 90; 89; 94; 98; 105; 112; 123; 126; 130; 155; 156; 161; 162; 167; 172; 173; 184; 191; 193; 199; 202; 203; 207; 209; 225; 229; 230; 231; 238; 247; 258; 259; 264; 268; 282; 283; 285; 287; 303; 305; 310; 316; 317; 326; 327; 330; 329; 338; 339; 341; 342; 343; 344; 348; 347; 352; 353; 354; 355; 357; 360; 364; 366; 365; 367; 384; 385; 386; 388; 390; 392; 396; 399; 398; 402; 405; 407]
	1/group	[36; 88; 200; 212; 217; 226; 227; 279; 369; 382; 400]

#	Value	Publications
	1/person	[17; 20; 31; 41; 42; 49; 58; 69; 72; 75; 93; 104; 113; 116; 114; 117; 115; 121; 127; 135; 136; 137; 151; 171; 183; 197; 213; 220; 221; 232; 233; 235; 236; 248; 256; 260; 265; 270; 272; 273; 280; 289; 292; 298; 299; 313; 314; 324; 346; 351; 356; 383; 389; 391; 395; 397; 401; 403]
	multiple	[6; 99; 242; 327; 404]
11	not used	[17; 20; 22; 31; 36; 39; 49; 60; 62; 66; 72; 73; 74; 78; 87; 90; 88; 93; 94; 105; 113; 115; 123; 127; 130; 135; 155; 156; 161; 167; 172; 173; 183; 191; 193; 202; 209; 212; 220; 225; 231; 232; 238; 247; 248; 264; 265; 268; 273; 285; 299; 303; 310; 317; 326; 327; 330; 338; 339; 346; 348; 351; 357; 360; 364; 367; 383; 384; 388; 390; 401; 403; 405]
	other	[6; 41; 58; 136; 137; 235; 236; 242; 272; 280; 283; 289; 342; 344; 392; 396; 404]
	state representation	[3; 10; 24; 42; 61; 63; 69; 75; 89; 98; 99; 104; 112; 116; 114; 117; 121; 126; 151; 162; 171; 184; 197; 199; 200; 203; 207; 213; 217; 221; 226; 227; 229; 230; 233; 256; 258; 259; 260; 270; 273; 279; 282; 287; 292; 298; 305; 313; 314; 316; 324; 327; 329; 341; 343; 347; 352; 353; 354; 355; 356; 366; 365; 369; 382; 385; 386; 389; 391; 395; 397; 399; 398; 400; 402; 407]
12	batch	[3; 73; 74; 87; 99; 104; 112; 116; 114; 117; 115; 136; 167; 184; 191; 200; 213; 225; 226; 227; 229; 230; 233; 258; 259; 272; 282; 283; 292; 324; 327; 330; 329; 338; 339; 343; 344; 346; 348; 347; 352; 366; 365; 384; 391; 392; 399; 398; 400; 402; 404; 407]
	n	[383]
	online	[17; 20; 39; 41; 42; 49; 58; 60; 63; 69; 72; 75; 89; 94; 98; 113; 121; 126; 127; 130; 135; 151; 162; 171; 172; 173; 183; 197; 199; 202; 203; 207; 209; 212; 220; 221; 232; 235; 236; 238; 247; 248; 260; 264; 265; 270; 273; 289; 298; 299; 303; 305; 310; 313; 314; 316; 326; 341; 342; 354; 356; 357; 360; 367; 385; 386; 388; 389; 390; 397; 401; 403; 405]
	other	[93; 137; 155; 242; 256; 279; 369; 395]

#	Value	Publications
	unknown	[6; 10; 22; 24; 31; 36; 61; 62; 66; 78; 90; 88; 105; 123; 156; 161; 193; 217; 231; 268; 280; 285; 287; 317; 327; 351; 353; 355; 364; 382; 396]
13	n	[3; 6; 17; 24; 41; 60; 61; 62; 63; 66; 72; 73; 74; 75; 87; 90; 99; 105; 123; 137; 167; 172; 173; 184; 191; 193; 199; 200; 202; 203; 207; 209; 217; 225; 226; 229; 230; 231; 232; 233; 238; 247; 258; 259; 260; 264; 265; 268; 273; 279; 282; 283; 287; 289; 292; 305; 310; 313; 314; 317; 324; 326; 327; 343; 344; 346; 348; 347; 353; 355; 360; 364; 365; 382; 383; 384; 385; 386; 388; 390; 392; 395; 396; 397; 399; 402; 403; 405]
	y	[10; 20; 22; 31; 36; 39; 42; 49; 58; 69; 78; 89; 88; 93; 94; 98; 104; 112; 113; 116; 114; 117; 115; 121; 126; 127; 130; 135; 136; 151; 155; 156; 161; 162; 171; 183; 197; 212; 213; 220; 221; 227; 235; 236; 242; 248; 256; 270; 272; 273; 280; 285; 298; 299; 303; 316; 327; 330; 329; 338; 339; 341; 342; 351; 352; 354; 356; 357; 366; 367; 369; 389; 391; 401; 398; 400; 404; 407]
14	n	[6; 10; 17; 20; 22; 31; 36; 39; 42; 49; 58; 60; 61; 62; 63; 66; 69; 72; 73; 75; 78; 89; 88; 93; 94; 112; 113; 116; 114; 117; 115; 121; 123; 126; 127; 130; 135; 136; 137; 156; 161; 162; 171; 172; 173; 183; 193; 197; 203; 212; 213; 217; 220; 221; 227; 235; 236; 238; 248; 256; 264; 268; 270; 272; 273; 280; 285; 289; 298; 299; 303; 305; 310; 314; 316; 326; 327; 338; 351; 353; 354; 355; 356; 357; 360; 364; 366; 369; 383; 388; 389; 390; 391; 401; 398; 400; 402; 403; 405]
	y	[3; 24; 41; 74; 87; 90; 98; 99; 104; 105; 151; 155; 167; 184; 191; 199; 200; 202; 207; 209; 225; 226; 229; 230; 231; 232; 233; 242; 247; 258; 259; 260; 265; 279; 282; 283; 287; 292; 313; 317; 324; 327; 330; 329; 339; 341; 342; 343; 344; 346; 348; 347; 352; 365; 367; 382; 384; 385; 386; 392; 395; 396; 397; 399; 404; 407]

#	Value	Publications
15	n	[3; 6; 20; 22; 24; 31; 36; 39; 41; 49; 58; 61; 62; 66; 69; 73; 74; 78; 87; 90; 89; 88; 93; 94; 98; 99; 104; 105; 112; 113; 115; 121; 126; 127; 130; 136; 151; 156; 161; 167; 171; 183; 184; 191; 199; 200; 202; 207; 209; 212; 213; 217; 221; 225; 226; 227; 229; 230; 231; 232; 233; 235; 236; 242; 247; 248; 256; 258; 259; 260; 265; 268; 270; 272; 273; 279; 280; 282; 283; 285; 292; 298; 299; 303; 313; 316; 317; 324; 327; 330; 329; 338; 339; 341; 342; 344; 348; 347; 351; 352; 353; 354; 355; 356; 357; 364; 366; 365; 369; 382; 383; 384; 385; 386; 391; 392; 396; 397; 399; 398; 400; 402; 404; 407]
	y	[10; 17; 42; 60; 63; 72; 75; 116; 114; 117; 123; 135; 137; 155; 162; 172; 173; 193; 197; 203; 220; 238; 264; 273; 287; 289; 305; 310; 314; 326; 327; 343; 346; 360; 367; 388; 389; 390; 395; 401; 403; 405]
16	n	[3; 6; 17; 24; 41; 60; 61; 62; 63; 66; 72; 73; 74; 75; 87; 99; 105; 123; 137; 151; 162; 167; 172; 173; 184; 191; 193; 199; 200; 203; 207; 209; 217; 225; 226; 229; 230; 231; 232; 233; 238; 247; 258; 259; 260; 264; 265; 268; 273; 279; 282; 283; 287; 289; 292; 305; 310; 313; 314; 317; 324; 326; 327; 343; 344; 346; 348; 347; 353; 355; 356; 360; 364; 365; 382; 383; 384; 385; 386; 388; 390; 395; 396; 397; 399; 402; 403; 405]
	y	[10; 20; 22; 31; 36; 39; 42; 49; 58; 69; 78; 90; 89; 88; 93; 94; 98; 104; 112; 113; 116; 114; 117; 115; 121; 126; 127; 130; 135; 136; 155; 156; 161; 171; 183; 197; 202; 212; 213; 220; 221; 227; 235; 236; 242; 248; 256; 270; 272; 273; 280; 285; 298; 299; 303; 316; 327; 330; 329; 338; 339; 341; 342; 351; 352; 354; 357; 366; 367; 369; 389; 391; 392; 401; 398; 400; 404; 407]

#	Value	Publications
17	n	[6; 10; 17; 20; 22; 31; 36; 39; 42; 49; 58; 60; 61; 62; 63; 66; 69; 72; 73; 74; 75; 78; 90; 89; 88; 93; 94; 112; 113; 116; 114; 117; 115; 121; 123; 126; 127; 130; 135; 136; 137; 156; 161; 162; 171; 172; 173; 183; 193; 197; 202; 203; 212; 213; 217; 220; 221; 227; 235; 236; 238; 247; 248; 256; 264; 268; 270; 272; 273; 280; 285; 289; 298; 299; 303; 305; 310; 314; 316; 326; 327; 329; 338; 346; 351; 353; 354; 355; 356; 357; 360; 364; 366; 369; 383; 388; 389; 390; 391; 392; 401; 398; 402; 403; 405]
	y	[3; 24; 41; 87; 98; 99; 104; 105; 151; 155; 167; 184; 191; 199; 200; 207; 209; 225; 226; 229; 230; 231; 232; 233; 242; 258; 259; 260; 265; 279; 282; 283; 287; 292; 313; 317; 324; 330; 339; 341; 342; 343; 344; 348; 347; 352; 365; 367; 382; 384; 385; 386; 395; 396; 397; 399; 400; 404; 407]
18	n	[3; 6; 20; 22; 24; 31; 36; 39; 41; 49; 58; 61; 62; 66; 69; 78; 87; 90; 89; 88; 93; 94; 98; 99; 104; 105; 112; 113; 115; 121; 126; 127; 130; 136; 151; 161; 167; 171; 183; 184; 191; 199; 200; 207; 209; 212; 213; 217; 221; 225; 226; 227; 229; 230; 231; 232; 233; 235; 236; 242; 248; 256; 258; 259; 260; 265; 268; 270; 272; 273; 279; 280; 282; 283; 285; 292; 298; 303; 313; 316; 317; 324; 329; 338; 339; 341; 342; 344; 348; 347; 351; 352; 353; 354; 355; 356; 357; 364; 366; 365; 369; 382; 383; 384; 391; 392; 396; 397; 399; 398; 400; 407]
	y	[10; 17; 42; 60; 63; 72; 73; 74; 75; 116; 114; 117; 123; 135; 137; 155; 156; 162; 172; 173; 193; 197; 202; 203; 220; 238; 247; 264; 273; 287; 289; 299; 305; 310; 314; 326; 327; 330; 343; 346; 360; 367; 385; 386; 388; 389; 390; 395; 401; 402; 403; 404; 405]

#	Value	Publications
19	n	[3; 6; 20; 39; 41; 49; 58; 60; 61; 62; 63; 66; 74; 75; 87; 90; 89; 88; 94; 104; 105; 112; 113; 116; 114; 117; 121; 126; 136; 151; 155; 161; 162; 167; 171; 172; 173; 191; 193; 200; 203; 207; 209; 212; 213; 217; 220; 221; 225; 226; 231; 232; 233; 235; 236; 238; 247; 248; 259; 260; 264; 268; 270; 272; 273; 279; 280; 283; 289; 292; 298; 299; 303; 305; 310; 316; 317; 326; 330; 329; 338; 339; 341; 348; 347; 351; 353; 354; 355; 356; 357; 364; 366; 369; 382; 383; 384; 388; 390; 391; 401; 399; 398; 402; 403; 404; 407]
	y	[10; 17; 22; 24; 31; 36; 42; 69; 72; 73; 78; 93; 98; 99; 115; 123; 127; 130; 135; 137; 156; 183; 184; 197; 199; 202; 227; 229; 230; 242; 256; 258; 265; 273; 282; 285; 287; 313; 314; 324; 327; 342; 343; 344; 346; 352; 360; 365; 367; 385; 386; 389; 392; 395; 396; 397; 400; 405]
20	n	[3; 6; 10; 17; 20; 22; 24; 31; 39; 41; 49; 58; 60; 61; 62; 63; 66; 69; 72; 73; 75; 78; 87; 90; 89; 88; 93; 94; 99; 104; 105; 112; 113; 116; 114; 117; 115; 121; 123; 127; 130; 135; 136; 137; 151; 155; 161; 162; 167; 171; 172; 184; 193; 199; 200; 203; 207; 213; 217; 220; 221; 225; 226; 229; 230; 231; 232; 236; 238; 247; 248; 256; 258; 264; 268; 270; 273; 279; 280; 285; 289; 292; 298; 299; 303; 316; 317; 326; 327; 330; 341; 342; 344; 346; 348; 347; 351; 353; 354; 355; 356; 360; 364; 366; 365; 369; 382; 383; 384; 388; 390; 391; 392; 401; 398; 400; 403; 404; 405]
	y	[36; 42; 74; 98; 126; 156; 173; 183; 191; 197; 202; 209; 212; 227; 233; 235; 242; 259; 260; 265; 272; 273; 282; 283; 287; 305; 310; 313; 314; 324; 329; 338; 339; 343; 352; 357; 367; 385; 386; 389; 395; 396; 397; 399; 402; 407]
	Commerce	[3; 41; 61; 99; 112; 155; 162; 200; 209; 217; 220; 221; 233; 242; 264; 287; 339; 343; 344; 348; 347; 369; 388; 389; 392; 397; 399; 402]
	Communication	[90; 183; 193; 305]

#	Value	Publications
	Domain Independent	[42; 58; 75; 230; 248; 256; 270; 341; 385; 386; 391]
	Education	[63; 66; 74; 105; 123; 137; 151; 156; 172; 173; 191; 212; 225; 226; 227; 268; 289; 292; 316; 317; 330; 329; 366; 365; 384]
	Energy	[171; 235; 236; 272; 357; 401]
	Entertainment	[10; 22; 39; 73; 78; 116; 114; 117; 115; 130; 135; 167; 199; 202; 231; 232; 247; 273; 283; 285; 298; 313; 327; 338; 342; 367; 390; 395; 404]
	Health	[6; 17; 20; 24; 62; 87; 89; 88; 93; 94; 98; 104; 126; 127; 136; 184; 197; 207; 213; 229; 238; 258; 259; 260; 279; 280; 282; 299; 303; 310; 324; 351; 352; 353; 354; 355; 360; 383; 396; 398; 400; 403; 405; 407]
	Other	[49; 60; 314; 356; 364]
	Smart Home	[72; 113; 203; 346]
	Transport	[31; 36; 69; 121; 161; 265; 273; 326; 382]

3

Cluster-based Reinforcement Learning

Chapter 3 was published as:

Grua, E. M., & Hoogendoorn, M. (2018, November). Exploring clustering techniques for effective reinforcement learning based personalization for health and wellbeing. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 813-820). IEEE.

Abstract - In this chapter we answer ***T.RQ*1**, namely: *How can RL-based personalisation for e-Health be improved?* For the domain of health and wellbeing personalisation can contribute to better interventions and improved health states of users. In order for personalisation to be effective in this domain, it needs to be performed quickly and with minimal impact on the users. Reinforcement learning is one of the techniques that can be used to establish such personalisation, but it is not known to be very fast at learning. Cluster-based reinforcement learning has been proposed to improve the learning speed. Here, users who show similar behaviour are clustered and one policy is learned for each individual cluster. An important factor in this effort is the method used for clustering, which has the potential to influence the benefit of such an approach. In this chapter, we propose three distance metrics based on the state of the users (Euclidean distance, Dynamic Time Warping, and high-level features) and apply different clustering techniques given these distance metrics to study their impact on the overall performance. We evaluate the different methods in a simulator with users spawned from very distinct user profiles as well as overlapping user profiles. The results show that clustering configurations using high-level features significantly outperform regular reinforcement learning without clustering (which either learn one policy for all or one policy per individual).

3.1 Introduction

Personalisation is defined by [109] as "a process that changes the functionality, interface, information access and content, or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals". Personalisation has become omnipresent in our society (e.g. [361; 95; 76; 160]). While applications were historically limited to web shops and alike, a whole range of applications can nowadays be seen.

What technique is best suited to obtain personalisation depends greatly on the task at hand. Take personalisation for health and wellbeing. In such a setting one aims to perform actions to influence the behaviour and physical state of the user to improve the overall health state. The health setting is challenging: consequences and appropriateness of actions cannot be observed immediately. Some actions might have a negative impact at first, only showing benefit in the distant future. In addition, the appropriateness of actions is likely very dependent on the user context. One technique which can be used for personalisation fits this setting very well is reinforcement learning (cf. [160]). Unfortunately it does have its downsides: the learning process can be very slow (requiring a lot of experiences) and exploring undesired or ineffective parts of the action space can lead to user disengagement.

Several approaches have been proposed to overcome these problems. One set of approaches includes the usage of transfer learning, i.e. reusing previously generated policies (cf. [345]). Alternatively, [406] have proposed to cluster users to make the reinforcement learning process more effective while still enabling a level of personalisation. In the case of [108], users are assigned to a cluster after some initial period, and a policy is learned per cluster. While the initial results are promising, the results highly depend on the quality of the clustering (cf. [108]), i.e. whether the users in a cluster are sufficiently alike in terms of the policy that works best for them.

In this chapter, we explore cluster-based reinforcement learning more in depth, focusing on the approach to cluster users. We define different distance metrics based on the states of the users (based on the Euclidean distance, Dynamic Time Warping cf. [35], and by deriving high-level features), and combine them with two well-known clustering techniques (Agglomerative Clustering and K-Medoids). Next, we study the influence of the choice upon the overall performance in terms of personalisation. In addition, we investigate how the presence or absence of very distinct groups of users impacts the benefit of using cluster-based reinforcement learning. We make use of an existing simulation environment [108] which allows the simulation of users in a health context (focused on getting people to perform sufficient daily physical exercise). Using such a simulator allows us to easily manipulate users, their behaviour and the existence of distinct profiles, hence, it allows us to purely focus on the clustering techniques themselves.

This chapter is organised as follows. First, we will describe related work in Section 3.2. Section 3.3 details our proposed clustering approach, while Section 3.4 briefly describes the simulator we use for our experiments. The experimental setup is described in Section 3.5 and the results in Section 3.6.

3.2 Related Work

As discussed in the introduction we use reinforcement learning as a mean to learn when to give the intervention to the user (in our case the generated agent). Reinforcement learning has not been applied frequently in health intervention settings while it is well suited for these types of problems (see e.g. [377; 333]). There are however some papers that have already explored its suitability.

[158] proposed the use of reinforcement learning to help decide on the correct type of message needed to be sent to users of a mobile application affected with diabetes type 2 to encourage physical activity. The role of the reinforcement learner was to correctly choose the type of message that would most effectively encourage the patient to increase his/her physical activity (which is beneficial for patients with diabetes type 2). This case is an example of a one-size fits all model.

[406] addresses the problem with using either a one-size fits all policy and using individual learning. They suggest the use of clustering to achieve a balance between the amount of data available to the learner and the individual personalisation. They show that with the cluster-based reinforcement learning, they manage to achieve higher values of reward compared to both other methods, though they assumed a fixed clustering approach and the action space was limited.

Whilst the previous studies have commonalities with our work, the most similar study is [108]. Here the authors expand on the work of [406] and built a dedicated simulator to evaluate the approach for more difficult scenarios. That same simulator is used in our study. Furthermore, we wish to employ the setting used by [108] whilst expanding the clustering analysis component.

Lastly, our work also contains similarities to transfer learning [345] where a learned policy from one task can be transferred to another, which in our case could apply to the use of the learned policy from one user (or group of users) to a new user. This is not done in our particular study due to the assumption of a universal timeline for all agents generated.

3.3 Approach

As explained before, we exploit cluster-based reinforcement learning to improve the learning speed of reinforcement learning algorithms in a health and well-being context. Here, we focus on learning how to provide the most effective interventions to improve the future health state of the user. Our precise case study will be explained in the next section. In this section, we focus on the reinforcement learning component first. As a starting point, we formulate the problem. We will use a model-free reinforcement learning formulation. After we have defined this formally, we will focus on learning reinforcement learning policies for users. Then we will go to the main contribution of this chapter, namely the introduction of different clustering approaches to cluster users and learn policies over such clusters to improve the learning speed and quality.

3.3.1 Reinforcement Learning Problem Formulation

The problem we are facing is a control problem, which we model using a Markov Decision Process (MDP) [377]. This formulation follows (cf. [108]). In our formulation, we identify a user with the subscript $\boldsymbol{u}$ (with $\boldsymbol{u} \in \boldsymbol{U}$). The MDP for our problem can be specified as $\boldsymbol{M_u} = \langle \boldsymbol{S_u}, \boldsymbol{I}, \boldsymbol{T_u}, \boldsymbol{R_u} \rangle$. Here, $\boldsymbol{S_u}$ specifies the user states, and $\boldsymbol{I}$ represents the interventions that can be selected (i.e. actions in reinforcement learning terms). $\boldsymbol{T_u}$ specifies the probabilistic transition function of a user $\boldsymbol{u}$ and is defined as follows $\boldsymbol{T_u} :: \boldsymbol{S_u} \times \boldsymbol{I} \times \boldsymbol{S_u} \rightarrow [\mathbf{0}, \mathbf{1}]$. This function expresses the probability of moving from one user state to another, provided that we have selected an intervention from $\boldsymbol{I}$. $\boldsymbol{R_u}$ is the reward function, which assigns a reward based on the observed state $\boldsymbol{s_u}$ and the intervention $\boldsymbol{i} \in \boldsymbol{I}$ provided to user $\boldsymbol{u}$. Since we are dealing with human subjects in our setting, we cannot assume complete knowledge. $\boldsymbol{T_u}$ cannot be directly accessed (i.e. we assume it to be unknown). Furthermore, we cannot observe the full state, but only a vector of features $\boldsymbol{\phi}$ derived from the state $\boldsymbol{s_u} \in \boldsymbol{S_u}$. Considering $\boldsymbol{p}$ features we specify this vector as follows: $\boldsymbol{\phi(s_u) = \langle \phi_1(s_u), \ldots, \phi_p(s_u) \rangle}$. While we cannot know up front whether the process in fact satisfies the Markov property, we assume the process to be sufficiently close such that we can employ standard reinforcement learning algorithms.

Given this problem formulation, we want to learn a policy $\boldsymbol{\pi_u}$ per user, that expresses what intervention should be selected in which state $\boldsymbol{\pi :: S_u \rightarrow I}$. Applying such a policy results in experiences for each time point $\boldsymbol{t}$: $\boldsymbol{\langle \phi(s_u^t), r_u^t, i^t \rangle}$. Here, we use $\boldsymbol{t}$ to identify the specific time point. These experiences together accumulate in traces (referred to as $\boldsymbol{\Sigma}$) for each user $\boldsymbol{u}$: $\boldsymbol{\Sigma_u}$ (with $\boldsymbol{T}$ being the last time point):

$$\boldsymbol{\langle \phi(s_u^t), r_u^t, i^t, \phi(s_u^{t+1}), r_u^{t+1}, i^{t+1}, \ldots, \phi(s_u^T), r_u^T, i^T \rangle} \tag{3.1}$$

We define the value of doing intervention $\boldsymbol{i}$ in state $\boldsymbol{s}$ as:

$$\boldsymbol{Q^{\pi}(s,i) = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r^{t+k+1} | s^t = s, i^t = i\}} \tag{3.2}$$

$\boldsymbol{\gamma}$ is a discount factor for future rewards. Then, the policy we strive to find maximizes this value (i.e. selects the best interventions in each state):

$$\boldsymbol{\pi'(s) = \arg\max_i Q^{\pi}(s,i), \ \forall s \in S} \tag{3.3}$$

To find such a policy, we deploy an off-policy reinforcement learning algorithm, namely Least Square Policy Iteration (LSPI) [189]. This uses the feature vector of the state ($\boldsymbol{\phi(s)}$) and finds a linear approximation of the Q function by means of a weight vector $\boldsymbol{\langle w_1, \ldots, w_p \rangle}$ containing a weight for each of our $\boldsymbol{p}$ features from a batch of experiences. Different alternatives are possible, but this is outside the scope of this chapter. The techniques explained below are however independent of the specific reinforcement learning algorithm that is selected.

3.3.2 Learning Policies

One of the problems when dealing with human users is that there is hardly room for an exploratory phase in which a lot of different actions can be tried. Furthermore, the state space (even when using our feature vector $\boldsymbol{\phi}$) is potentially very large. When we learn our policy, we can make a choice how user specific we want to learning such a policy. We can:

- learn one policy over all users (Pooled approach)
- learn one policy per user (Separate approach)
- learn one policy per group of similar users (Clustering approach)

The first two options are simple. For learning, we can simply vary what experiences we feed to our reinforcement learning algorithm. For learning one policy over all users, we provide $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma_u} | \boldsymbol{u} \in \boldsymbol{U}\}$ and generate a single policy across all users. For learning a policy for a single user, we only provide the experience for that user: $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma_u}\}$. Both options come with downsides. Learning one policy across all users will highly likely result in insufficiently tailored interventions, while learning per individual will suffer from a lack of experiences to learn a reasonable policy in a short time frame. We therefore study learning across groups of users that seem to be relatively alike (following [108]). We define these groups using clustering techniques, and want to learn policies per cluster. We provide the learning algorithm with the following experiences: $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma_u} | \boldsymbol{u} \in \boldsymbol{C}\}$. While learning across such clusters has already shown to be beneficial (cf. [108]), the impact of the clustering approach itself has not been studied in depth.

3.3.3 Clustering

In order to define clusters, we need to have (1) a clustering technique, and (2) a distance metric. Let us consider the distance metric first. We will refer to the distance between a user $\boldsymbol{u_1}$ and user $\boldsymbol{u_2}$ as $\boldsymbol{d(u_1, u_2)}$. What can we base this distance metric on? Initially, we assume to have no knowledge about the specific users (and hence, we cannot determine a distance between users). We therefore start with a so-called *warm-up phase* where we gather experiences of users with a random policy. Once collected, we can define a distance between the experiences we have gathered for the users. These experiences are in fact temporal sequences of the information we have available about the user at each time point (the features describing the state of the user, the intervention, and reward information). We define three distance metrics between experiences of users: (1) using the Euclidean distance, (2) using Dynamic Time Warping (cf. [108]), and (3) using derived features.

For the *Euclidean distance*, we measure the distance between the states of the user, and do not consider the actions and rewards. The rational behind our decision on only including the states is because the states are a closer representation of the behaviour of the agent as defined by the profile settings. We did not want to include information that is more dependant on the setup of the learner in the clustering of the agents. We assume that the feature vector $\boldsymbol{\phi}$ (representing what we observe about the state f the user) only contains numerical features. If there are categorical features, we can encode categorical features using one hot encoding. To calculate the distance we simply compare the difference between the state features as follows:

$$\boldsymbol{d_{ED}(u_1, u_2) = \sum_{t=0}^{T} \sqrt{\sum_{i=1}^{p} (\phi_i(s^t_{u_1}) - \phi_i(s^t_{u_1}))^2}} \tag{3.4}$$

In this calculation, we assume that the sequences of both users are of equal length and their start times have been synchronized. The second approach considers *Dynamic Time Warping (DTW)* [35]. This allows for a more flexible matching between the experiences of users, where the speed of the sequences might be different. As a basic building block, a distance function between two experiences of users is defined:

$$\boldsymbol{d_{ED}(u_1^t, u_2^{t'}) = \sqrt{\sum_{i=1}^{p}(\phi_i(s_{u_1}^t) - \phi_i(s_{u_1}^{t'}))^2}} \tag{3.5}$$

Again, we only consider distances between the features of the states. DTW tries to match time points in order to minimize the sum of the distances over time points provided that: (1) the first and last time points of both sequences are matched, and (2) a monotonicity condition is satisfied. See [35] for more details. For the DTW, we split the sets of experiences into a number of intervals of $\boldsymbol{k}$ discrete time points within which we perform the DTW (i.e. $\boldsymbol{[t, \ldots, (t+k)), \ldots [(T-k), T)}$). For example, think of splitting the sequences of experiences into days, and comparing how equal the states within a day are. This is done for computational reasons, but also because we do not want to match outside of these boundaries to avoid overly optimistic matches over days. The overall user distance is defined as:

$$\begin{aligned}\boldsymbol{d_{DTW}(u_1, u_2) = \sum_{i=0}^{T/k} dtw(\langle \phi(s_{u_1}^{i \cdot k}), \ldots, \phi(s_{u_1}^{i \cdot k+(k-1)})\rangle,} \\ \boldsymbol{\langle \phi(s_{u_2}^{i \cdot k}), \ldots, \phi(s_{u_2}^{i \cdot k+(k-1)})\rangle)}\end{aligned} \tag{3.6}$$

The final distance metric we consider is *derived features* from the sequences of experiences and comparing on that higher level. An example is to derive the average values per feature over the entire series of experiences and compute the distance between those averages:

$$\boldsymbol{d_{DF}(u_1, u_2) = \sqrt{\sum_{i=1}^{p}\left(\frac{\sum_{t=0}^{T}\phi_i(s_{u_1}^t)}{|\{0, \ldots, T\}|} - \frac{\sum_{t=0}^{T}\phi_i(s_{u_2}^t)}{|\{0, \ldots, T\}|}\right)^2}} \tag{3.7}$$

Given these distance metrics, we can apply standard clustering techniques (which we deliberately leave open in this approach). These are commonly parameterized algorithms, which require a selection of the number of cluster (e.g. $\boldsymbol{k}$ in K-Medoids clustering) or a threshold to be set which in fact determines the number of clusters (e.g. in Hierarchical Clustering). To select the best value for a parameter, we use an evaluation metric commonly used in clustering to evaluate the quality of the clusters: the silhouette (cf. [302]). We run the clustering algorithms for various parameter settings and select the setting which results in the highest quality clustering with this metric.

3.4 Simulator

To test our approach, we utilize a simulation environment[1] which is able to generate realistic behaviour of human-like agents for a health and wellbeing setting. The simulator we use is described more extensively in [108]. It focuses on trying to coach people towards a healthier lifestyle by engaging them more in sports, a common goal among health apps available in the iTunes or Google Play store [321]. The simulator emulates the behavior of human beings by generating their activities throughout the day (e.g. working, eating, working out) as well as their responses to interventions they receive in the form of messages that encourage them to work out. How schedules and responses are generated is based on certain generic profiles (e.g. think of an average working person). States are observed once per hour. The features of the state ($\boldsymbol{\phi}$) are the current day of the week, the current hour of the day, if the agent has worked out within the current day, the fatigue level and which of the possible activities he performed in the captured hour.

As said, the acceptance of the intervention depends on the schedule of the agent, their fatigue level, as well as their profile. The reward is given based on a few conditions. If the agent accepts the intervention given, a reward of $+1$ is recorded, whilst if the intervention is rejected then a negative reward of -0.5 is returned. If the intervention was accepted an extra reward of $+10$ is given when the workout is completed. The duration of the workout can also be considered but for our setup we have decided not to do so. The final condition that can score reward is the level of fatigue of the agent. The amount of negative reward

[1] 'https://github.com/EMGrua/MultiAgentSimulation-MultiClusterVariation'

recorded increases with the amount of fatigue. For our case fatigue is defined as an incremental integer that starts from 0 and increases for every consecutive workout. The moment the agent skips, rejects or is not told to workout the fatigue level is reset to 0. As an example, if an agent works out three days in a row (each day working out once) its current fatigue level is equal to three. When the fourth day the agent does not workout the fatigue level gets reset.

For our investigation, we use sets of three profiles from which agents are spawned. The technicalities of each profile used are explained in subsection 3.5.3. The simulator has been implemented in Python3.

3.5 Experimental Setup

In order to evaluate our approach, we perform a number of experiments. In this section, we explain the different experimental conditions, the performance evaluation, and the parameters and simulator settings.

3.5.1 Experimental Conditions

We are interested in studying the performance of our cluster based learning approach compared to the two alternative variations we mentioned in Section 3.3.2 (pooled and separate). In addition, we want to understand how the distance metric and the selected clustering algorithm impacts performance. We use our three distance functions and combine these with two commonly known clustering algorithms, namely K-Medoids clustering [170] and Hierarchical Clustering (Agglomerative Clustering, using the complete linkage criterion) [393]. While more advanced clustering algorithms are available, we want to start with relatively simple approaches which can also easily be combined with the various distance functions chosen. Overall, this results in $\mathbf{2 \times 3 = 6}$ variations for the clustering. Thus we have 8 variations of the reinforcement learning algorithm in total.

How easily groups of users can be distinguished (and whether they are present or not) is likely to have a severe impact on the advantage of using a cluster-based approach. To study this influence, we try two different setups of our simulation environment. One setup features three highly distinctive profiles (both in terms of their daily schedules and responses to the received interventions) while the second setup will again be three profiles but with two being very difficult to distinguish. Subsection 3.5.3 shows the specification of the profiles used in both settings.

3.5.2 Performance Evaluation

To evaluate the performance of the algorithms, we focus on two aspects.

To study the performance of the clustering itself, we apply clustering to the traces of experiences we collect during the *warm-up phase* in which we apply a random policy. We study the users residing in the resulting clusters and consider the original profiles they were spawned from. A desirable outcome would be to see low diversity of profiles within a single cluster. We perform five runs per clustering algorithm as the results are highly dependent on the random initialization of the centres (certainly for K-Medoids).

The second evaluation is the performance of the reinforcement learning algorithm and the resulting reward. Hereto, we consider the average reward we obtain. Next to the aforementioned *warm-up* period, we apply a *learning period* during which we measure the reward. For all variants, after the *warm-up* days we create a policy using LSPI and train each LSPI instance over the traces of the associated agents. Each policy is then updated on a daily basis over the remaining *learning period* and used to select the interventions. We compute the average daily rewards over all runs, agents and time points per day (called the average daily reward).

The best performing clustering configurations will be selected and compared to both the *separate* and the *pooled* cases. To determine whether the difference between trends is statistically significant we used the Wilcoxon signed-rank test. We define various levels of significance: one star ($\star$: P ≤ 0.05); 2 stars ($\star\star$: P ≤ 0.01), and three stars ($\star\star\star$: P ≤ 0.001).

3.5.3 Parameter and Simulator Settings

For each experiment the simulation was ran with a constant set of parameters. These parameters were chosen based on preliminary experiments and feasibility of the run times. The parameters chosen were:

– *Number of agents*: the number of agents for all runs was set to 100, with the agent profiles being equally distributed among them, so we always expect a profile distribution of 33-33-34.

– *Warm-up phase*: the 'warm-up phase' was set for all runs to 7 days.

– *Learning phase*: the 'learning phase' was set to 60 days (which is enough to obtain a stable policy).

The simulation parameters that were changed according to the executed experiment were the profile types. Below we list the two sets used (distinct and overlapping) as well as the key differences between each type of profile. The *distinct profiles* are:

– *Worker*: works 5 days a week plus he has a 80% of working on the sixth day (Saturday). The Worker starts anywhere from 8 a.m. to 9 a.m. and works for 10 to 11 hours. Gets fatigued after 2 consecutive workouts and has a 10% chance of accepting a second workout in the same day.

– *Athlete*: works 3 days a week (Monday, Tuesday and Thursday) starting from around 9 a.m. for 8 hours. The athlete gets fatigued after 4 consecutive workouts and has a 50% chance of accepting a second workout in the same day.

– *Retired*: never works. The retiree gets fatigued after one workout and will never accept a second workout on the same day.

The *overlapping profiles* are:

– *newWorker*: identical to Worker but does not have a chance of working a sixth day. The newWorker is also identical in the way it behaves with working out and fatigue pattern.

– *newAthlete*: identical to the athlete but has a 60% chance of working on Wednesday and a equal chance of working on Friday. NewAthlete is also identical to Athlete in the fatigue and workout settings.

– *Athlete*: identical to the previously described athlete.

It is important to remember that apart from these differences all of the profiles include routine actions, such as eating (breakfast, lunch, dinner) and sleeping.

3.6 Results

In this section, we present the results we obtained using the experimental setup we have just described. We start with the analysis of the clusters, followed by the performance of the reinforcement learning techniques[2].

3.6.1 Clustering Analysis

Let us analyse the clusters found for the two different profile setups.

3.6.1.1 Distinct Profiles

Let us first consider the distinct profile case. Table 3.1 provides an overview of the results we obtained. Each row represents one of these variations whilst the first 5 columns show the number of clusters found per run. The following columns show the mode value/s of the number of clusters for the set of runs and the median. In order to keep the following tables and graphs clear and compact we have abbreviated the various experimental cases as follows:

k : is used when the clustering technique used was K-medoids.

h : is used for when Hierarchical Clustering was utilised

eu : stands for the Euclidean distance metric

[2] The data can be found here:'http://doi.org/10.5281/zenodo.1215905'

Table 3.1: *Number of clusters returned by each experimental case (for the Distinct Profile case)*

	run1	run2	run3	run4	run5	Mode	Median
keu	2	2	3	3	4	2,3	3
heu	6	4	6	2	3	6	4
kdtw	3	3	3	4	3	3	3
hdtw	2	4	4	3	5	4	4
kdf	3	3	4	3	3	3	3
hdf	3	3	3	3	3	3	3

dtw : signifies the use of Dynamic Time Warping

df : indicates the use of the derived features

An example abbreviation is 'keu'. This abbreviation stands for the experimental setup that used K-Medoids as clustering algorithm with the use of the Euclidean distance metric on features directly related to the state. In contrast 'kdf' is the same but with the use of the derived features.

Important to note that in the 'kdf' and the 'hdf' median cases the resulting clustering of the agents corresponded to the perfect distribution of the profiles to the agents. Furthermore the 'kdtw' median case resulted in near perfect clustering; cluster A contained 31 out of 33 athlete agents and one retiree agent, cluster B contained the remaining 2 athlete agents and the rest of the retiree agents with the last cluster only containing all of the Worker type agents. In the case of 'keu' even though three clusters are found, one of the clusters contains most of the agents, with the second cluster containing 3 athlete agents and 7 retiree agents and the last cluster containing only one athelete agent. Similar happened with the 'heu' case, where one cluster contains most agents and the others only have a few.

3.6.1.2 Overlapping Profiles

For the overlapping profiles the results are shown in Table 3.2. The major outcomes that can be taken away from the table are that in hardly any run three clusters were found. Furthermore, it is interesting to note that in the

Table 3.2: *Number of clusters returned by each experimental case (for the Overlapping Profile case)*

	run1	**run2**	**run3**	**run4**	**run5**	**Mode**	**Median**
keu	3	5	5	2	2	2,5	3
heu	2	2	2	2	2	2	2
kdtw	2	6	6	2	3	2,6	3
hdtw	2	2	2	2	2	2	2
kdf	2	2	2	2	2	2	2
hdf	2	2	2	6	2	2	2

methods using the derived features, one of the clusters contained most, if not all, of the 'newWorker' type agent. Finally, a similar behaviour can be seen in the case of the Hierarchical Clustering using DTW, but instead of dividing the 'newWorker' agents from the rest, it divided the 'Athlete' type agents from the 'newWorker' and 'newAthlete' agents.

3.6.2 Reinforcement Learning Results

Given the clusters that have been found, we will study the impact on the RL performances now.

3.6.2.1 Distinct Profiles

Within this subsection we will be describing the results found by the reinforcement learning analysis in terms of reward over time for the case of the distinct profiles. The first table presented, Table 3.3, shows a comprehensive overview of all of the experiments run. This overview clearly shows that several of the cluster-based approaches obtain higher cumulative rewards compared to the pooled and separate cases. It seems that the derived features perform best, the Euclidean distance approaches perform worst, and the DTW approaches reside in the middle (while still performing better than the separate and pooled cases). The clustering technique does not seem to have a severe impact on the overall rewards that are obtained.

Let us look into the rewards collected over time. To ease comparison we have

Table 3.3: *Cumulative Average Daily Reward for all experimental cases (Distinct)*

KEU	**HEU**	**KDTW**	**HDTW**
340.1	419.4	740.2	778.7
KDF	**HDF**	**POOLED**	**SEPARATE**
878.3	889.5	310.0	681.9

selected only four of the six clustering methods used. To make the selection we have excluded the two worst performing methods in terms of cumulative average daily reward (both of the Euclidean distance cases). Note that the performances during the warm up period are identical as a random policy is followed. Analysing Fig. 3.1 we can notice a recurring pattern that holds for all plots that will be presented, all of the reward trends have the same kind of 'rhythm' to them. This is causes by the fatigue concept. What is important to note in this particular figure is how K-Medoids with DTW has consistently the lowest average reward. This suggests that this method (in this particular case) was the least effective (out of the four selected ones) in aiding the reinforcement learners in producing effective policies.

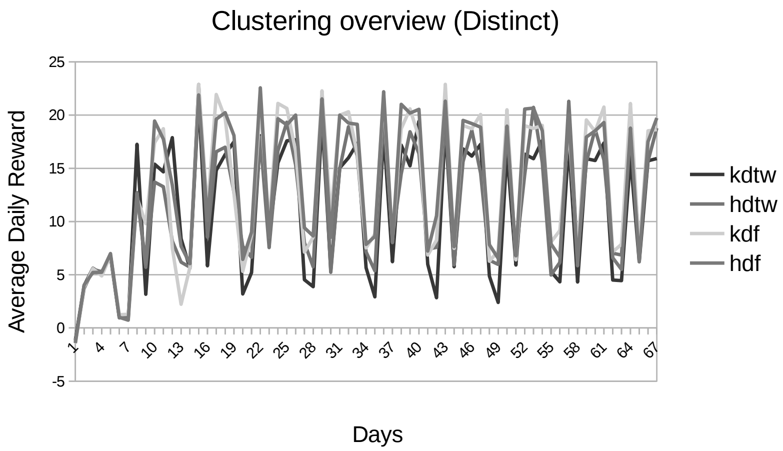


Figure 3.1: *Plot of the Average Daily Reward over time for the four better performing clustering methods*

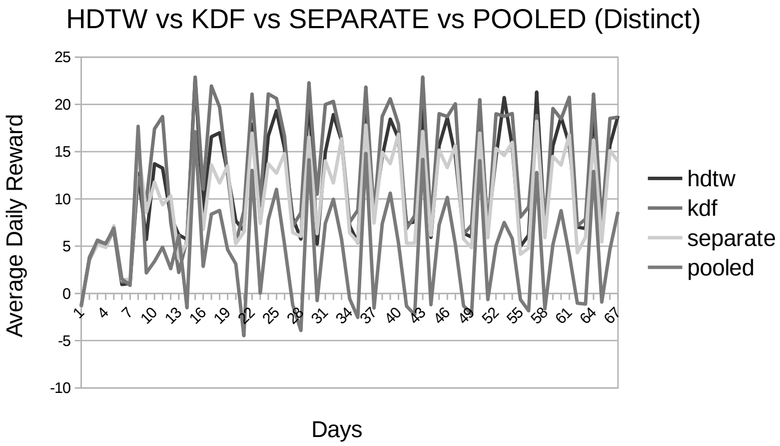


Figure 3.2: *Plot of the Average Daily Reward over time comparing the two selected clustering methods and the two non-clustering methods (Separate and Pooled)*

Figure 3.2 illustrates the final two selected clustering methods and compares it to the *pooled* and *separate* approaches. To have a comprehensive comparison, we chose the best clustering technique for the non-derived features and the best one for the derived features. In this figure we can easily notice how poorly the *pooled* aided at the creation of a good policy. Furthermore, the daily average reward resulting from *separate* appears to always be below our selected methods.

In order to draw critical conclusions from the comparison we used the Wilcoxon signed-rank test on all possible combinations of the selected methods to find potential statistical differences (as reported in Table 3.4).

Table 3.4: *Table of returned Wilcoxon p-values for all of the selected experimental methods (Distinct)*

hdtw vs kdf	pvalue=3.0675e-06⋆⋆⋆
hdtw vs separate	pvalue=8.2819e-07⋆⋆⋆
hdf vs separate	pvalue=3.0421e-10⋆⋆⋆
separate vs pooled	pvalue=5.1079e-12⋆⋆⋆
hdtw vs pooled	pvalue=4.8880e-12⋆⋆⋆
kdf vs pooled	pvalue=5.8275e-12⋆⋆⋆

The table shows that all of the Fig. 3.2 plotted lines are indeed statistically different from each-other (with the highest rating).

3.6.2.2 Overlapping Profiles

In this subsection we repeat the analysis in the same mode as previously described, but on the results obtained by the overlapping profiles.

Table 3.5: *Cumulative Average Daily Reward for all experimental cases (Overlapping)*

KEU	**HEU**	**KDTW**	**HDTW**
1327.7	1281.7	1380.5	1248.9
KDF	**HDF**	**POOLED**	**SEPARATE**
1586.9	1662.1	1312.4	1322.5

Similarly to Table 3.3, Table 3.5 shows a comprehensive overview of all of the experiments done within the overlapping profiles case by illustrating the cumulative average daily rewards. We see that again the derived features perform best, also better than the pooled and separate approaches. This is positive, since the clustering is less obvious for this case.

By selecting the best four clustering methods, with the same criteria as before, we have therefore discarded the two Hierarchical Clustering cases not utilising the derived features. Fig. 3.3 shows the results. Here we observe that the K-Medoids Euclidean method is consistently scoring the lowest average reward, and close to it is the K-Medoids DTW. This once again illustrates the enhanced difficulty in clustering that the 'overlapping profiles' have, compared to the 'distinct' case.

As of last, Fig. 3.4 shows the two chosen clustering methods compared to the case of *pooled* and *separate*. For clarity, the selection criterion of the final two clustering methods is the same as the one used in the choice of the final two clustering methods in the 'distinct profile' case. Furthermore, the *pooled* case is once again the lowest of all cases, but is not reaching negative values as it was happening in the other profile case. This, plus the overall rise in average

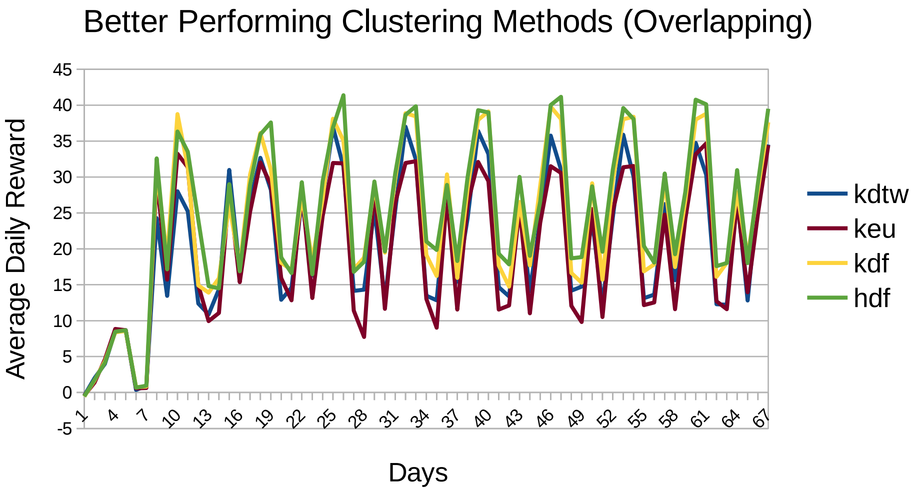


Figure 3.3: *Plot of the Average Daily Reward over time for the four better performing clustering methods*

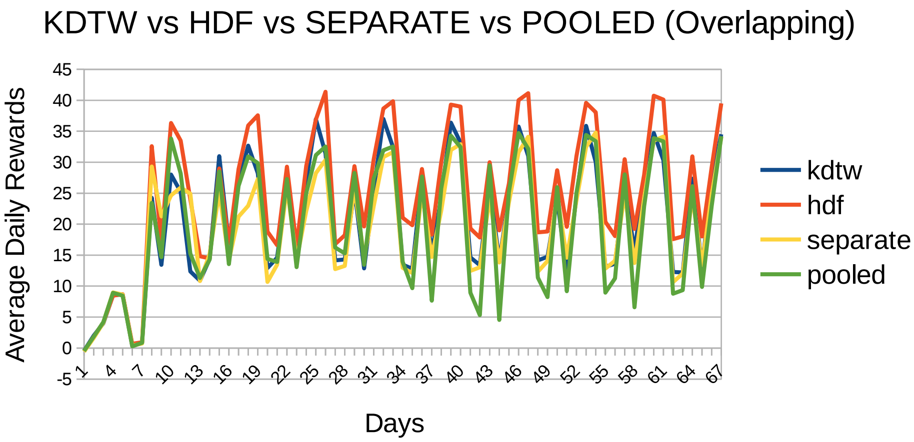


Figure 3.4: *Plot of the Average Daily Reward over time comparing the two selected clustering methods and the two non-clustering methods (Separate and Pooled)*

reward across all methods can be attributed by the lack of the 'Retired' agent profile combined with the profile's low maximum fatigue threshold.

Table 3.6: *Table of returned Wilcoxon p-values for all of the selected experimental methods (Overlapping)*

kdtw vs hdf	pvalue=8.6357e-12⋆⋆⋆
kdtw vs separate	pvalue=0.0005⋆⋆⋆
hdf vs separate	pvalue=1.0275e-11⋆⋆⋆
separate vs pooled	pvalue=0.7477
kdtw vs pooled	pvalue=0.0456⋆
hdf vs pooled	pvalue=2.1030e-12⋆⋆⋆

Table 3.6 presents the significance results. We want to bring to the attention the now non-statistically significant difference between the *separate* and the *pooled* methods and how our selected DTW method, whilst remaining statistically significant, has now a one-star p-value when compared to the *pooled* method in contrast to the three-star significance when the same comparison was made in the 'distinct profiles' scenario. Nonetheless, even though the 'overlapping profiles' case caused the clustering methods to produce what seemed like worst clusters, we still outperformed both the *separate* and *pooled* case in a statistically significant manner. Therefore showing the benefit of using cluster based reinforcement learning.

3.7 Discussion and Future Work

With this chapter we answer ***T.RQ*1**, namely: *How can RL-based personalisation for e-Health be improved?*. We explored in depth the benefits that cluster-based reinforcement learning can have on personalisation in e-Health. We set up our study in-line with the related work we have found in this field, and expanded the analysis on the different cluster methodologies that can be used in this setting.

Our results show that with distinct profiles the clustering methods utilising DTW and the derived features produced good clusters that were either perfectly matching the profile assignments or extremely close to it. For overlapping profiles, we see that a logical division of the agents was made but it still remains one that does not match the original assignment of the profiles to the agents. For both cases we outperform the *separate* and the *pooled* reinforcement learning approaches. Here, the derived features approach performs best, but the dynamic time warping also performs reasonably well. This seems even more remarkable given the somewhat poor clustering that resulted in the case of the overlapping agent profiles. This finding supports our initial intuition and the findings brought forth by [406; 108].

As future work it would be good to expand on our study and test other types of clustering techniques to see how the reinforcement learner reacts to potentially different patterns in the clusters. Another interesting variation to study would be to dynamically change the clusters over the course of the simulation, similarly as the reinforcement learner is continuously updating its policy over time. To do this we would need to use an online-clustering algorithm. To the best of our knowledge, there is currently a lack of online clustering algorithms tailored for e-Health. In the next chapter we will be tackling this limitation.

4

Clustering Growing Timeseries

Chapter 4 was published as:

Grua, E. M., Hoogendoorn, M., Malavolta, I., Lago, P., & Eiben, A. E. (2019, October). Clustream-GT: online clustering for personalization in the health domain. In IEEE/WIC/ACM International Conference on Web Intelligence (pp. 270-275).

Abstract - The goal of this chapter is to answer our ***T.RQ*2**, namely: *How can online-clustering be used to efficiently and effectively cluster e-Health data?* Clustering of users underlies many of the personalisation algorithms that are in use nowadays. Such clustering is mostly performed in an offline fashion. For a health and wellbeing setting, offline clustering might however not be suitable, as limited data is often available and patient states can also quickly evolve over time. Existing online clustering algorithms are not suitable for the health domain due to the type of data that involves multiple time series evolving over time. In this chapter we propose a new online clustering algorithm called CluStream-GT that is suitable for health applications. By using both artificial and real datasets, we show that the approach is far more efficient compared to regular clustering, with an average speedup of 93%, while only losing 12% in the accuracy of the clustering with artificial data and 3% with real data.

4.1 Introduction

Personalisation in the health domain can contribute greatly to an improved wellbeing among patients [371; 294; 157]. For example, personalisation entails selecting a dedicated intervention for a patient that is most likely to improve the patients health state. Applications vary from more medical cases in hospitals [7] to mobile health apps such as sport trackers [239] or apps to battle depression [281] . Performing personalisation in the health domain is challenging: bad suggestions are highly undesirable, data can be limited for specific types of health-related domains, and doing real life experiments to collect more data is not trivial at all.

Clustering is often considered a valuable step in providing personalised support or recommendations to users (see e.g. [177; 81]) in order to have enough data to base recommendations on. Using clustering, like-minded users are grouped, and recommendations (or the aforementioned interventions) are found that are relevant for the users in the group. Mostly, this clustering is done in an offline fashion, i.e., once the clusters are determined they do not change in real-time and are only updated in a batch mode over possibly long intervals.

While this is fine for companies such as Netflix and Amazon with a user base of millions and without rapid changes of user preferences, for the health domain this might come with severe disadvantages: (1) the number of users could be very limited and one would want to exploit the most recent data of all patients, and (2) the health state of users can vary greatly and change rapidly over time. Hence, clustering in a real-time fashion is much more desirable.

In the literature, there are various algorithms that allow for the online updating of clusters. Well-known examples include CluStream [5], ODAC [301], and others [57; 70]. They do however not fit the health domain well.

In the health domain, we mostly consider measurements over time per patient, and intend to cluster on a patient level. This means that online clustering approaches need to cope with: (1) new patients arriving, and (2) new data of known patients coming in. Both situations potentially require an update of clusters, while in existing approaches only the second case is tackled, assuming the number of data points (in our case patients) do not change as time progresses. In this chapter, we present an online clustering approach for the health care domain that is called CluStream-GT (standing for: CluStream for Growing Timeseries). It is able to online cluster patients with evolving time series (i.e. an increasing amount of data per patient over time). This approach is an extension of the popular CluStream approach. CluStream-GT online clusters inputted time-series by first checking if the data is meant as an update of an already clustered patient or is categorising a new one. In the former case it then decides if the newly updated time-series has to be re-clustered or can be kept in the already assigned cluster. Whilst in the latter case it will always decide which cluster is better suited to include the new patient data. In order to evaluate the CluStream-GT, we use both real medical EEG data and artificial data. We compare the quality of the clusters found by CluStream-GT to k-means (cf. [216]) and ODAC (as it is the closest existing method to CluStream-GT) by having both algorithms re-cluster at each timepoint and storing the average silhouette score [302]. We also compare the total execution time of the approaches. We therefore analysed the two metrics and found a beneficial trade-off in the use of CluStream-GT.

4.2 Approach

In this section, we formally define the problem we are addressing, followed by an explanation of the proposed algorithm.

4.2.1 Problem Description

We assume that we have a set of users $\boldsymbol{U}$: $\boldsymbol{u_1, \ldots, u_k}$ (patients in our case) which can generate health related data. The health data contains a number of features that are measured around the users at each time point: $\boldsymbol{\{f_1, \ldots, f_n\}}$. The domain of each feature $\boldsymbol{f_i}$ is denoted by $\boldsymbol{F_i}$. The values of these features are measured over time, we assume that at each time point when a measurement is performed, all feature values are measured. For a measurement at time point $\boldsymbol{t}$ for user $\boldsymbol{u_i}$, the value of the feature $\boldsymbol{f_j}$ is noted as $\boldsymbol{v(u_i, f_j, t)}$. We use $\boldsymbol{S_{t_0}^{t_1}(u_i)}$ to denote the time series of a user $\boldsymbol{u_i}$, containing vectors of values of all features over all time point between $\boldsymbol{t_0}$ and $\boldsymbol{t_1}$. Furthermore, $\boldsymbol{t_{start}(u_i)}$ is used to specify the time when the time series started for user $\boldsymbol{u_i}$ and $\boldsymbol{t_{end}(u_i)}$ when it ended.

Our task is to cluster the series of values over all features for the users. Here, the start, end, and length of the series of the users can vary freely across the users. In order to specify our algorithm, we assume that some aggregation function $\boldsymbol{a}$ is available that summarises the the entire time series into a single number that can be compared to other time series (i.e. this is our distance metric). We require the following property of the function (to simplify, we assume one feature $\boldsymbol{f_j}$ here):

$$\boldsymbol{a(S_{t_0}^{t_k}(u_i)) = a(a(S_{t_0}^{t_k-1}(u_i)), S_{t_k}^{t_k}(u_i)))} \tag{4.1}$$

This property allows updating the aggregate of the time series without having to maintain a history of all values.

4.2.2 CluStream-GT algorithm

To solve the clustering problem, we deploy an approach similar to CluStream. CluStream works based on microclusters. These are used as intermediate step before the clustering of the entire dataset is used. They are initialised offline with a small sub-sample of the dataset. Microclusters are specified by means of five components that summarise the data within the microcluster:

- the sum of all values of the datapoints
- the sum of all squared values of the datapoints
- the sum of all time points associated with the datapoints
- the sum of the squared values of all time points associated with the datapoints
- the number of datapoints contained in the microcluster

Given our formal notation before, we slightly adjust the microcluster definition to make it suitable for our setting:

- the sum of all values of the aggregation function of the users in the microcluster:
  $\sum_{\forall u \in U_i} a(S_{t_{start}(u)}^{t_{end}(u)}(u))$
- the sum of all squared values of the aggregation function:
  $\sum_{\forall u \in U_i} \left(a(S_{t_{start}(u)}^{t_{end}(u)}(u))\right)^2$
- the sum of the last time points for all users:
  $\sum_{\forall u \in U_i} t_{end}(u)$
- the sum of the squared values of the last time points for all users:
  $\sum_{\forall u \in U_i} (t_{end}(u))^2$
- the set of users contained in the microcluster ($U_i \subset U$)

As a second step, these microclusters are used as datapoints in a standard clustering approach resulting in macroclusters. Having the microclusters as an intermediate step saves valuable storage space, but also computational effort in the clustering. Of course, it is essential to have appropriate microclusters that group users in a suitable way. CluStream therefore, when new data arrives, assigns the new data point to an existing microcluster if it is sufficiently alike, or its own microcluster in case it is too different. In the latter case, two existing microclusters are merged. Microclusters containing too many old datapoints can also be removed. In our setting, life becomes slightly more complicated as data points are now time series. Hence, we can have two cases: (1) a new datapoint arrives in a time series of an existing user/patient, or (2) a datapoint arrives of a new patient. We need to accommodate for both cases. Algorithm 4 shows our adjusted version of CluStream to accommodate for this setting.

In our extension, we consider the update when a new patient arrives the same as with CluStream (of course, using our aggregation function again). In case a new datapoint for an existing patient arrives we only update the properties of the microcluster that patient had already been assigned to in case it results in a minor adjustment of the aggregate value. Otherwise, we go to a full blown re-clustering. This process can be seen in line 7 of Algorithm 4 where the adjustment is judged against a threshold value $\boldsymbol{\delta}$.

In the pseudocode, the additions that were made to CluStream are seen from line 1 to 9 in Algorithm 4. We also assume that within the health domain we want to retain the information of all the clustered time-series, no matter how old they are. If they would ever have to be deleted, the user of the algorithm can do so manually, but we do not want to give permission to the algorithm to automatically remove information. Therefore we have removed the possibility of the algorithm deleting micro-clusters.

For the offline cluster creation (or macro-clustering process) CluStream-GT works similarly to CluStream. It uses k-means but uses the micro-clusters as the input data. Furthermore, the centroids are chosen as the k most populated micro-clusters. This can be easily achieved by analysing the last element of each micro-cluster tuple and choosing the k highest ones. This macro-clustering

Algorithm 4 CluStream-GT pseudocode

```
Require: new_data, id
  t ← current_time
  // We have seen the patient already
  if id is known then
    m_c ← get_micro_cluster(id)
    prev_t ← get_last_t(id)
    prev_aggr ← get_prev_aggregate_value(id)
    new_aggr ← calculate_aggregate(get_prev_data(id),
                   new_data)
    // We see only a small change, just update the microcluster properties
    if |new_aggr − pre_aggr| < δ then
      m_c ← update_microcluster(m_c, id, t, prev_t,
                   new_aggr, prev_aggr)
      // Major change, remove from microcluster, find the best fitting
      cluster and update
    else
      m_c ← remove_from_microcluster(m_c, id, prev_t,
                   prev_aggr)
      goto find
    end if
    // New patient
  else
    new_aggr ← calculate_aggregate(null, new_data)
  find:
    m_c ← find_best_microcluster(id, t, new_aggr)
    if distance(new_aggr, t, m_c) < max_boundary then
      m_c ← update_microcluster(m_c, id, t, null, new_aggr,
                   null)
      // Microclusters need to merge, patient in a new microcluster
    else
      [m_c_1, m_c_2] ← min_{m_c_1, m_c_2 ∈ micro_clusters}
                   distance(m_c_1, m_c_2)
      m_c_1 ← merge_microclusters(m_c_1, m_c_2)
      m_c_2 ← null
      m_c_2 ← update_microcluster(m_c_2, id, t, null,
                   new_aggr, null)
    end if
  end if
```

process has the clear advantage of not requiring storage of the whole time-series dataset as we use only use the micro-clusters, making our approach better suited for cases with limited resources.

4.3 Experimental Setup

In this section we explain the experimental conditions and evaluations we have used to test CluStream-GT's performance. We compare CluStream-GT against two alternatives: (1) k-means clustering in each iteration, and (2) ODAC (cf. [301]).

4.3.0.0.1 Scenarios To assess the difference in performance we performed tests for three scenarios: two were done with generated synthetic data and one with the use of a real world dataset. For the generated data conditions we utilised sine functions to generate our time-series. In the first test (henceforth referred to as: the base case) we had each time-series associated to a specific set of parameters inputted in the sine function. This includes the function itself and noise surrounding the curve. In the second test (henceforth referred to as: the advanced case) each time-series started with an associated set of parameters, analogous to the base case. However, at each timestep the considered time-series had a 10% chance of changing its parameters and therefore have its data generated by a different sine function. This extra factor was introduced as a method of representing the potential change in behaviour that can be observed in time series associated with human behaviour, especially within the health domain (e.g. vital signs getting better, mood improving, more frequent physical activity, etc.). To add a more practical test scenario we have also used of a real dataset. This dataset is a collection of EEG recordings that were published by Andrzejak et al. in 2001 [13].

4.3.0.0.2 Performance metrics The two aspects investigated across all of our tests were accuracy of the clustering and speed of execution. To asses clustering accuracy, we decided to utilise the silhouette score [302]. For the execution time, we kept track of the total length of the execution of the algorithms thereby performing the experiments on the same machine with no other processes open in order to minimise potential variance.

4.3.0.0.3 Algorithm setup All of the tests are set up to represent a realistic scenario for all techniques (CluStream-GT, k-means and ODAC). We therefore update at every single timestep as in the health care domain data can be scarce and therefore all available data should be exploited as much as possible to create the most up-to-date clusters. This means that we re-cluster every time any form of new data is given to the algorithm. That includes both a new time-series and any amount of new data related to an already clustered one.

The k-means used as benchmark clustered using the means of each time-series present in the dataset. The mean was used since it was also utilised as our selected aggregate function (described in Section 4.2) for distance computation during the online phase of CluStream-GT. Both ClusStream-GT and k-means clustering require the number of clusters to be set. To make the results comparable we fixed this value to $\boldsymbol{k} = \mathbf{3}$ for both cases as initial experiments have shown this to be the best value for all scenarios. Our replication package contains the full experimental setup implemented in Python as used to perform these experiments[1].

4.3.0.0.4 Experimental Conditions For both generated cases, we had a starting population of 120 time-series uniformly distributed over three clusters. The experiments were performed over the course of 30, 60 and 90 simulated days, with each case repeated 30 times. At each day there was a 50% chance of adding new time-series to the dataset (simulating the addition of a new patient). The amount added ranged from one to five chosen with the use of a uniform distribution. It is important to note that this feature of our experiment was not used whilst testing ODAC. This is because ODAC cannot work on datasets with changing numbers of timeseries. Per day, each generated time-series consisted

[1] https://github.com/EMGrua/CluStream-GT

of 24 points. This was selected to simulate pooling done once at each hour of the day.

For the real dataset, we ran tests on three cases: 100, 200 and 300 total patients. Each patient had a time-series containing 4097 individual datapoints. Each one of these scenarios was repeated five times. ODAC was run only a single time as it lacks stochasticity and therefore would always return the same results.

4.4 Results

Section 4.4.1 illustrates the results we gathered from the two generated data test cases, whilst Section 4.4.2 does so with the EEG dataset results. As we could never compute the silhouette score for ODAC we illustrate those results separately (see Section 4.4.3).

4.4.1 Results from the generated data

We will first discuss the results for the base case. Figure 4.1 illustrates the distribution of the average silhouette scores obtained for each of the test scenarios. The CluStream-GT mean averages at 0.86 and the median to 0.9, whilst the k-means mean and median average at 0.93. Secondly, it is interesting to note the skewed distribution occurring for all cases of CluStream-GT. Furthermore, we can observe some runs that result in differences deviant enough from the mean to be classified as outliers. The potential cause of this behaviour could be attributed to poor initialisation of the micro-clusters within those runs, which then followed with worse overall clustering and therefore a lower silhouette score. Nevertheless, the distributions favour a smaller difference with the IQR ranging from the smallest Q1 equal to 0.84 to the highest Q3 equal to 0.92. Differently is the execution time trends of the two techniques (shown in Figure 4.2). We clearly observe that with the growing amount of data the saved execution time also grows. This can be certainly attributed to the use of the micro-cluster tuples for the generation of the macro-clusters, which caps the amount of data used to cluster. Therefore, the time increase is only due to the higher number of runs of the online component needed by CluStream-GT to update the micro-clusters. No matter the amount of data, CluStream-GT provides at least a 90% speedup.

Examining now the silhouette score for the advanced case, we observe a bigger difference between CluStream-GT and k-means, and a less skewed distribution for all the scenarios of CluStream-GT (as shown in Figure 4.3) as compared to the base case. In this case we observe a number of outliers, although smaller than with the base case. The overall higher difference between the two approaches is to be expected as CluStream-GT is trying to cluster a now far more complex time-series with only the use of the meta-data contained in the micro-cluster tuples. This provides somewhat of an advantage to our benchmark k-means which has access to the mean of each time-series present in the generated dataset.

Finally, we examine the execution times recorded for each scenario of the advanced case (shown in Figure 4.4). Similar to the execution times of the base case, CluStream-GT minimally grows as the data does, whilst the execution time of k-means continues to grow. This indicates that the more data is clustered and the higher is the speed gain achieved by using CluStream-GT. In fact in the 90 days test case we achieve an average 94.7% speed-up.

4.4.2 Results from the real dataset

We start by analysing the results collected from the silhouette scores (shown in Figure 4.5). The mean values recorded from both algorithms are extremely similar, with only a small loss in the silhouette score by CluStream-GT compared to k-means. Furthermore, the standard deviation for each case was minimal. This suggests reliable clustering over repetitions and therefore reliable clustering overall. This is somewhat in contrast with the generated data, where both CluStream-GT and k-means showed wider standard deviations, reinforcing our assumption that the deviation in the generated data is due to the noisier nature of said data. Moving to execution times we observe the huge advantage that using CluStream-GT gives over k-means. In Figure 4.6 we see that whilst

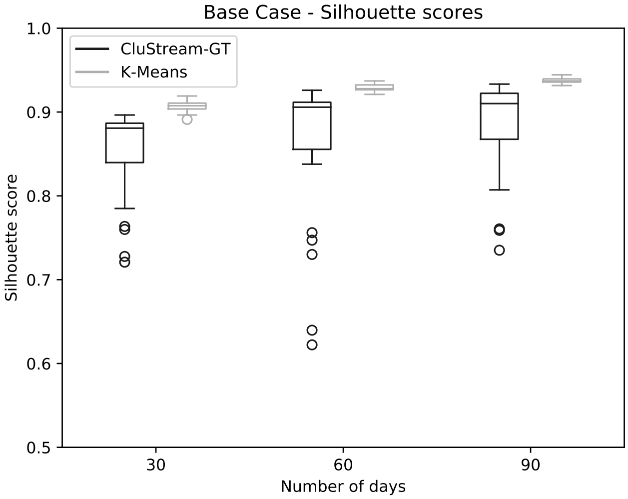


Figure 4.1: *Decrease of the the average silhouette score using CluStream-GT compared to k-means (Base Case)*

k-means drastically increases its execution time with the increase of data, CluStream-GT barely increases. This leads to a difference in execution time that becomes more substantial the bigger the dataset is. Taking the case of 300 patients the average execution time for k-means is of 20000 seconds (5 hours and 33 minutes) whilst CluStream-GT's average execution time is only 1036 seconds (17 minutes and 20 seconds). This effectively is a 95% improvement.

4.4.3 Results obtained by the use of ODAC

Over all tests, ODAC consistently maintained only one node of its tree structure, hence clustering all data under one cluster. As a result, it was impossible for us to measure the silhouette score. In order to make such measure, ODAC would have had to result in at least two separate clusters. The reason for this behaviour seems to stem from the algorithm stalling on the first node, splitting and aggregating consecutively. This type of behaviour has also been reported by the authors of ODAC as well [300]. A cause could be that updates are

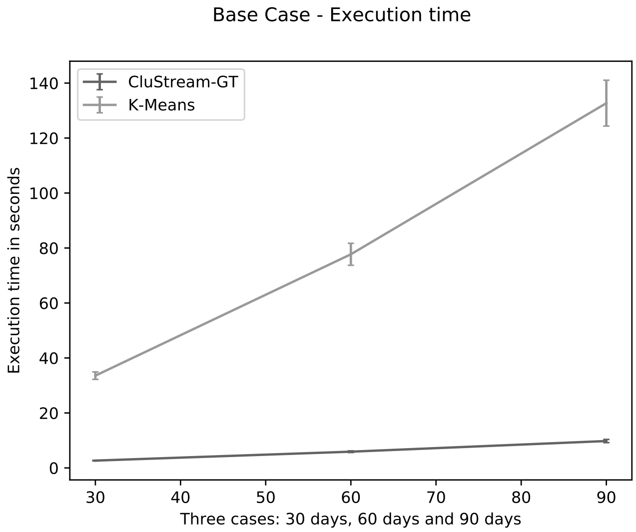


Figure 4.2: *Decrease of the the average execution time using CluStream-GT compared to k-means (Base Case)*

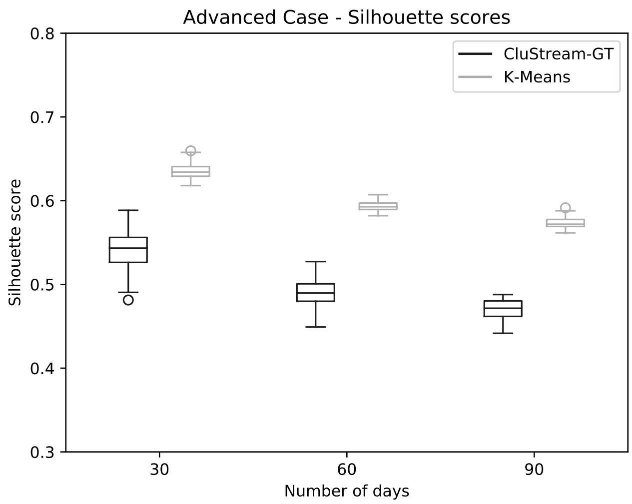


Figure 4.3: *Decrease of the the average silhouette score using CluStream-GT compared to k-means (Advanced Case)*

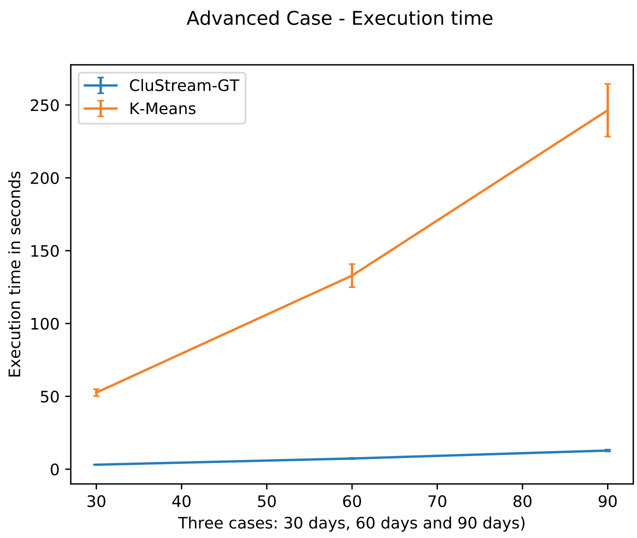


Figure 4.4: *Decrease of the average execution time using CluStream-GT compared to k-means (Advanced Case)*

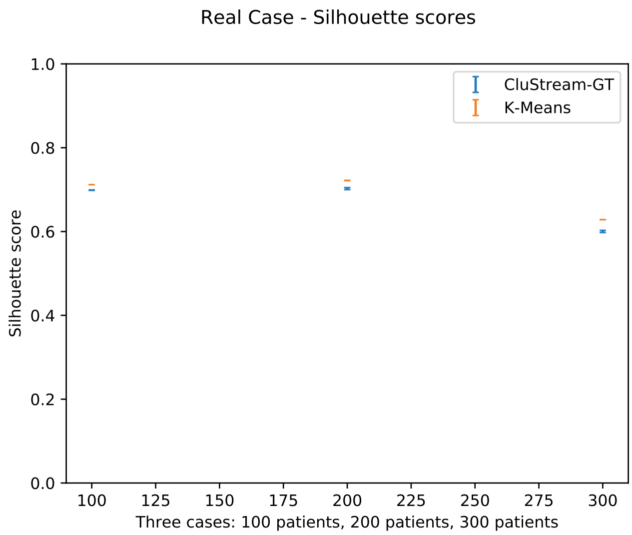


Figure 4.5: *Decrease of the average silhouette score using CluStream-GT compared to k-means (Real Case)*

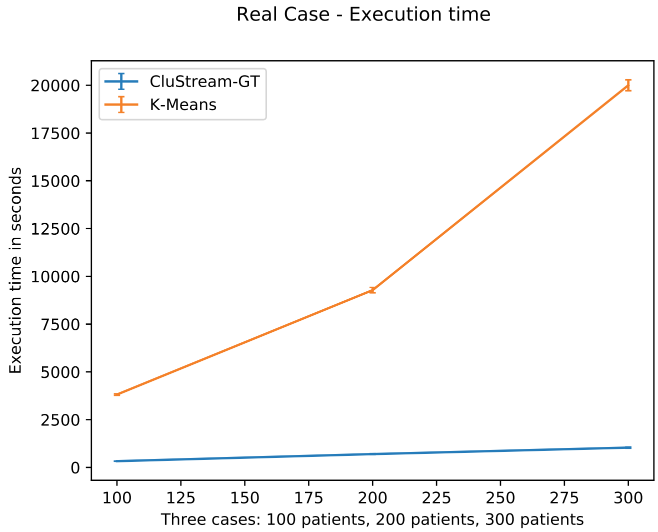


Figure 4.6: *Decrease of the average execution time using CluStream-GT compared to k-means (Real Case)*

performed at each time step, while in experiments using ODAC often batches of time points are used. ODAC was also consistent in the registered execution times. For all cases tested, CluStream-GT was, on average, 98% faster than ODAC (as reported in Table 4.1). This is an expected consequence, given that ODAC increases in speed with an increasing number of leaves, otherwise needing to recompute all dissimilarities each time new data is clustered (a calculation that has a quadratic complexity on the number of data streams) [301].

4.5 Discussion and Future Work

In this chapter we address ***T.RQ2***, namely: *How can online-clustering be used to efficiently and effectively cluster e-Health data?*

We have answered this research question by developing an online clustering algorithm, tailored for e-Health, that can cluster growing timeseries.

We have developed this algorithm by modifying the already existing data

	Clustream-GT	ODAC
Base Case 30 Days	2.4	197.8
Base Case 60 Days	4.6	398.2
Base Case 90 Days	6.9	605.8
Advanced Case 30 Days	2.7	197.8
Advanced Case 60 Days	5.7	402.7
Advanced Case 90 Days	8.8	610.1
Real Dataset 100 patients	327.4	19368
Real Dataset 200 patients	697.9	79965.7

Table 4.1: *Execution times (in seconds) for Clustream-GT and ODAC on all executed tests*

stream clustering algorithm CluStream and so named ours CluStream-GT. We formalised CluStream-GT's function in Section 4.2 where we present pseudocode and explain the input and global variables used by the algorithm to perform the micro-cluster updates. We then evaluated our approach by the use of three test scenarios: two of them were executed using generated data, whilst the third one was performed using a real EEG dataset [13].

As described in Section 4.3, for all test cases we recorded the total execution time and the average silhouette score obtained by re-clustering at each timestep. We compared Clustream-GT against k-means and ODAC for three scenarios. ODAC clustered all data under one cluster for all experimental conditions, it was impossible to compute, and therefore compare, the silhouette score with that of CluStream-GT. CluStream-GT was 98% faster than ODAC on all cases. We explain this as ODAC, remaining on a structure of one node, had to execute under its worst case scenario. Therefore, having to recompute all dissimilarities at each new time step (an operation with quadratic complexity).

When comparing k-means with CluSteam-GT for the base case, CluStream-GT provides a good trade-off between accuracy and execution time by speeding up the performance by 92% whilst only loosing an average of 0.06 on the silhouette score. For the advanced case, the trade-off is similar as we lose an average of 0.1 on the silhouette score but still achieve significant speedup with CluStream-GT performing 94.5% faster. The bigger divide in silhouette score, as compared to the base case, can be explained by the increase in noise that the advanced case brings to the data due to the chance of timeseries suddenly switching behaviour and therefore making the clustering a more challenging task. This is especially apparent for CluStream-GT since it only uses the descriptive data contained in the micro-clusters for the formation of the final macro-clusters.

Lastly, in the test case performed with the EEG dataset CluStream-GT performed excellently. The speed-up was of at least 91% with it improving to 95% with the 300 patients run. This meant that on the machine used for testing k-means it took a total of 5 and a half hours whilst CluStream-GT only took a little more than 17 minutes. This was achieved with an extremely small trade-off on the silhouette score, with the worst case being the 300 patients run in which CluStream-GT had on average 0.028 less on the silhouette score.

For future work we would like to augment CluStream-GT with a mechanism to detect poor micro-cluster initialisation at an early stage. Whilst it was not a problem for the less noisy EEG data, we did record a few outlier cases in some of the runs in the generated data. We therefore aim to create such a mechanism in order to reduce or remove the possibility of such outliers appearing and therefore increasing the average silhouette score obtained.

Furthermore, as mentioned in Section 4.3 we measure execution time by measuring the time difference with the python module Time. Whilst we minimised the risk of variance with repeated runs and assuring that the machine had no other processes open apart from our experiment, it would be desirable to repeat the experiments on other machines in order to further validate our findings.

Lastly, we have mentioned throughout our work that the execution time gap increases with the size of the data and have explained this phenomenon by CluStream-GT's use of the micro-clusters and lack of needing to store the entire dataset. However, we have not investigated how much more efficient CluStream-GT can be on storage space. This would be an interesting fact to investigate especially for CluStream-GT's therefore potential use on lower spec hardware, such as mobile devices.

5
Predicting User Engagement

Chapter 5 was published as:

Barbaro, E., Grua, E. M., Malavolta, I., Stercevic, M., Weusthof, E., & van den Hoven, J. (2020). Modelling and predicting User Engagement in mobile applications. Data Science, (Preprint), 1-17.

Abstract - The mobile ecosystem is dramatically growing towards an unprecedented scale, with an extremely crowded market and fierce competition among app developers. Today, keeping users engaged with a mobile app is key for its success since users can remain active consumers of services and/or producers of new contents. However, users may abandon a mobile app at any time due to various reasons, *e.g.,* the success of competing apps, decrease of interest in the provided services, etc. In this context, predicting when a user may get disengaged from an app is an invaluable resource for developers, creating the opportunity to apply intervention strategies aiming at recovering from disengagement (*e.g.,* sending push notifications with new contents).

The goal of this chapter is to answer ***T.RQ3***, namely: *How can we predict user engagement in apps?* To achieve our answer we propose, apply, and evaluate a framework to model and predict User Engagement (UE) in mobile applications via different numerical models. The proposed framework is composed of an optimised agglomerative hierarchical clustering model coupled to (i) a Cox proportional hazards, (ii) a negative binomial, (iii) a random forest, and (iv) a boosted-tree model.

The proposed framework is empirically validated by means of a year-long observational dataset collected from a real deployment of a waste recycling app. Our results show that *in this context* the optimised clustering model classifies users adequately and improves UE predictability for all numerical models. Also, the highest levels of prediction accuracy and robustness are obtained by applying either the random forest classifier or the boosted-tree algorithm.

5.1 Introduction

Mobile applications (hereinafter "apps") dominate the digital world today, reaching incredible numbers and showing no signs of slowing down its market growth anytime soon [195]. For example, as of March 2018, there are more than 3.3 million Android applications available [328], with more than one thousand apps being published *everyday* [195]. Mobile apps are not only being published in large numbers, but are also being consumed by users in large numbers, with more than 1.5 billion downloads from Google Play Store every month [12]. A medium of such a large scale leads to a crowded market with strong competition. Under this perspective, *mobile app developers must keep their users active over a sufficiently long period of time to be considered successful.* Recognising and understanding user motivations are key to leading to a greater app usage [178]. To date, despite significant efforts, over 95% of smartphone owners stop using an app by the end of the third month of download [288]. In other words, the majority of mobile solutions fail to achieve long-term usage. This can be explained by a variety of reasons, such as lack of personalisation, user context, and finally failure to seamlessly integrate with other apps or technologies [363; 340; 323].

A high disengagement rate is obviously non desirable to app developers, whose success depends on the usage of their app. Furthermore, it is also a problem for researchers and other professionals who use apps to provide services aimed at improving the user's quality of life. For example, waste recycling has been shown to be a positive practice for improving sustainability and diminishing carbon emissions [237; 186]. Waste recycling apps can be used as an effective tool to help users engage in recycling [40]. They can achieve this with game-like features that remind and reward the user for consistently recycling. However, for the app to succeed, it must be regularly utilised by the user. Hence, as a crucial quality, it must be engaging. Crafting personal "smart interactions" is an effective way to ensure that users remain active, on-line, and motivated [53]. Furthermore, tailored interactions aim to maintain, encourage and ultimately increase app usage over time. Take people tracking as an example: mobile location tracking has to be used on a opt-in basis, due to privacy issues [322]. However, once a device is being tracked, apps may send out alerts when the tracking is turned off aiming to prevent the user to go off-line. The nature of these interactions may vary wildly, since it is likely that users react very differently to such interventions [18].

In the context of this study, UE can be intuitively defined as the assessment of the response of the user to some type of activity or service provided by the mobile app. For example, in social networking apps (*e.g.,* Facebook or Twitter) UE is about user's posts, comments, and interaction with other users;

differently, in shopping apps (*e.g.,* Amazon or Wish) UE is about the products being purchased, being listed, saved for later purchases, and so on.

Despite there being a good understanding of what is UE in different domains and which factors contribute to it, there seems to be a lack of literature on whether it is possible to predict UE in mobile apps and how different methods perform.

In this study, we provide evidence that **it is possible to predict the engagement of mobile app users with good levels of accuracy**. We achieve this result by characterising and evaluating a framework for predicting user engagement of mobile apps. The framework is based on the application of different types of numerical models, *i.e.,* survival, counts, and classification. The numerical models take as input a minimal set of information about the user, which are relatively straightforward to collect at run-time, *e.g.,* the current point balance of the user (assuming the app is employing a potentially implicit gamification mechanism), the time of the last interaction with the app, geographic position, etc. In this study, we explore four different types of numerical models, namely: (i) survival analysis, (ii) negative binomial regression, (iii) random forest, and (iv) gradient-boosted trees. In order to complete our approach, one of the most important steps to achieve better predictions is to group users based on their past behaviour [208]. In that way, it is possible to separate - or "cluster" - users based on how (often) they interact with the mobile app. Therefore, we also incorporate a clustering algorithm to our proposed framework, aiming at targeting user interactions more accurately by means of drawing similarities between users [208].

We empirically evaluate the performance of our proposed numerical framework in predicting UE on an industrial dataset, which has been built in the context of a *real mobile app* in the area of waste recycling. The dataset is composed of approximately 27,000 entries distributed over 1,500 unique users.

Summarising, the main contributions of this study are:

- a reusable framework for modelling and predicting UE in mobile apps;
- a characterisation of UE by means of 4 different types of numerical models;

– the empirical evaluation of the prediction accuracy of the 4 different types of numerical models in the context of a waste recycling mobile app.

The contributions above benefit both mobile apps developers and researchers. Developers can re-use the proposed framework for accurately predicting the engagement of their users at run-time and counteract it in a timely fashion (*e.g.,* by sending a push notification for triggering new conversions) - see [319], and (ii) learn from the evaluated numerical models which one is better suited for their own mobile app. We support researchers since we (i) provide evidence about how various numerical models can accurately estimate UE in mobile apps and (ii) provide a framework for modelling and predicting UE, which can be further extended or used in other scientific studies.

It is important to note that the the aim of this study is not to provide a general solution for predicting UE for all mobile apps, instead we aim at providing (i) evidence that it is possible to predict UE with good levels of accuracy and (ii) a flexible framework for modelling and predicting UE in mobile apps which can be re-used by both researchers and practitioners in other projects, provided that it will be customised according to the app under consideration, its usage scenarios, and the available data.

The remainder of this chapter is organised as follows. Section 5.2 presents the fundamental background needed throughout this research. Section 5.3 presents the modelling framework, whereas the results of the evaluation of the prediction accuracy of the modelling framework are reported in Section 5.4. Finally, Section 5.5 discusses and puts into context the obtained results and Section 5.6 closes the chapter.

5.2 Background

In this section we provide background information about the definition of user engagement in the context of mobile apps (Section 5.2.1) and present the waste recycling app dataset (Section 5.2.2).

5.2.1 Defining User Engagement

User engagement is not a trivial concept to define, especially in the mobile segment. As a first attempt, UE can be described as a proxy for quantifying an outcome or, more generically, interpreting an action. In [263] the authors summarised and combined several prior definitions of engagement. They argue that UE consists of users' activities and mental models, manifested as attention, curiosity and motivation. As shown in Figure 5.1, UE can be seen as a process composed of four main steps, namely: users (i) start engaging with a mobile application, (ii) remain engaged, (iii) disengage, and finally (iv) potentially re-engage. Building on that argument, in a later study, the same authors argued that engagement is not only a product of experience, but also a cycle-process that depends on the interaction with technology [262]. Closely related to [262] and [257], [194] defined UE as the quality of the experiences that emphasise the positive aspects of the user interactions.

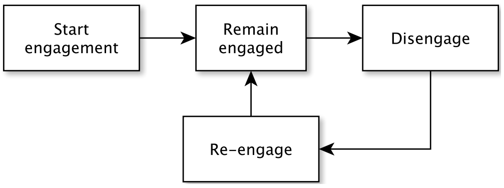


Figure 5.1: *Overview of UE life cycle. The arrows indicate the possible places of interaction with technology. Figure inspired in the four-step engagement process proposed by [263] and [262]*

More recently, on-line behaviour was analysed to better understand the temporal evolution of UE in massive open on-line courses [291] . Their findings suggest the use of diverse features - such as last lecture watched, last quiz taken, and current/total number of posts - as good quantitative indicators for modelling UE at different points in time. In their study, they used these parameters to accurately predict student survival rates already at the beginning of the course [291].

In the remaining of this section we introduce the fundamental concepts associated with the numerical tools used to model UE in mobile apps. Naturally, the first point to address here is to properly *classify* if a customer is engaged or not at the present time. Different definitions can be used - or combined - to address that. Here, we discuss:

- The application is still installed on their phone after a certain number of days;
- The number of user activities is bigger than a given threshold;
- The frequency of user activities is higher than a given threshold.

One of the simplest definitions available is called **User Engagement Index** ($UE_{\boldsymbol{I}}$). The $UE_{\boldsymbol{I}}$ compares the time of inactivity with the time the customer has been engaged. Mathematically it reads:

$$\boldsymbol{UE_I = \frac{LastEvent - FirstEvent}{Today - FirstEvent},} \tag{5.1}$$

where all the terms on the right-hand side are dates. We see in Equation 5.1 the ratio of the time difference between both last event and present time to the time of the first interaction. If $UE_{\boldsymbol{I}} > \mathbf{0.5}$ (where **0.5** is a threshold defined a priori) the user is considered engaged *today*.

Another possible way to determine UE is by defining a threshold on the **recency** (R). This threshold has to be calculated to determine if the time between actions is (long)short enough for the user to be considered (dis)engaged. Recency is trivially defined in Equation 5.2:

$$\boldsymbol{R = \Delta t,} \tag{5.2}$$

where $\boldsymbol{\Delta} t$ is the time past between one action and its *subsequent* action. In doing so, user engagement based on recency ($UE_{\boldsymbol{R}}$) can be calculated for every interaction, and not only for the last one as in $UE_{\boldsymbol{I}}$.

We base the choice of threshold to determine $UE_{\boldsymbol{R}}$ on the statistical distribution of R. The threshold is set as being at the edge of one standard deviation from the average recency. By doing so, we ensure that to be considered disengaged the user's recency has to be less than around 32% of our entire sample recency. That is a compromise between allowing for later re-engagement (by not tackling only users at the very end of the distribution, i.e. almost totally disengaged) and not sending too re-engagement messages to still engaged users (users close to the centre of the distribution). Similarly to $\mathrm{UE}_{\boldsymbol{R}}$, we explore the fact that user engagement can also be defined by setting a threshold on the **total number of actions** ($\mathrm{A}_{\boldsymbol{T}}$) a user performed within a given time frame. Mathematically, it reads:

$$\boldsymbol{A_T} = \sum_{\boldsymbol{t_0}}^{\boldsymbol{t_N}} \boldsymbol{A(t)}, \tag{5.3}$$

where $\boldsymbol{t_0}$ and $\boldsymbol{t_N}$ are respectively the initial and final times of the counting. Every user surpassing a given threshold can be considered engaged.

5.2.2 The Waste Recycling App Dataset

In this study, we use a dataset from a mobile app that promotes waste recycling. The app grants points every time an event is performed by the user, *e.g.,* disposing trash in their selected bins, reading educational material, or inviting friends to join the app. These points can then be redeemed for rewards at selected partners, such as savings on local shops or discounts on sustainable goods. Extending the framework described in [252] for tablets, we argue that the app needs to be designed and optimised having in mind that the user is most likely on their mobile phone either redeeming points at a shop or collecting points at the recycle bin. That is fundamental to create an intuitive interface that facilitates these activities and promotes engagement.

The dataset contains approximately 27,000 entries distributed over 1500 unique users and 122 variables. The data was collected between April 2015 and January 2016. Each entry of the dataset contains the following 6 features:

1. the current point balance of the user,

2. the time of the user's last event within the app,

3. the number of days since the last event,

4. the current weekday,
5. the current ZIP code,
6. the current geographical position of the user in terms of latitude and longitude.

We expand each of the 27,000 entries of the dataset to contain 122 unique variables in total. We achieve that by first generating combinations of these variables, e.g. *number of days since the first event during weekdays* or *time of the user's last event within the app during a weekday/weekend*. We then proceed to calculate the following statistics (*max/min/mean/med/sum/sd*) for all of the variables. That allows for more feature creation, e.g. *standard deviation of the number of days since the first event during weekdays*. We calculate the most simple statistics such as *mean of the current point balance* or *minimum number of days since last event*, but also combinations of variables with statistics - such as *median of the minutes since last event per user in a certain zip code*, or the *standard deviation of the number of days since the first event during weekdays*. Note that geographical position provides more detailed information than just zip-code, given that there may be more than one recycle bin in a given area.

Figure 5.2 shows the strategy that we follow for splitting the dataset into four main subsets, namely: training, test, cross-validation, and validation sets [102].

Specifically, a fraction of the dataset (60%) is used to train our models and the remaining data to test (20%) and cross-validate (20%) their performance. The last three parts (observation 1,2,3) are the validation sets. They also start at the beginning of the dataset (April, 24) and continue after the end of the training period - as shown in Figure 5.2[1]. It is important to mention that the validation sets only contain users that remained active, or started new interactions after the training period. Those are depicted in red in Figure 5.2. We highlight that this setup is general/flexible enough to be used by all our numerical models.

[1] For simplicity, we extrapolate the use of the term training period to indicate the period between April 24 and Dec 1, 2015.

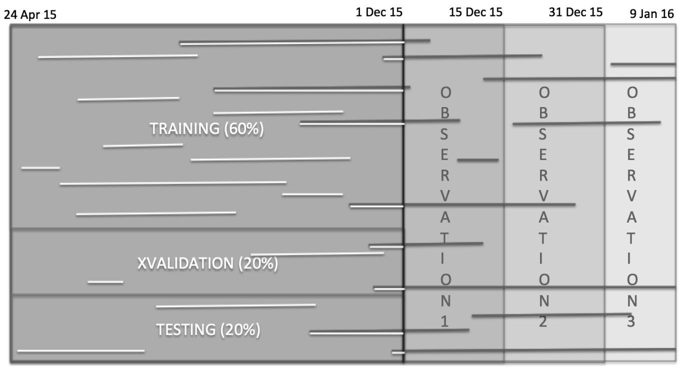


Figure 5.2: *Sketch of users lifespan over time. The red lines indicate customers engaged after the end of the training period.*

Concerning the definitions of UE, in this study, we rely on the definitions based on *recency* (see Equation 5.2) and *total actions* (see Equation 5.3). The user engagement index (see Equation 5.1) does not fit the purpose of this study since it is a too coarse-grained definition and it does not provide any information concerning the daily evolution of UE. In our case, the threshold for recency is set constant and equal to 9 days. For the counting model, we choose a threshold of 5 interactions per 2 weeks. These thresholds have been defined based on (i) a number of informal interviews we had with professionals working in the company developing the waste recycling app and (ii) the need to simulate the quick reaction of the app as soon as the users start to be disengaged. We *extensively experimented with a series of other levels of the recency and interaction thresholds around the ones used in this study, and the results of the re-applied models did not significantly vary in all the cases* ($<$ **5%**). For the sake of brevity, we do not report the whole set of the performed replications in this study. Finally, it is important to note that the values of the thresholds used in this study strongly depend on the application domain (*i.e.,* waste recycling, in our case); we suggest researchers and developers willing to re-use our framework in other domains/organisations to fine tune the selected thresholds according to the specific characteristics of the app under consideration and its typical usage scenarios (*e.g.,* social media users may be considered disengaged much earlier than after 9 days of total inactivity). In addition to that, note the modelling results - especially the quantitative component - discussed here remain specific for this dataset. Hence, it should not be directly transferred to other application domains. Instead, the main contribution of this chapter lies on the fact that we show, by means of different types of algorithms, that it is possible to accurately predict user engagement as well as a reusable framework that can be used to better understand UE in mobile apps.

5.3 Modelling User Engagement of Mobile Apps

In this Section, we detail our modelling strategy and explain the multiple steps and assumptions we make to predict UE or counts (actions) until disengagement. Here, despite the numerical model we choose, the first step is to describe the process of assigning our users to different groups, the so-called *clustering process.* In doing so, we are firstly grouping similar users together in order to reduce uncertainties and improve the predictability of our numerical models [208].

5.3.1 The clustering model

In this study, we use a modified Agglomerative Hierarchical Clustering (AHC) model [91]. That means, we assign each data point to one exclusive cluster, and then combine the two clusters that are closest to each other. This process is repeated until there is only one cluster left - containing all the observations. We utilise average linkage to perform the clustering, *i.e.,* the average distance between each point in one cluster to every point in the other cluster. We use the so-called Pearson-γ correlation as our criterion to select an appropriate number of clusters [8; 152]. This metric looks at the correlation of all the distances between data points and a binary matrix, that is equal to zero for every pair of observations in the same cluster and equal to 1 in case points are in different clusters.

Hierarchical clustering methods require a distance metric to define similarity between two observations. Here, we implement the so-called Gower's metric [139] with optimal weights, as proposed in [358]. This metric allows for the calculation of the dissimilarity between rows of our dataset for nominal, binary, and ordinal variables. The optimization is done with the intent to maximise the cophenetic correlation coefficient (CPCC), see [308]. The CPCC is the correlation between the distance matrix used for the clustering and the cophenetic distance matrix of the resulting hierarchical clustering. This cophenetic distance matrix is calculated as the distance at which two observations are combined into one cluster.

The optimisation of the CPCC is done through the use of the L-BFGS-B method (Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with Bounds), a Quasi-Newton algorithm which uses the first order derivative of a given function and an approximation of its second-order derivative to obtain the extrema of the given function (non-linear optimisation) - see [261]. We apply this method to iteratively search for an optimal set of weights to Gower's metric to optimise the CPCC of the resulting agglomerative hierarchical clustering. The bounds of the L-BGFS-B method are set to [0,1] to ensure no weight is negative. Next to that, we also use an approximation of the analytical derivative of the CPCC with respect to the weights to ensure we do not have to use finite differences for the L-BFGS-B method, hence significantly reducing computation time [358].

As the last part for the configuration of our clustering, we choose which variables we consider to be used for clustering. The variables we pick determine what our clusters represent. As an initial set of variables for our clustering algorithm, we choose all 122 variables mentioned above. In this context, our clusters represent different characteristics of the users and their behaviour, ranging from regional data to frequency of use and point collection. Users in the same cluster are thus expected to be more similar when it comes to app behaviour and geographical location compared to those in other clusters. Hence, these clusters capture useful information for our different user engagement models to use in their predictions.

With our users set to a particular cluster, we use these results as a predictor of UE improving modelling results [358]. In the next subsections, we explain the numerical models we use to predict UE for every user. We detail the three different model types - *survival, counts, and classification* - to evaluate the potential of each approach and the validity of their assumptions.

5.3.2 The Cox proportional hazards model

In this subsection, we explore the Cox Proportional Hazards model [82]. The Cox Proportional Hazards (CPH) model is a very popular regression model that calculates survival times based on the effect of selected predictors. It becomes especially useful here since our predictors are (highly) non-linearly related and we may not know their distributions beforehand. Another advantage of the CPH model is the fact that it is able to handle missing observations, i.e. sparse user interactions. The CPH model only requires as independent parameters (i) the time of the analysis and (ii) the engagement status. In our case, the status indicates if disengagement happened or not at any particular time. With these two parameters, we estimate two functions called conditional survival and baseline hazard. The former provides the probability of not experiencing disengagement while the latter gives the probability that disengagement will occur up to a given time - see [83]. In our context, the term *proportional hazard* indicates that the hazard ratio comparing two observations is constant in between events. Furthermore, the impact of the different factors on the hazard remains constant over time [33]. We use a threshold equal to 0.5 to determine engagement/disengagement and follow [150] to ensure monotonic Receiver Operating Characteristic (ROC) curves by means of the nearest neighbour method.

5.3.3 The negative binomial model

Here, we describe a regression model with count data (negative binomial model). This approach is interesting because, in contrast to the CPH model, it allows us to model re-engagement. Here, rather than the time of disengagement, we aim to predict the total number of actions before disengagement. The idea is to target smart interactions aiming to keep the user engaged if the actual counts fall too close from the prediction of disengagement. Briefly, the negative binomial (NB) distribution is the distribution of the number of trials (actions) needed to get a fixed number of failures (in our case disengagement) - see [205]. This distribution describes the probabilities of the occurrence of integers greater than or equal to 0. By analysing the distribution function, we can set a threshold on the probability of disengagement and extract the number of counts before disengagement. NB is specially suitable to model over-dispersed count variables. This specific regression method is implemented by fitting a generalised linear model using a boosting algorithm based on component-wise univariate linear models - see [52; 50], and [51]. In each boosting iteration, a simple linear model is fitted (without intercept) to the negative gradient vector and in the update step only the best-fitting linear model is used. This machine learning method optimises prediction accuracy and carries out variable selection. In our case, we perform **500** non-centered boosting iterations with a step length equal to **0.05**.

5.3.4 The random forest model

The RF model [43] basically creates many random independent subsets of the dataset containing features and a training class. In our case, the features are the information about the user, e.g. number of interactions and type of interaction, and the class is simply a flag indicating engaged or disengaged at that particular moment. These subsets are used to create a ranking of classifiers. It is important to state that RF models are typically accurate and computationally efficient. The randomness component ensures the RF model to generalise well, and to be less likely to overfit [201].

In contrast to the other approaches, the RF model is not predicting days (CPH) or counting actions to disengagement (NB). Here, based on past behaviour, we use the RF algorithm as a classifier (engaged/disengaged) *at the moment*. That means, we obtain as outcome a probability value ranging between

0 and **1**. With that in hand, we define a cutoff threshold to determine if the user is engaged or disengaged. For our dataset, the cutoff threshold is chosen as equal to **0.42** as it maximises the F1 score [138].

Interestingly, the RF classification has a predictive component. This is because the RF model simulates $\text{UE}_{\boldsymbol{R}}$. As shown in Section 5.2.2, this metric is defined as the difference in days between an action now and in the next 9 days. Due to that, we assume that our results are "valid" not only *at the moment* but within the recency threshold as well. Note that this links the validity of the RF model to the recency threshold. This further motivates the choice of a short recency time, just enough to allow the app developer to send re-engagement notifications and monitor their effectiveness.

To build this random forest model, we use **1000** non-stratified trees with replacement (to decrease variance without increasing bias). The number of variables randomly sampled as candidates at each split equal to **10**. We use a 10-fold cross validation with **5** repeats to augment model accuracy without increasing bias. The cross validation involves splitting our dataset into **10** subsets. Each subset is then put apart and the model is trained on the leftover subsets. The overall accuracy of our model is then determined after averaging the results obtained with the **5** individual repeats.

5.3.5 The XGBoost Model

The last approach used to predict UE takes advantage of boosted-trees algorithms. XGBoost is a very popular and scalable end-to-end tree-boosting system [67] currently applied to several different fields of knowledge, such as Physics, stock market prediction, biology and language networks, among others [68; 100; 350; 64]. In a nutshell, this classifier constructs trees to make the predictions, but unlike RF, where every tree provides a definite answer and the final result is obtained by a voting process (*i.e.,* bagging), every tree in XGBoost contains a continuous score, which are combined to provide an answer (*i.e.,* boosting). Despite differences with the RF algorithm, the implementation and the use of XGBoost, however, is done very similarly. We utilise the same features to train the model and the output is also a probability percentage indicating whether the user is disengaged *at the moment*. We use a small learning rate equal to 0.001 to ensure convergence and error minimisation. The maximum depth of each tree is capped at 15 and the maximum number of trees is fixed at 1000 (similar to RF).

5.4 Evaluation of Predicting User Engagement of Mobile Apps

In this section we report on the empirical evaluation of the proposed modelling framework *in the context of a waste recycling mobile app*. Specifically, we aim at answering the following research questions:

- $\boldsymbol{RQ_1}$ – To what extent using a clustering algorithm impacts the accuracy of UE prediction?
- $\boldsymbol{RQ_2}$ – Which types of numerical models provide the most accurate UE prediction?
  - $\boldsymbol{RQ_{2.1}}$ – What is the prediction accuracy of the Cox proportional hazards model?
  - $\boldsymbol{RQ_{2.2}}$ – What is the prediction accuracy of the negative binomial model?
  - $\boldsymbol{RQ_{2.3}}$ – What is the prediction accuracy of the random forest model?
  - $\boldsymbol{RQ_{2.4}}$ – What is the prediction accuracy of the XGBoost model?

We begin by showing the performance of our AHC algorithm followed by the predictions of UE for our other numerical models. We highlight that a direct comparison between numerical models is not always possible due to their different natures - classification and regression. Thus, we aim to characterise and evaluate them mostly individually. When possible, we try to place our results in a broader perspective. To keep to the brief character of this manuscript we summarise our model results in terms of ROC curves [110]. These are plots that illustrate the performance of a binary classifier, outlining their overall performance. The true positives are defined as the engaged users who were correctly classified as engaged by our model. False negatives represent the engaged users incorrectly classified as disengaged. The area under the ROC curve (AUC) represents the model accuracy, where unity means a perfect model and 0.5 indicates a random result. We use the ROC curve as our performance indicator - similarly to [257] - because it evaluates the performance of the models across all possible thresholds. In addition, AUC delivers a result comparable across all our model approaches and is threshold independent. This is important in our case since the impact of a false positive vs false negative is comparable.

5.4.1 Impact of the clustering model (RQ_1)

Implementing the weight-optimised Gower's metric - as described by [358] - augments the CPCC by around 15% (from 0.84 to 0.97) if compared with the case where all weights are set to unity. We calculate the Pearson-γ correlation for our dataset to further investigate the benefits of our optimised clustering methodology. The results are shown in Figure 5.3.

Implementing the optimised weights for Gower's metric increases the Pearson-γ correlation by around 11%. That, together with the 15% improvement in the CPCC, indicates that our methodology to optimise weight works significantly better than the standard procedure. We note a slight decrease in the Pearson-γ correlation for the AHC optimised results at 4 clusters followed by a sharp decrease at 13 clusters. From the 13 clusters with a high Pearson-γ correlation, 4 main clusters contain around 98% of the total amount of unique users. Nevertheless, we include all 13 clusters in our analysis to ensure that these outliers do not influence these main 4 clusters.

5.4.2 Prediction accuracy of the Cox proportional hazards model ($RQ_{2.1}$)

In Figure 5.4 we show our results for the CPH model. We do so, by means of a ROC plot for four different time spans within the testing set.

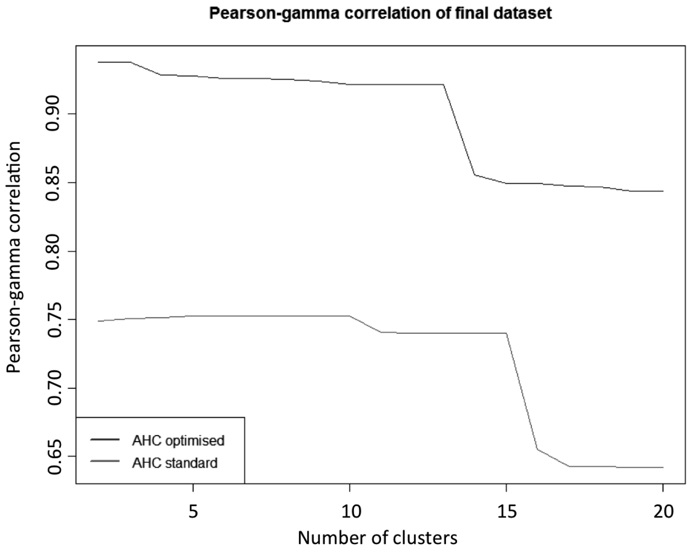


Figure 5.3: *Pearson-γ correlation for the AHC - optimised (blue) and standard (red) - against the number of clusters.*

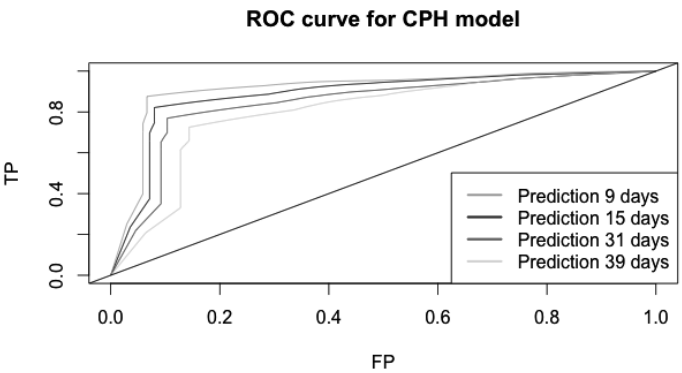


Figure 5.4: *ROC curves for the CPH model based on the testing set. TP and FP indicate true positive and false positive, respectively. The legend indicates the different time spans.*

We observe in Figure 5.4 the predictions for increasing time spans. As expected, the ROC curves approach the diagonal line (random prediction) as we move forward in time. Note that these predictions are based on the testing set, and not yet on the validation sets. That is because, at this stage, we are interested in the generalisation capabilities of this model. We explain: these ROC curves are derived from the survival chance as a function of time. This means 100% survival chance for day 0, decaying eventually to 0% as time progresses (Kaplan-Meier curve). Based on the these probabilities, the ROC curves are generated within the testing set as an universal discrete prediction for the CPH model from 9 to 39 days. We see that both short- and long term predictions are accurate. The AUC ranges from 0.8 to 0.91 for 39 and 9 days, respectively.

5.4.3 Prediction accuracy of the negative binomial model ($RQ_{2.2}$)

Figure 5.5 presents the ROC curves for the NB model. Contrarily to the CPH model, the NB model predicts actions until disengagement. That means it would be fairly impossible to create a binary classifier able to estimate the exact number of actions before disengagement. Instead, we use 5 counts per 14 days as a threshold to determine if a user is engaged or not. In this case, a user is considered engaged if exceeding the threshold. Nevertheless, we note that the outcome is inferior compared to the results obtained by the CPH model. Due to the unexpected results for the testing set, we also analyse the performance of the NB model for the validation sets. The results, shown in Figure 5.5, remain reasonably similar to the ones obtained for the testing set. The AUC is fairly constant and equal to 0.67 for all the sets.

Figure 5.6 presents the number of events observed and predicted by the model to further understand the performance of the NB model.

Besides the fact that some of the predictions coincide with the observations, a very significant part of the observed values is crudely underestimated by the model. That means the model is able to reasonably predict the so-called "true

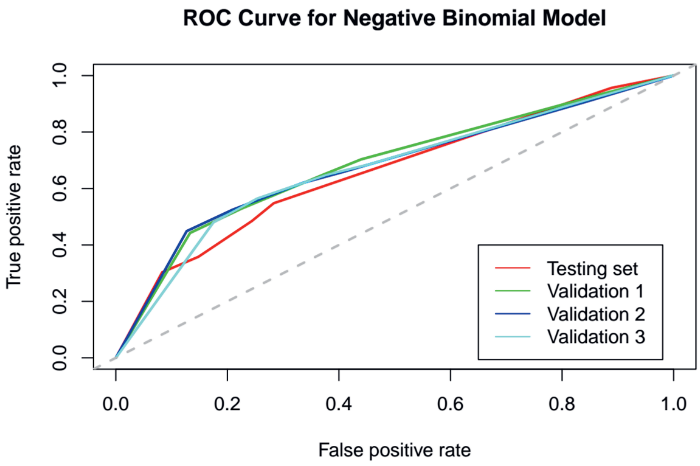


Figure 5.5: *ROC curves for the NB model. The legend indicates the datasets. The timespan is fixed to 14 days.*

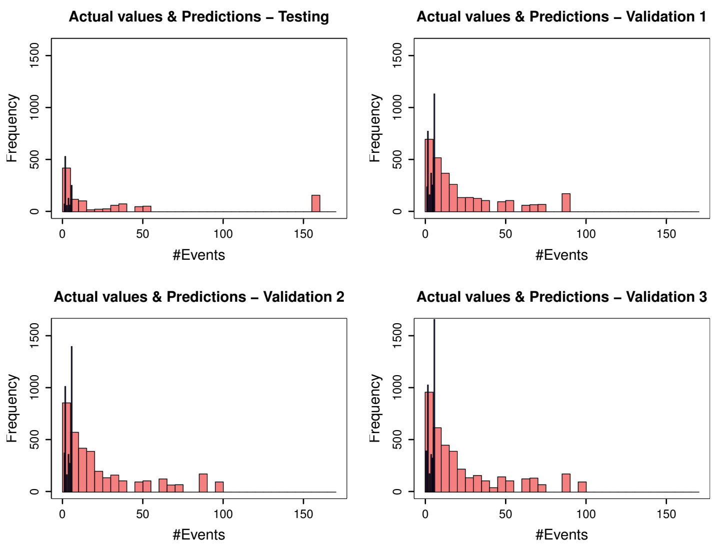


Figure 5.6: *Comparison between the number of events predicted (black) and observed (red) for the different sets, as indicated in the headers. The time span is fixed to 14 days.*

positive" values but fails to predict the "true negative" ones. These results suggest that this model is, to a certain extend, accurately predicting the right counts to disengagement, albeit with many inaccurate predictions included as well.

5.4.4 Prediction accuracy of the random forest model ($\boldsymbol{RQ_{2.3}}$)

In Figure 5.7, we visualise the ROC curves for the RF model applied to the different sets. We find that the AUC ranges from 0.93 to 0.83 for the *testing* and *validation 3* sets, respectively. The high AUC values mean that the RF model is generic enough to classify our user as engaged or disengaged for all our dataset.

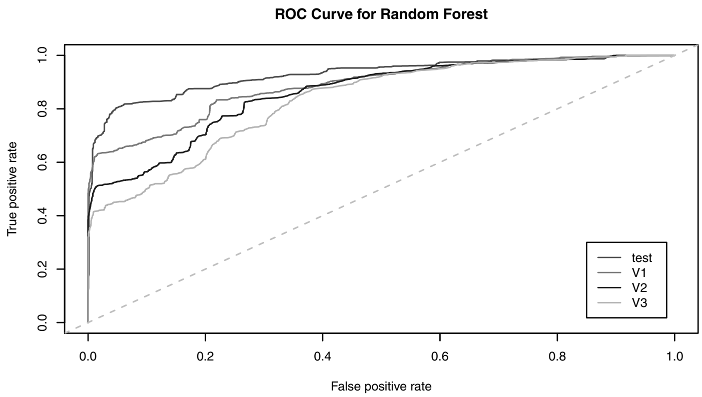


Figure 5.7: *ROC curves for the RF model. The test set curve is shown in red, followed by the Validation 1,2, and 3 sets in green, blue, and cyan, respectively.*

To further understand which processes/features determine the behaviour of this model, in Table 5.1 we show the mean decrease in accuracy (MDA) for some of the predictors. The MDA is calculated by permuting the values of each predictor and then measuring by how much the predictive accuracy decreases.

In our case, removing *groups*, *number of actions*, *longitude*, or *weekday*,

Table 5.1: *Selection of predictors and their respective mean decrease in accuracy (MDA)*

Predictor	**MDA** (%)
Groups	36.5
Number of actions	35.5
Longitude	34.0
Weekday	33.5
Latitude	27.0
Observation time	21.0

from the predictors list would decrease the accuracy of this model by over 30%. We point out to the reader that the MDA is computed after the RF is trained. Therefore, training the model without these predictors will not drop the performance by the amounts shown in Table 5.1. Instead, the new model may find new correlated features unknown to the current model. We also notice in Table 5.1 the importance of adequately clustering users since *groups*, calculated with the optimised AHC algorithm, is responsible for the highest MDA value.

5.4.5 Prediction accuracy of the XGBoost model ($RQ_{2.4}$)

Figure 5.8 presents the XGBoost curves for the different sets. The AUC range is virtually the same as the one for the RF, with the values from 0.93 to 0.82 for the *testing* and *validation 3* sets, respectively.

To keep our comparison similar to that of the RF we have selected the same predictors and seen if there was any difference in their relative importance distribution. To calculate their importance we examined the “Gain” value. Interestingly, we see that the order of the importance remains the same as per the RF with “groups” being the predictor with the highest Gain value (0.06) and “obs time” with the lowest (**0.0001**). That reinforces the importance of having well-defined and accurate groups as output from the clustering algorithm.

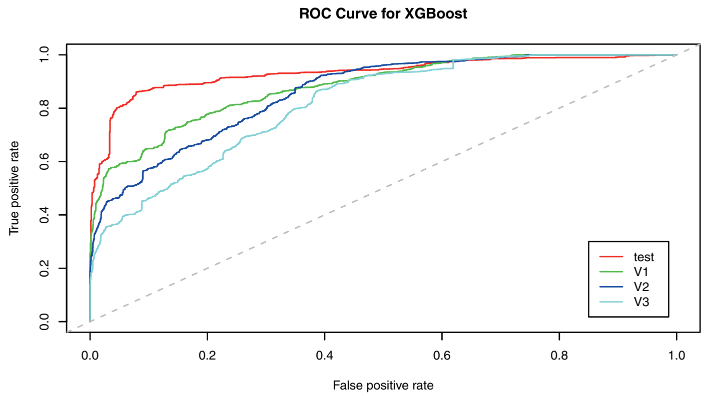


Figure 5.8: *ROC curves for the XGBoost model. The test set curve is shown in red, followed by the Validation 1,2, and 3 sets in green, blue, and cyan, respectively.*

5.5 Discussion

Concerning $\boldsymbol{RQ_1}$, the modified clustering algorithm containing optimised weights for Gower's metric performed adequately. The results showed an improvement of $\approx$ 11% on the Pearson-γ correlation, and $\approx$ 15% on the cophenetic correlation, if compared to a standard clustering methodology. The clustering outcome proved to be the most important predictor for both RF and XGBoost algorithms. That provides further motivation to optimise the clustering process aiming at sharpening the groups definition and as a consequence improve the machine learning results.

Concerning $\boldsymbol{RQ_2}$, we applied the four models on the dataset and analysed the results obtained, mainly via the use of ROC curves. All models performed well, in their own right, with Cox proportional hazards, random forest and the boosted-tree models resulting in similar performance when predicting UE. The performance of the negative binomial model was not comparable to the other three algorithms. Most importantly, we concluded that under this framework we were able to better understand our observations.

As shown in Section 5.4, CPH, RF and XGBoost models result in similar values of accuracy. Their AUC values are similar, ranging roughly from 0.8 to 0.9. Our fourth model, the NB model, resulted in an AUC of 0.67. It is important to re-iterate that this AUC values should be taken as individual measures of performance and not used to compare models, as the manner of predicting and even the element of prediction is different according to the algorithm used.

Even with a high AUC score, there are still, however, a number of caveats concerning the generalisation of the CPH model. More specifically, the results obtained with this model vary significantly for different sets of predictors. Interestingly, the good results found by the RF and XGBoost models can be partially explained by their generality. We will take advantage of this feature and use these models to "classify" UE in the future as well.

We are also interested to model re-engagement. Given the fact that the CPH model is unable to do so (since it predicts survival times), a Markov-like stochastic model becomes then a plausible replacement. The reason is that these models are able to provide the transition paths between engaged-disengaged and to obtain the rate parameter of these transitions. We emphasise that the RF and XGBoost models are also able to model re-engagement. In the near future, we aim to compare in detail the results obtained by the RF and XGBoost, with the transition model.

Finally, it is important to note that the accuracy we obtained in our evaluation is specific to the dataset related to the waste recycling app and cannot be directly transferred to other mobile apps or application domains. Indeed, the aim of this study is not to provide a general solution for all mobile apps in all domains, but rather, we focus on (i) providing evidence that it is possible to predict when app users are getting disengaged with good levels of accuracy and (ii) providing a reusable modelling framework for UE in mobile apps. Researchers and practitioners in application domains other than waste recycling can re-use our proposed framework and its underlying techniques, provided that they will be customised according to (i) the characteristics of their specific app domain (*e.g.,* a user of a social media app may be considered as disengaged after 1 day of inactivity, instead of 9 days) and (ii) the performance of the trained models

(*e.g.,* in a different domain the negative binomial model may perform the best.)

5.6 Summary and Future Work

In this chapter we answer ***T.RQ*3**, specifically: *How can we predict user engagement in apps?* We achieve this result by proposing and evaluating a framework to model and predict user engagement in mobile applications. The framework consists of a modified clustering model that serves as baseline for other four numerical models: (i) a Cox proportional hazards, (ii) a negative binomial, (iii) a random forest, and (iv) a boosted-tree algorithm. These models were trained and validated against an observational dataset obtained from a real waste recycling mobile application. Our results show that *in our case* both machine learning approaches (RF and XGBoost) are adequate to model user engagement for the considered app. In this study, we tested our framework on data obtained from a waste recycling app. Hence, our findings would likely remain valid only for applications with usage dynamics and features similar to those within the waste recycling domain. Specifically, geographical information plays a crucial role to determine different user behaviours (as seen in our study) and hence a successful application of this methodology to a different domain would most likely be dependent of a strong tie to location. As an example, domains such as fitness or language learning [15; 311], tend to have daily activities presented to the user in a game-like manner and have a strong tie to the user's location. Given these were key features used to train our models, it is plausible that this framework could be applied to these and other similar domains.

Analysing user behaviour to predict and prevent disengagement certainly poses a significant challenge, both from the methodological and analytical points of view. Due to the complexity of this task, we limited this study to characterising and evaluating our methodology to *predict* UE. In a follow-up study, we will investigate how to ultimately influence user behaviour by increasing re-engagement rates and decreasing disengagement. Moreover, further research will touch upon studying the re-engagement *process.* We then intend to use push notification information - extending on the work of [319] - to ultimately determine the most appropriate interaction for each user at any given time, aiming to augment usage (maintain engagement) and prevent disengagement. Understanding the role gamification plays in mobile apps is also crucial. It can be done by further investigating how people redeem their points earned (e.g. immediately after achieving a minimum threshold or after some accumulation). That information helps in determining the type of notification that can be sent to each user.

Acknowledgements

All the authors thank the Amsterdam Network Institute for partially funding this research and Coen Jonker for his remarks on the initial version of the manuscript. Eduardo Barbaro thanks Mobiquity Inc. for the support and the hours available throughout the research.

Part II

Self-adaptation in mobile applications

6

Self-adaptation in Mobile Applications

Chapter 6 was published as:

Grua, E. M., Malavolta, I., & Lago, P. (2019, May). Self-adaptation in mobile apps: a systematic literature study. In 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) (pp. 51-62). IEEE.

Abstract - With their increase, smartphones have become more integral components of our lives but due to their mobile nature it is not possible to develop a mobile application the same way another software system would be built. In order to always provide the full service, a mobile application needs to be able to detect and deal with changes of context it may be presented with. A suitable method to achieve this goal is self-adaptation. However, as of today it is difficult to have a clear view of existing research on self-adaptation in the context of mobile applications.

In this chapter, we apply the systematic literature review methodology on selected peer-reviewed papers focusing on self-adaptability in the context of mobile applications. Out of 607 potentially relevant studies, we select 44 primary studies via carefully-defined exclusion and inclusion criteria. We use known modelling dimensions for self-adaptive software systems as our classification framework, which we apply to all selected primary studies. From the synthesised data we obtained, we produce an overview of the state of the art. The results of this chapter give important background information, which is used to answer ***T.RQ*4**. Furthermore, these results also give researchers and developers a solid foundation to plan for future research and practice on engineering self-adaptive mobile applications.

6.1 Introduction

Since the announcement of the iPhone in 2007 and the sale of Android based smartphones, the number of mobile applications has been increasing and so is the number of mobile users [195; 328].

With their increase, smartphones have become more integral components of our lives but do to their mobile nature it is not possible to develop a mobile application the same way another software system would be built. In order to always provide the full service, a mobile application needs to be able to detect and deal with changes of context it may be presented with. A suitable method to achieve this goal is self-adaptation [277].

While there is a lot of work that has been done in the field of self-adaptation

[387], to the best of our knowledge there is no published literature review that explores self-adaptation in the specific context of mobile applications. Within this context we can identify several questions related to the most common goals that the self-adaptive systems are aiming to achieve, what kind of changes can trigger adaptation processes, how is it achieved in current published work and what would be the outcomes and effects of the adaptation to the mobile application. Unveiling the above mentioned aspects will give a better understanding of the current landscape of self-adaptation for mobile applications.

In this study we aim to fill the knowledge gap present with self-adaptive systems in the context of mobile applications. To do so, we apply the systematic literature review methodology [180] and target peer-reviewed papers focusing on self-adaptability in the context of mobile applications. Out of 607 potentially relevant studies, we select 44 primary studies via carefully-defined selection criteria. We then utilise and customise known modelling dimensions for self-adaptive software systems [9] and use them as our classification framework, which we apply to all selected primary studies.

Obtained **results** reveal that the most common sources of change are hardware (which includes the battery of the device) and the internet connectivity. Most analysed approaches perform the self-adaptation in an autonomous manner and adaptation happens within the application itself, with sometimes the use of the backend (*e.g.,* cloud offloading systems). Furthermore, in all primary studies adaptation is event triggered and performed in a best-effort manner, without a strict guarantee on the duration of the self-adaptation process. Most of the approaches are not specific to any application domain, with a lack of case study evaluation.

The **main contributions** of this study are:

- an up-to-date *systematic review* of the literature on self-adaptation in the context of mobile applications;
- a customised *classification framework* for understanding, classifying, and comparing approaches for self-adaptation in the context of mobile apps;

- a *discussion* of the main implications of this study, the application domains covered by the literature so far, and future research challenges;
- a *replication package* including the research protocol, raw data, and analysis scripts for independent replication and verification of this study.

The **target audience** of this chapter includes: researchers working in the field of self-adaptation and want to have better insight of the literature when specifically dealing with mobile applications, researchers and mobile application developers looking to implement self-adaptation in their system but do not have prior experience in the field and need a guide to understand what has been done so far.

The rest of the chapter is organised as follows. In Section 6.2 we give background information on self-adaptation in mobile applications. Section 6.3 explains the study design, whereas its results are reported in Section 8.3. In section 6.5 we provide a discussion of the emerging results, followed by section 8.5 in which we present threats to validity. Section 6.7 presents related work and lastly we close the chapter in section 6.8.

6.2 Self-Adaptation in Mobile Applications

Figure 6.1 shows an overview of a **mobile-enabled system**. We use the entities shown in the figure to settle with the terminology used throughout the paper. Mobile apps consist of binary executable files that are downloaded directly into the user's device and stored locally [222]. Mobile apps are developed directly atop the services provided by their underlying mobile platform [224]. Platform services are exposed via a dedicated Application Programming Interface (API) and provide functionalities related to communication and messaging, graphics, location, security, etc. [119]. Moreover, the platform API abstracts and provides access to the hardware components of the device such as its proximity sensor, GPS, accelerometer, battery, networking devices, and so on. Apps can also communicate with other apps installed on the device via a dedicated event-based communication system (*e.g.,* Android intents and broadcast receivers[1]). Smart objects such as fitness trackers, smart headphones, external sensors, and smartwatches can be connected to the mobile device either via short-range communication protocols (*e.g.,* Bluetooth) or by passing through the Internet.

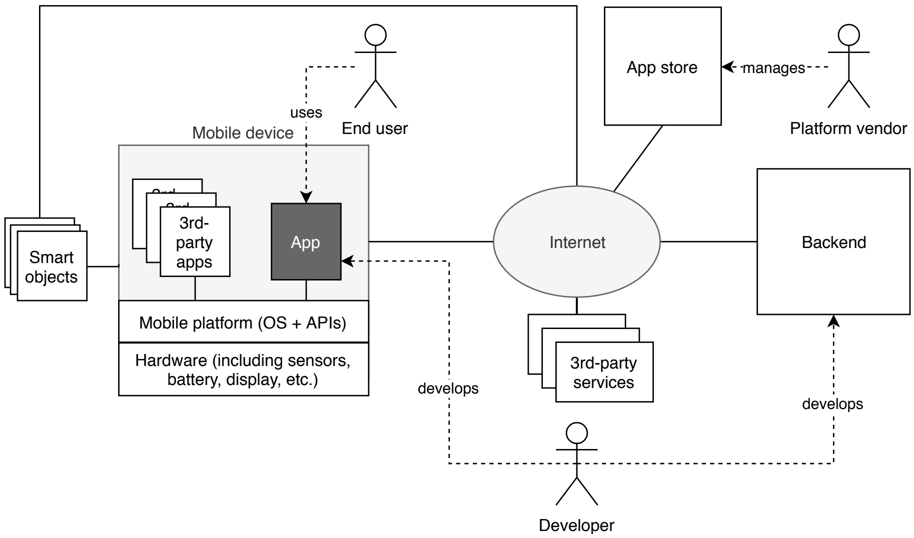


Figure 6.1: *Overview of a mobile-enabled system*

The vast majority of mobile apps send and receive data to their remote backends in order to persist data across usage sessions, share data across apps instances, etc. The communication between the app and its backend is usually performed in a RESTful fashion via the HTTP protocol [1; 214]. Similarly, apps can also communicate with 3rd-party services, for example for authentication via Facebook, accessing the mapping services of Google Maps, sharing data to social networks. Mobile apps are distributed via dedicated app stores, such as the Google Play Store for Android apps and the Apple app store for iOS apps. App stores are managed by platform vendors like Google and Apple [119].

Self-adaptation can happen in any part of a mobile-based system (*e.g.,* in the app itself, in the backend, in a smart object) and can be applied to different

[1] https://developer.android.com/guide/components/broadcasts

levels of the technology stack of computing systems (*e.g.,*, at the hardware level, at the platform level, in the business logic of the app). In the context of this study, a self-adaptive system is defined as a system that can autonomously handle changes and uncertainties in its environment, the system itself and its goals [373]. A self-adaptive system is internally composed of two parts: one has the responsibility of performing the business capabilities of the system (*i.e.,* the operations for which the system is built), whereas the second part interacts with the first one and is responsible for the adaptation process [373]. As an example of a self-adaptive mobile application we take the one described by Moghaddam et al. [243]. The work describes a framework built to enhance energy efficiency in mobile apps. The framework was built with the MAPE model functionalities and consists of a scheduler that is in charge of allocating resources in real-time. In this particular implementation case the authors focus on the network scheduling strategies.

6.3 Study Design

In this section we present the design of this study. We firstly present the research questions (Section 6.3.1) and then we explain our search and selection process in Section 6.3.2. We report on the data extraction process and the framework used to classify the information extracted from our primary studies in Section 6.3.3. Lastly, we explain how we synthesised the main findings from the extracted data in Section 6.3.4.

The *replication package* is publicly available to researchers interested in replicating and independently verifying the study [142]. The replication package includes the raw data of the search and selection phases of the study, the raw data extracted from each primary study, and the full list of all primary studies.

6.3.1 Goal and Research Questions

Below we show the formalisation of the goal of this study according to the Goal-Question-Metric approach [54].

Purpose	Identify, classify, and evaluate
Issue	the characteristics
Object	of existing approaches for self-adaptation in mobile apps
Viewpoint	from the researcher's and practitioner's point of view.

By building on the modelling dimensions for self-adaptive software systems proposed by Andersson *et al.* [9], we can elicit the following research questions targeted by our study.

RQ1 *What are the goals of self-adaptation in the context of mobile apps?*
Answering this question we aim to identify the characteristics of the goals that self-adaptation should achieve in the context of mobile apps. As an example, the the self-adaptation mechanism proposed in the primary study by Moghaddam *et al.* [243] has the main goal of *reducing the energy consumption* of the application in order to prolong the smartphone's battery life.

RQ2 *What are the changes triggering the self-adaptation in the context of mobile apps?*
By answering this question we want to gain insight on the characteristics of the changes triggering self-adaptation in mobile apps. For example, referring back to the previously mentioned study by Moghaddam *et al.*, one of the possible sources of a change is the event in which a new application requests to transfer data since it requires adaptation on resource scheduling within the whole mobile device.

RQ3 *What are the mechanisms used for self-adaptation in the context of mobile apps?*
Answering this question will allow us to better understand the characteristics of the mechanisms for self-adaptation within the context of mobile applications. For example, in the case of Moghaddam *et al.*, the mechanism for adaptation is *structural* since adaptation involves the reconfiguration of the overall architecture of the whole system.

RQ4 *What are the effects of self-adaptation in the context of mobile apps?* By answering this question we will gain better understanding of what are the effects of self-adaptation upon mobile-enabled systems. In this context, a dimension for judging the effect of self-adaptation is by understanding its criticality, *i.e.,* the impact that the self-adaptation process would have on the mobile application in case said adaptation fails. For example, returning to the primary study by Moghaddam *et al.* the criticality of the self-adaptation process is *harmless* since the mobile app is able to function even if the adaptation fails (the only downfall would be the continuation of the current use of energy, instead of reducing it).

The research questions shape the whole study, with a special influence on (i) search and selection of primary studies, (ii) data extraction, and (iii) data synthesis.

6.3.2 Search and Selection

As shown in Figure 6.2, the search and selection process of this study has been designed as a multi-stage process, so to have full control over the studies being considered during the various stages.

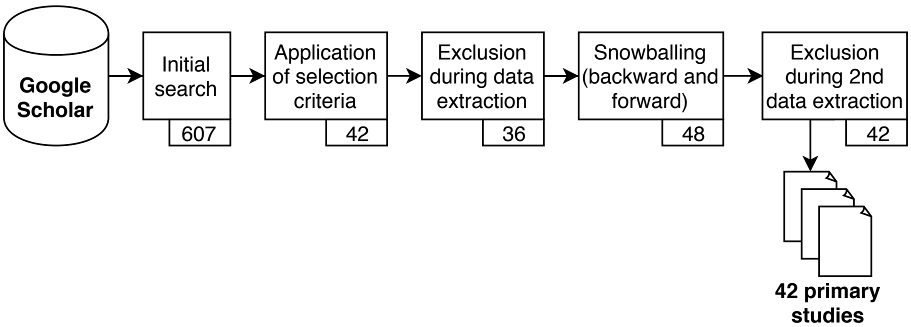


Figure 6.2: *The search and selection process of this study*

Initial search. In this stage we perform an automated search on *Google Scholar*, which at the time of writing is one of the largest and most complete databases and indexing systems for scientific literature. We use such a data source for the following main reasons: (i) it provides the highest number of potentially relevant studies compared to other four relevant libraries (Scopus, ACM Digital Library, IEEE Explore, and Web of Science), (ii) as reported in [379], the adoption of this indexer has proved to be a sound choice to identify the initial set of literature studies for the snowballing process, (iii) the query results can be automatically extracted from the indexer. The query we use to perform the initial search is provided in Listing 1

```
(adaptive OR "self-adaptation" OR "self-adaptive")
AND (android OR ios OR mobile)
AND (apps OR applications OR application))
```

Listing 6.1: *Search string used for the automatic search*

In order to cover as much potentially relevant studies as possible, we kept our search string as generic as possible and considered exclusively the object of our research. Indeed, the search string can be divided into three main components, one for each line of the listing, where the first component captures self-adaptive systems, the second captures the mobile nature of the targeted approaches, and the third one is about apps and applications. The search string has been tested by executing pilot searches on Google Scholar. In order to keep the results of this initial search as focused as possible, the query has been applied to the title of the targeted studies. The considered timeframe ranges from 2007[2] and ends at the time in which the query has been executed (*i.e.,* November 2018).

Application of selection criteria. In this stage we consider all the 607 studies resulting from the initial search and filtered them according to a set of well-defined inclusion and exclusion criteria. In this stage it is crucial to select studies objectively and in a cost-effective manner, so we apply the adaptive reading technique [275], as the full-text reading of clearly excluded studies is not necessary. In the following we report the inclusion and exclusion criteria of this study.

[2] The first announcement about the existence of mobile apps as defined in Section 6.2 has been done in the well-known keynote where Steve jobs firstly launched the iPhone in 2007 [38].

Inclusion criteria:

1. The study focuses on self-adaptability, as defined in [9].

2. The study focuses on mobile applications, as defined in Section 6.2.

Exclusion criteria:

1. Secondary or tertiary studies (*e.g.,* systematic literature reviews, surveys, etc.).

2. Studies in the form of editorials, tutorial, poster papers, because they do not provide enough information.

3. Studies that have not been published in English language.

4. Duplicate papers or extensions of already included papers.

5. Studies that have not been peer reviewed.

6. Papers that are not available, as we cannot inspect them.

Each paper is included as primary study if it satisfies *all* inclusion criteria, and it is discarded if it meets *any* exclusion criterion. The definition of the above mentioned criteria has been incrementally refined and tested by two researchers by considering a set of pilot studies. It is important to note that we excluded secondary studies because of the first exclusion criterion, but we discuss them in our related work section (see Section 6.7).

Exclusion during data extraction. When going through each primary study in detail for extracting information, we agreed that 6 analysed studies were semantically out of the scope of this research and we excluded them, leading to a set of 36 potentially relevant studies.

Snowballing. To reduce potential bias introduced with the use of our selected search string we also carry out a snowballing process[141]. The main goal of this stage is to enlarge the set of potentially relevant studies by considering each study selected in the previous stages, and focusing on those papers either citing and cited by it. More technically, we perform a closed recursive backward and

Attribute	*Possible values*	*Definition*
Goals - goals are objectives the system under consideration should achieve		
Quality requirement	functional suitability (FUN), performance efficiency (PERF), compatibility (COMP), usability (US), reliability (REL), security (SEC), maintainability (MAINT), portability (PORT), energy (EN), any or all of the possibilities (?)	The system/software quality the goal is aiming to achieve
Evolution	static (S), dynamic (D)	Whether the goals can change within the lifetime of the system
Flexibility	rigid (R), constrained (C), unconstrained (D)	Whether the goals are flexible in the way they are expressed
Duration	temporary (T), persistent (P)	Validity of a goal thoroughout the system lifetime
Multiplicity	single (S), multiple (M)	How many goals are there?
Change - Change is the cause of the adaptation		
Source	APP, 3rd party app (3A), mobile platform (PLAT), hardware (HW), smart objects (SMARTO), end user (USER), backend (BACK), 3rd party services (3S), developer (DEV), app store (STORE), platform vendor (VENDOR), INTERNET	Where is the source of the change?
Frequency	rare (R), frequent (F)	How often a particular change occurs?
Anticipation	foreseen (FN), foreseeable (FE), unforeseen (UN)	Whether the change can be predicted
Mechanisms - what is the reaction of the system towards change		
Type	parametric (P), structural (S)	whether adaptation is related to the parameters of The system components or to the structure of the system
Autonomy	autonomous (AU), human assisted (H)	What is the degree of the outside intervention during adaptation
Organization	centralized (C), decentralized (D)	whether the adaptation is done by a single Component or distributed amongst several components
Scope	APP, 3rd party app (3A), mobile platform (PLAT), hardware (HW), smart objects (SMARTO), end user (USER), backend (BACK), 3rd party services (3S), developer (DEV), app store (STORE), platform vendor (VENDOR), INTERNET	Where in the system is the adaptation localized
Duration	short (S), medium (M), long (L)	How long the adaptation lasts
Timeliness	best effort (B), guaranteed (G)	Whether the time period for performing self-adaptation can be guaranteed
Triggering	event-triggered (E), time-trigger (T)	Whether the change that triggers adaptation is associated with an event or a time slot
Effects - What is the impact of adaptation upon the system		
Criticality	harmless (H), mission-critical (M), safety-critical (S)	Impact upon the system in case the self-adaptation fails
Predictability	non-deterministic (N), deterministic (D)	Whether the consequences of the adaptation can be predictable
Overhead	insignificant (I), reasonable (R), failure (F)	The impact of system adaptation upon the quality of services of the system
Resilience	resilient (R), semi-resilient (S), vulnerable (V)	The persistence of service delivery that can justifiably be trusted, when facing changes

Table 6.1: *Classification framework utilised for the data extraction*

forward snowballing activity [380]. In both backward and forward snowballing the initial screening of additional studies is based on their title only, whereas the final decision about their inclusion into the set of primary studies is based on their full text and on the selection criteria discussed above. Duplicates have been removed at each iteration of the snowballing activity.

Exclusion during 2nd data extraction. In this phase we extract data from the 12 additional papers resulting from the snowballing activity and agree that 6 of them are semantically out of the scope of this research and we exclude them. This final check leads to the final set of 42 primary studies, which are then analysed in details for answering our research questions.

6.3.3 Data Extraction

In this section we present how we perform the data extraction on the selected primary studies. The main goal of this phase is to collect data from each primary study, so to be able to suitably compare them in the subsequent data synthesis phase. The data extraction phase is executed collaboratively by two of the authors of this study. In order to have a rigorous data extraction process and to ease the management of the extracted data, a well-structured classification framework has been designed upfront.

As anticipated, we build our classification framework on the modelling dimensions for self-adaptive software systems presented by Andersson *et al.* [9]. In order to better fit the framework to the characteristics of mobile applications, we *customise* the modelling dimensions for self-adaptive systems presented in [9]. The customisation of the classification framework has been performed as follows: (i) firstly we selected a subset of 10 pilot studies from the 42 primary studies, (ii) then two researchers independently extracted the data from the 10 pilot studies by using the original version of the self-adaptation modelling dimensions proposed in [9], (iii) the two researchers then discussed the results of the data extraction, with a special focus on too generic/abstract attributes, those attributes which did not fully fit with the characteristics of the primary studies, attributes whose values were redundant, (iv) based on the discussion, the self-adaptive modelling dimensions have been customised into the final version of the classification framework, and lastly (iv) the final version of the classification framework has been applied to all 42 primary studies.

The customised classification framework is presented in Table 6.1. Specifically, as part of the Goals dimension, we add an attribute called *quality requirement* so to keep track of the system/software quality requirement that self-adaptation aims to achieve [166] (*e.g.,* security, usability, functional suitability, etc.). Here we are not considering how the (potentially multiple) goals of self-adaptation are related to each other, *i.e.,* the *dependency* attribute, as this attribute resulted to be too fine grained for the objective of our study. For the Changes dimension, we extend the attribute *source* so to directly map it to the main elements of mobile-enabled systems as they are depicted in Figure 6.1. Also, we are not considering the attribute *type*, which originally was distinguishing between functional or non-functional changes, since we notice that in our primary studies it was strictly contained by the *quality requirement* attribute. For the Mechanisms dimension, the only change we made is related to the extension of the attribute *scope*, which we are also mapping to the main elements of Figure 6.1. Lastly, we reuse the Effects dimension just as defined in the original framework presented by Andersson *et al.*

6.3.4 Data Synthesis

The data synthesis activity involves collating and summarising the data extracted from the primary studies [180] with the main goal of understanding, analysing, and classifying current research on self-adaptation in the context of mobile applications. Specifically, we performed a combination of *content analysis* (for categorising and coding the studies under broad thematic categories) and *narrative synthesis* (for explaining in details and interpreting the findings coming from the content analysis). This phase is performed by all of the authors of this study.

6.4 Results

In this section we report the results in the context of all the research questions of our study. On a technical note, some of the plots (*e.g.,* subfigure 6.3a) contain the '?' bin. The meaning of the symbol is that the examined primary studies in that bin are configurable by developers/users so to fit a variable number of the bins of that category, hence we have classified it as a separate category.

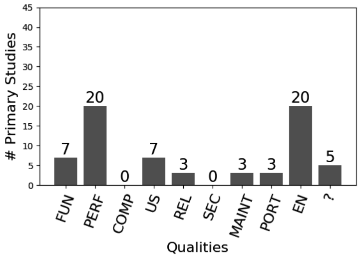
6.4.1 Goals of Self-Adaptation (RQ1)

Figure 6.3 shows the distributions of the goals characteristics across the primary studies. As shown in Figure 6.3a, the most common **quality** attributes are *performance efficiency* and *energy.* This is an expected result since both performance and energy are fundamental aspects of the user experience in mobile applications, potentially impacting the app user ratings and reviews which, unless properly addressed, can negatively contribute to the app's success [266; 174]. Furthermore, it is interesting to note that self-adaptation for either *compatibility* or *security* is never mentioned in our primary studies, thus unveiling two potentially fruitful research gaps to be filled by researchers in the future.

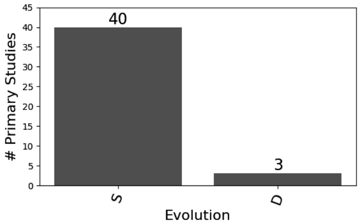
For what concerns **evolution** (see Figure 6.3b), we observe that the vast majority of primary studies presents a system with *static*ally-defined goals, whereas only 3 primary studies present an approach in which goals can evolve during the execution of the system. For example, in MAsCOT [255] developers can configure at any time the self-adaptation objectives and trade-offs (*e.g.,* acceptable latency vs available CPU power) via an XML-based dynamic decision network, which is then used at run-time by the system for deciding whether computation should be executed on the mobile device or in the cloud.

The **flexibility** dimension is quite fragmented (see Figure 6.3c), where we can see that primary studies are pretty evenly distributed among *rigid*, *constrained*, and *unconstrained* goals, with a slight tendency towards *unconstrained* goals.

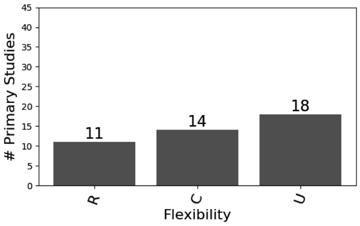
If we examine the **duration** attribute in Figure 6.3d, we observe that most of the primary studies support goals with a *persistent* validity, as opposed to only 5 studies supporting *temporary* goals. The approach presented in [84] is an



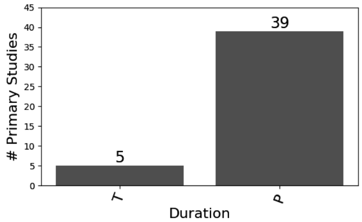
(a) *Quality requirements*



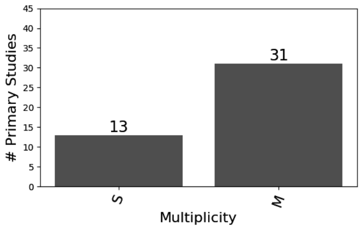
(b) *Evolution*



(c) *Flexibility*



(d) *Duration*



(e) *Multiplicity*

Figure 6.3: *Characteristics of the goals of self-adaptation*

example of study dealing with temporary goals; the purpose of the approach is to allow developers to develop mobile applications in a declarative manner; then it will be the responsibility of the system to adapt the application to the device on which it is deployed. Since the adaptation goal is limited to the deployment phase, we can consider it as temporary.

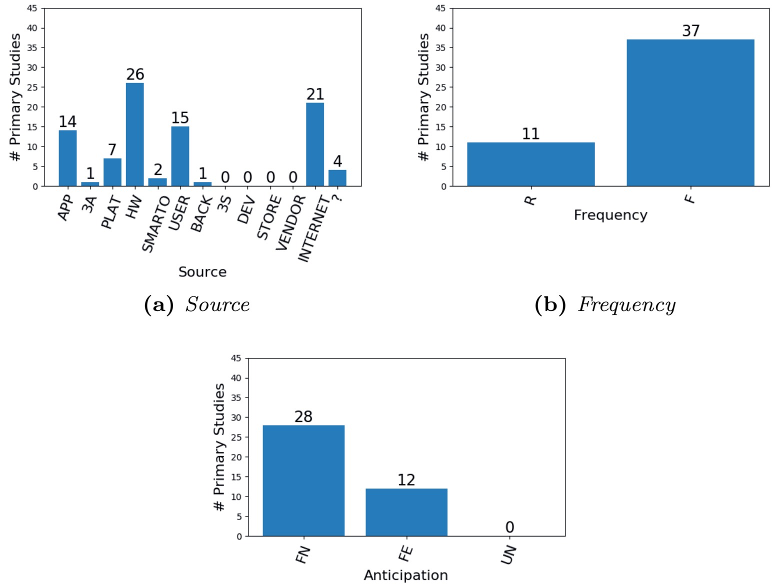
Lastly, if we focus on goals **multiplicity**, in figure 6.3e we can observe that the majority of primary studies have *multiple* goals. This result has to do with most systems employing self-adaptation not only to optimize the system in terms of a single dimension (*e.g.,* to reduce battery consumption), but they focus on the trade-off among different dimensions and types of resources, such as Internet connectivity, CPU usage, user experience, etc.

6.4.2 Changes Triggering Self-Adaptation (RQ2)

In Figure 6.4a we can notice that the most common **source** of changes is the *hardware* of the device, with close second being *Internet* connectivity. This result can be due to the vary nature of mobile applications being deployed on smartphones; as such, a common concern for developers is to optimize the utilization of the hardware the application is installed on and to provide the best service at all times (*e.g.,* to do not consume too much battery at runtime or to react to sensors faults). Furthermore, we can explain the fact that Internet is the second most common source of changes in the primary studies because of a significant number of studies dealing with cloud offloading. In those cases, the system decides to offload computation to the cloud depending on the available bandwidth (among other parameters) and adapts its behaviour accordingly. As a last remark on this attribute, we can observe that none of the primary studies consider as a source of change the following entities of mobile-enabled systems: third-party services, developers, app store, and platform vendor. Among them, it comes as a surprise that no primary study considers third-party services as a source of changes. Indeed, it is quite common that third-party services change their provided APIs (*e.g.,* Facebook changing the signature of its GraphAPI[3] endpoint for sharing a link) and it may be interesting to investigate on how self-adaptation techniques can help in automatically keeping the calling apps as reliable as possible, despite those (potentially unforeseen) changes.

The second examined attribute is **frequency**, where we observe that the

[3] https://developers.facebook.com/docs/graph-api



(a) *Source*

(b) *Frequency*

(c) *Anticipation*

Figure 6.4: *Characteristics of the changes triggering the self-adaptation process*

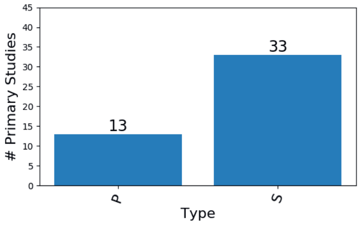
most common type of changes are *frequent* (see Figure 6.4a). This result was expected, especially after having observed that the two most common sources of changes are hardware and Internet availability and that by their own nature their status can drastically change in a matter of nanoseconds.

For what concerns the **anticipation** of changes, as shown in Figure 6.4c, changes are mostly *foreseen*, followed by fewer occurrences of approaches supporting *foreseeable* changes. It is important to note that in self-adaptive systems *foreseen* changes are known at design time and considered as expected to occur during the normal operation of the system [192]; examples of foreseeable changes we encountered in the primary studies include: a drop of available bandwidth [120], the user reaching its home address [240], the mobile device getting in proximity of a smart object [80], the user starting to drive a car [306]. Differently, changes are *foreseeable* when they are not known at design time, but they can be resolved at runtime and there is a plan for managing them during the execution of the system [192]. Examples of foreseeable changes include: the backend of the app has a failure [59], the GPS sensors of the mobile device produces incorrect data [84], etc. Finally, it is interesting to note that no primary study considers *unforeseen* changes, *i.e.,* drastic changes that have been not planned for and that are unknown until their first occurrence [192]. Unforeseen changes are extremely challenging to be managed due to their intrinsic level of uncertainty, both at design time and runtime. We speculate that investigating on how to incorporate them into self-adaptive mobile-enabled systems will be a scientifically challenging research area of the future.

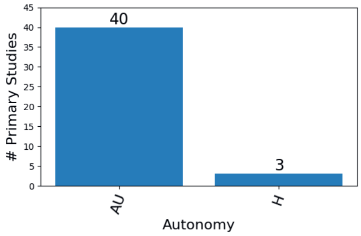
6.4.3 RQ3 – Mechanisms for Self-Adaptation (RQ3)

The **type** of self-adaptation supported in more than half of the primary studies is *structural* (see Figure 6.5a), *i.e.,* the adaptation involves structural changes in software architecture of the system [9]. This result can be explained by the high number of primary studies focussing on communication middleware, generic frameworks and meta-approaches which allow the self-adaptation process to reconfigure the architecture of the system at run-time and are, therefore, structural in nature. In 13 primary studies the self-adaptation is *parametric*, *i.e.,* the adaptation process involves only the policy files and configuration of specific components of the system, without changing its overall organization [9]. Examples of primary studies supporting structural self-adaptation include approaches for autonomously adapting the bitrate of the video streaming to the app [244], approaches for sending personalized notifications to the user which automatically adapt to his/her stress levels [210], etc.

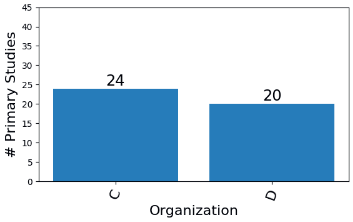
For what concerns **autonomy**, we can observe that nearly all approaches are *autonomous* (see Figure 6.5b). This result is most likely influenced by the definition of self-adaptation that we have used throughout this study, in which the system has to be able to self-adapt and therefore can only have minimal



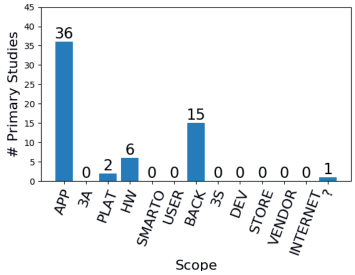
(a) *Type*



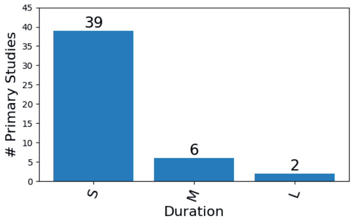
(b) *Autonomy*



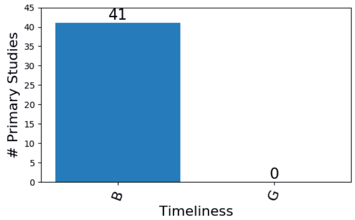
(c) *Organization*



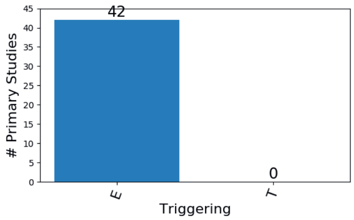
(d) *Scope*



(e) *Duration*



(f) *Timeliness*



(g) *Triggering*

Figure 6.5: *Characteristics of the mechanisms for self-adaptation*

human assistance in the process. Nonetheless, we have found 3 primary studies presenting *human assisted* approaches. For example, the approach presented in [290] aims at improving the user experience of the app with the use of an adaptive user interface. This approach is human aided because it needs the user to participate in a brief test in order to determine where the elements of the user interface should be positioned and their dimensions in the graphical layout of the app. This is obviously in contrast with a fully automated system that would monitor user actions in background and adapt accordingly, without requiring the initial test performed explicitly by the user.

The **organization** attribute is almost evenly split among our primary studies (see Figure 6.5c). This implies that we have nearly just as many primary studies where the adaptation is *centralized* into a single component as systems where adaptation is *distributed* among several components. An example of centralized self-adaptation is [372], where there is a single software component which is in charge of adapting the user interface of the app to the preferences, knowledge, and skills of the user. An example of distributed self-adaptation is presented in [120], where both the app and its backend are involved in the code offloading process.

Considering our mobile-enabled system in Figure 6.1 and examining the **scope** of self-adaptation, we notice that the vast majority of the self-adaptation mechanisms are executed in the application itself (see Figure 6.5d). Quite a few approaches also have mechanisms executing in the back-end of the app, where a significant part is due to the fact that the primary studies deal with cloud offloading (such as the previously given example). Furthermore, we have observed that some studies have self-adaptation mechanisms executing in the hardware and the platform of the mobile device. This result is a confirmation that in a mobile-enabled system the intelligent entities are either the app, its backend, or the software stack on which the app is running on the client side.

Regarding the **duration** of the mechanism (see Figure 6.5e), we notice that the vast majority of the mechanisms have a *short* duration, followed by 6 approaches having a *medium* duration, and only 2 approaches having a long duration. As an example of a long duration mechanism, the authors of [133] monitor user actions and apply a set of algorithms to perform self-adaptation for adapting the items in the graphical menus of the app according to the usage patterns exhibited by the user. In this way, the app is enhanced with a transformable and movable menu component with adaptable and adaptive features, which improves the overall efficiency of the user when using the app. Due to the fact that the app is constantly learning the usage patterns of the user, the mechanism presented in [133] can be considered as having a long adaptation duration.

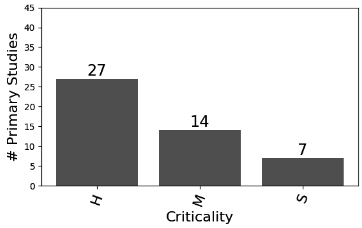
As shown in Figure 6.5f, the **timeliness** attribute falls fully in the *best-effort* bin (*i.e.,* the time for executing the adaptation is not guaranteed) and there are no primary studies proposing a *guaranteed* time for the self-adaptation process. This result is extremely interesting since in some application domains (*e.g.,* emergency-related, post-disaster apps) it may be a strict requirement to know and respect upper bounds on the execution time of the self-adaptation process, thus guaranteeing the timeliness associated with self-adaptation.

For what concerns self-adaptation **triggering** (see figure 6.5g), all the primary studies are based on *event triggered* mechanisms. This result is quite expected, given that mobile apps are mostly front-end software reacting either to user- or system-generated events.

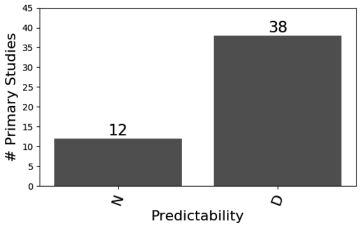
6.4.4 RQ4 – Effects of Self-Adaptation (RQ4)

Starting with **criticality** (see Figure 6.6a), we observe that the majority of primary studies describe a self-adaptation process that has a *harmless* impact on the system in the case such adaptation were to fail. Fourteen and seven studies can have *mission-critical* and *safety-critical* consequences, respectively. Furthermore, we can also observe that safety-critical consequences are highly dependent on the domain in which the approach is being applied. This is logical, as, for example, a video-streaming adaptation process would most likely not run the risk of hurting its users in case of a failed adaptation.

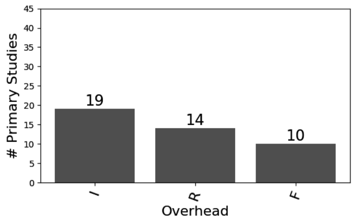
In terms of **predictability**, the majority of the self-adaptation approaches are *deterministic* (see Figure 6.6b), followed by twelve approaches with non-



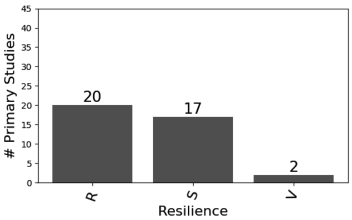
(a) *Criticality*



(b) *Predictability*



(c) *Overhead*



(d) *Resilience*

Figure 6.6: *Characteristics of the effects of the self-adaptation process*

deterministic effects. This means that in the majority of the cases the users of self-adaptive approaches know the possible states of the mobile app (and of the overall system) after adaptation.

Examining the **overhead** attribute (see Figure 6.6c), we observe a nearly even distribution among all possible values, ranging from *insignificant*, to *reasonable*, and finally to system *failure* overhead. The relatively high number of primary studies whose approaches may lead to system *failure*s is mainly due to the relatively high number of approaches cloud offloading (or other network-related mechanisms), in which the apps' functionalities have to stop being provided whilst the work is being offloaded and in some cases until its results have been obtained from the cloud.

Finally, under **resilience** there is a nearly even number of *resilient* and *semi-resilient* approaches, with hardly any being *vulnerable* (see Figure 6.6d). Resilience is defined as the persistence of service delivery that can justifiably be trusted, when facing changes [192]. As examples of approaches falling under the *vulnerable* category, four cases of faults and failures of context-aware adaptive applications are presented in [306]. One of those cases is related to an app supporting a so-called "meeting profile", which was autonomously applied whenever the app infers that the user is in a meeting with a colleague (based on the device's calendar, the current time, and on Bluetooth discovering another person in the room); however, the approach was falling into an adaptation cycle between the office and the meeting profiles since both of their conditions were triggered whenever a meeting was held in the office, leading to inconsistencies in the behaviour of the app.

6.5 Discussion

In this section we present the research implications that we derived from our results (Section 6.5.1), followed by an overview at the application domains we encountered in our primary studies (Section 6.5.2) and the main challenges reported in the examined literature (Section 6.5.3).

6.5.1 Research Implications

We conducted this systematic literature study to gain insight on self-adaptation in the context of mobile apps. To analyse our data we formulated four research questions (see section 6.3.1). Here we report on the main findings related to our answers.

In our findings related to the **goals** of self-adaptation (*RQ1*) we observed that the vast majority of approaches have persistent and static goals. This implies that *the majority of the identified self-adaptive approaches are relatively rigid in their objectives*, unveiling a certain research gap involving approaches where goals can change at runtime depending on the ever-evolving context in which mobile apps are used today. Furthermore, the most frequently pursued goals are related to technical quality properties of mobile-enabled systems, such as performance and energy consumption. Interestingly, *there are few approaches targeting non-technical goals*, such as promoting user behavioural change and lifestyle improvement. Researchers in the field can direct their future studies towards filling this identified gap in the state of the art of self-adaptive mobile-enabled systems. For developers working on self-adaptive apps with static goals, the set of primary studies could represent a valuable source of knowledge.

In analysing the **changes** that trigger self-adaptation (*RQ2*), we observe that the majority of approaches adapt due to changes within the hardware of the mobile device or its Internet connectivity. This result unveils a very interesting research gap related to potential *self-adaptive approaches which can adapt their behaviour or structural configuration to changes occurring in third-party applications running on the mobile device, in third-party services running in the cloud, or in smart objects surrounding the user of the app.* The above gap points to a potential unexplored market: developers looking for innovative self-adaptive apps could consider developing apps that adapt due to their third-party services, *e.g.,* to provide a better user experience, as opposed to competing apps that use these services without self-adaptation.

When discussing the **adaptation mechanisms** (*RQ3*), we observe some interesting findings as well. Firstly, the majority of approaches perform the adaptation in an autonomous manner, and therefore do not need a human in the loop or any other forms of human assistance to accomplish their adaptation. Moreover, in most cases the adaptation is performed by the application itself, with only sometimes requiring the help of the backend (such as in the case of cloud offloading applications). Secondly, all of the analysed adaptations are event-triggered and perform in a best-effort manner; therefore they do not have a guarantee on the duration of the self-adaptation process. This can be justified by the overwhelming short and medium duration of the adaptation processes we have studied, which by their nature are nearly impossible to guarantee in their timeliness. Nevertheless, an interesting research direction is about self-adaptive approaches where *formal and rigorous reasoning* plays a central role into making self-adaptation feasible also for apps belonging to critical domains (*e.g.,* energy, defence, transportation). Some works into this direction are starting to emerge in other fields, such as for the Internet of Things [374; 251], but *the application of formal reasoning in the context of self-adaptive mobile applications still seems to be an under-explored research area.*

Lastly, we need to discuss the **effects** of the self-adaptation process (answering *RQ4*). However, before doing so we must note that the reported information on effects was challenging to collect as it was rarely explicitly stated within the primary studies and, most of the time, had to be deduced by the description of the self-adaptation process and the analysed software system. *We advice researchers working on self-adaptive software systems to pay special attention to the effects dimension of self-adaptation*, so to provide a clear and complete overview of their proposed solutions. Having disclosed this, we nonetheless have a few noticeable results with the most prominent finding being that most of the analysed approaches had predictable consequences to their self-adaptation approach. Furthermore, the majority of analysed adaptation mechanisms had a harmless effect on the app in case the adaptation failed. However, we also found cases in which this effect was mission-critical or even safety-critical, and have noticed that there seems to be a strong link between a system being mission- or safety-critical and its application domain. This seems reasonable as *e.g.,* failed adaptation in the health domain is more likely to be safety-critical as opposed to a video streaming application, which poses no threat to the safety of the user in case of failed adaptation. This emerging link between domain and level of criticality, should be considered carefully by any developer working on self-adapting apps in mission/safety-critical domains.

6.5.2 Application Domains

When analysing the primary studies we also traced the application domains in which the self-adaptive mobile apps have been applied. Of the cited application domains, *health* is the most popular with 7 primary studies. Other application domains in which self-adaptive mobile apps have been applied include: tourism, e-learning, mapping, education, science, conferencing, e-commerce, social networks, smart city, art, video streaming, image manipulation, and emergency management. Overall, such a high number of application domains hints to the general applicability of self-adaptive mechanisms, provided that the context and specifics of the application domain are taken into consideration (*e.g.,* apps should respect the intrinsic privacy-related concerns of domains like health and e-learning).

On the other hand, 22 primary studies do not mention any particular application domain. This indicates that a substantial amount of research has focused on self-adaptation mechanisms regardless of their application domain. This could be due to such mechanisms being broad enough to be applied in general. In the future, it might be interesting for researchers to investigate if there exist categories of self-adaptation techniques that are application-specific and others that are general-purpose.

6.5.3 Emerging Challenges

By extracting meaningful paragraphs from 'future work' and 'challenges' sections of our primary studies and then analysing them, we have managed to find some common points of interest both for researchers and practitioners. Specifically:

- 12 primary studies mention the need or to further improve the implementation of their approach in order to reduce bias or eliminate potentially wrongful assumptions and of these studies 4 specifically mention only having implemented a research prototype;

- 9 of our primary studies mention the need of performing a more robust evaluation of their proposed approach;

- 4 primary studies mention the need to test their proposed approach on case studies as it was only tried in simulation or just theoretically;

- 3 primary studies mention the need for performing an in-depth comparison between their approach and the ones proposed by other researchers.

The information we have extracted seems to be reinforcing our previously given observation in subsection 6.5.2. It would seem as a significant number of primary studies are working on a more theoretical level, therefore in a state of still needing further improvement and testing on practical scenarios and real-world applications. From this, we would therefore suggest that future research effort should be devoted not only to the improvement of the existing theoretical underpinnings of self-adaptation for mobile apps, but also to its application in real-world, realistic scenarios, at best by applying empirical case studies in industrial settings [381].

6.6 Threats To Validity

The following reports on the potential threats to validity of this study according to [381].

Internal Validity. We mitigated internal threats to validity by using already established modelling dimensions [9] as our classification framework. For the validity of the synthesis of the collected data, we utilised well assessed descriptive statistics in order to minimise potential threats.

External Validity. In our study the main external threat to validity may come from our primary studies not being representative of the whole research on self-adaptation in the context of mobile applications. In order to mitigate such risk, we employed a search strategy including of both automatic search as well as backward-forward snowballing of the selected primary studies found with the automatic search. Furthermore, we chose to consider only peer-reviewed papers and excluded any work that could be defined as grey literature. We do not foresee this criterion to have impacted our study as the considered papers need to have undergone a rigorous peer-review process, which is an established requirement for quality publications. Lastly, we applied well defined inclusion and exclusion criteria, which we have rigorously followed during our manual selection phase.

Construct Validity. To be sure that the found primary studies would be able to competently answer the chosen research questions we manually carried out the selection process using the chosen inclusion and exclusion criteria, reported in subsection 6.3.2. Such results were then further expanded by also conducting forward and backwards snowballing on those same selected studies.

Conclusion Validity. In order to reduce potential bias our classification framework is based on established modelling dimensions found in [9]. This way we can confidently guarantee that the data extraction process was aligned with our chosen research questions. Furthermore we reduced potential threats to conclusion by following well-known systematic literature review guidelines [179; 276; 381].

6.7 Related Work

Works related to ours are secondary studies on self-adaptation.

Yang *et al.* [387] focused on requirements modelling and analysis for self-adaptive systems. They carried out a systematic literature review of 101 primary studies, from which they elicited 16 modelling methods and 10 requirement quality attributes. They observed that some of the modelling methods need further study, and most qualitative studies need better evaluation.

Krupitzer *et al.* [185] survey the engineering approaches found for self-adaptive systems. To this aim, they use a taxonomy for self-adaptation extended with the "context perspective", *i.e.,* the ability of systems to adapt their context. The survey identifies and classifies several approaches used to build self-adaptive systems.

Macias-Escriva *et al.* [215] analyse self-adaptability from the perspective of computer science and cybernetics, and examine the approaches found in the literature, to gain an overview of the state-of-the-art techniques used for self-adaptation. As one of the main conclusions, they identify feedback control and artificial intelligence as enabling fields to help further develop self-adaptive systems.

Both Krupitzer *et al.* and Macias-Escriva *et al.* do not report on the number of studies used in the data extraction for the surveys.

Mahdavi-Hezavehi *et al.* [219] conducted a systematic literature review of 54 primary studies, with the goal of understanding 'the state-of-the-art of architecture-based methods for handling multiple quality attributes (QAs) in self-adaptive systems'. They found that the most frequently addressed QAs are performance and cost, and the most common domains are robotics and web-based system.

Muccini *et al.* [250] focused on self-adaptation in the context of cyber-physical systems, and analysed 42 primary studies. As part of their main results the authors found MAPE (Monitor-Analyze-Plan-Execute) as the most common mechanism used to perform adaptation in this context, and energy as that the most common application domain.

Lastly, Weyns *et al.* [375] examine the claims that are associated with self-adaptation. They analysed 96 primary studies identified from the SEAMS conference series between 2006 and 2011, and the papers published in 2008 in [71]. They observe that (i) the main focus is on architecture and models, (ii) the most common application domain is service-based systems, and (iii) at the time of publishing only a few empirical studies were performed with no industrial evidence.

In spite of the relatively large number of secondary studies on self-adaptation and self-adaptive systems, none explored the state of the art of self-adaptation in the context of mobile applications. Our study certainly fills this gap, which with the increasing pervasiveness of mobile software in all application domains is turning into a necessity.

6.8 Conclusions and Future Work

This paper presents a systematic literature review on self-adaptation in the context of mobile applications as defined in section 6.2. Starting from 607 possibly relevant studies, we found 44 primary studies which we analysed via the presented classification framework, in order to answer our chosen research questions. By answering these questions, we give an in-depth look at the field of self-adaptation in the mobile application context, and therefore provide valuable information for researchers and developers who wish to work in the future within this area.

As future work, we will perform a longitudinal analysis across the various dimensions of our classification framework as it would help discover more complex (and hidden) patterns among the analysed approaches. Furthermore, a more in-depth analysis of the contents of the primary studies could contribute in better understanding the current research gaps about self-adaptation in the context of mobile applications. Finally, in our next chapter we present our work on the use of cluster-based reinforcement learning and self-adaptation to automatically adapt and personalise e-Health mobile apps. Allowing these apps to better support users in following medical advice and improving their wellbeing.

Part III

Creating self-adaptive and personalised e-Health mobile applications

7

A Reference Architecture for e-Health mobile applications

Part of chapter 7 was published as:

Grua, E. M., De Sanctis, M., & Lago, P. (2020, September). A Reference Architecture for Personalized and Self-adaptive e-Health Apps. In European Conference on Software Architecture (pp. 195-209). Springer, Cham.

Furthermore, this chapter will appear as a book chapter:
Grua, E. M., De Sanctis, M., Malavolta, I., Hoogendoorn, M., & Lago, P. (2021). Social Sustainability in the e-Health Domain via Personalized and Self-adaptive Mobile Apps. Software Sustainability. Springer, Cham. To appear (book chapter).

Abstract - Within software engineering, social sustainability is the dimension of sustainability that focuses on the 'support of current and future generations to have the same or greater access to social resources by pursuing social equity'. An important domain that strives to achieve social sustainability is *e-Health*, and more recently e-Health mobile apps.
A wealth of e-Health mobile apps are available for many purposes, such as life style improvement, mental coaching, etc. The interventions, prompts, and encouragements of e-Health apps sometimes take context into account (e.g., previous interactions or geographical location of the user), but they still tend to be rigid, e.g., apps use fixed sets of rules or they are not sufficiently tailored towards individuals' needs.
Personalisation to the different users' characteristics and run-time adaptation to their changing needs and context provide a great opportunity for getting users continuously engaged and active, eventually leading to better physical and mental conditions.
The overall goal of this chapter and it's contents is to answer ***T.RQ*4**, namely: *How can AI-based personalisation and self-adaptation be used to create e-Health apps that dynamically adapt to the user and their context?* To this goal, we present *a reference architecture for enabling AI-based personalisation and self-adaptation of mobile apps for e-Health.* The reference architecture makes use of a dedicated goal model and multiple MAPE loops operating at different levels of granularity and for different purposes.
The proposed reference architecture is instantiated in the context of a fitness-based mobile application and exemplified through a series of typical usage scenarios extracted from our industrial collaborations.

7.1 Introduction

E-Health mobile apps are designed for assisting end users in tracking and improving their own health-related activities [378]. With a projected market growth to US$102.3 Billion by 2023, e-Health apps represent a significant market [134] providing a wide spectrum of services, i.e., life style improvement, mental coaching, sport tracking, recording of medical data [269]. The unique characteristics of e-Health apps w.r.t. other health-related software systems are that e-Health apps (i) can take advantage of smartphone sensors, (ii) can reach an extremely wide audience with low infrastructural investments, and (iii) can leverage the intrinsic characteristics of the mobile medium (i.e., being always-on, personal, and always-carried by the user) for providing timely and in-context services [119].

However, even if the interventions, prompts, and encouragements of current e-Health apps take context into account (e.g., previous interactions or geographical location of the user), they still tend to be *rigid* and not fully tailored to individual users, e.g., by using fixed rule sets or by not considering the unique traits and behavioural characteristics of the user. In this context, we see *personalisation* [109] and *self-adaptation* [147; 373; 387] as effective instruments for getting users continuously engaged and active, eventually leading to better physical and mental conditions. The addition of intervention tailoring (via personalisation and self-adaptation) is a crucial step in addressing the main sustainability concern that e-Health mobile apps want to achieve: *social sustainability*. By providing better interventions, we are not only more likely to have the user interested in maintaining engagement with the app but also help the user achieve better physical and mental conditions; allowing the app to better address the personal needs and by extension the social ones too.

In this work, we combine personalisation and software self-adaptation to provide users of mobile e-Health apps with a better, more engaging and effective experience. To this aim, we propose a *reference architecture that combines data-driven personalisation with self-adaptation*. The main design drivers that make the proposed reference architecture unique are:

– the **combination of multiple *Monitor - Analyze - Plan - Execute* (MAPE) loops** [163] operating at different levels of granularity and for different purposes, e.g., to suggest users the most suitable and timely activities according to their (evolving) health-related characteristics (e.g., active vs. less active), but also to cope with technical aspects (e.g., connectivity hiccups, availability of IoT devices and third-party apps on the user's device) and the characteristics of the physical environment (e.g., indoor vs. outdoor, weather);

- a **dedicated goal model** for representing health-related goals via a descriptive concise language accessible by healthcare professionals (e.g., fitness coaches, psychologists);

- the exploitation of our **online clustering algorithm** for efficiently managing the evolution of the behaviour of users as multiple time series evolving over time. This online clustering algorithm has been already extensively tested in a previously published article [146], showing promising results by doing better than the current state-of-the-art.

The main characteristics of the proposed reference architecture are the following: (i) it caters the personalisation of provided services to the specific user preferences (e.g., preferred sport activities); (ii) it guarantees the correct functioning of the provided features via the use of connected IoT devices (e.g., a smart-bracelet) and runtime adaptation strategies; (iii) it adapts the provided services depending on contextual factors such as environmental conditions and weather; (iv) it supports a smooth participation of domain experts (e.g., psychologists) in the personalisation and self-adaptation processes; and (v) it can be applied in the context of a single e-Health app and by integrating the services of third-party e-Health apps (e.g., already installed sport trackers). All of the above mentioned characteristics are shown in this work by evaluating the reference architecture and the goal model with fitness coaching scenarios. We want to emphasise how most characteristics have been engineered with the main goal of achieving social sustainability. A possible exception are characteristics (ii) and (v) which more specifically addresses technical sustainability of the reference architecture. Our emphasis on social sustainability will be further explained and explored throughout the paper.

Lastly, in a previous study [143] we reported a preliminary version of our Reference Architecture. Here we extend the work by: (i) framing the work in the overall context of social sustainability, (ii) document the methodology used to design our Reference Architecture, (iii) report a scenario-based evaluation of our Reference Architecture, (iv) provide a goal model to be used with the Reference Architecture, (v) a viewpoint definition used to create the view of our Reference Architecture.

7.2 Background

The notion of *reference architecture* (RA) is borrowed from Volpato *et al.* [362], who define it as “a special type of software architectures that provide a characterisation of software systems functionalities in specific application domains”, e.g., SOA for service orientation and AUTOSAR for automotive. In the context of this study, a *self-adaptive software system* is defined as a system that can autonomously handle changes and uncertainties in its environment, the system itself and its goals [373].

For the definition of *personalisation* we build on that by Fan and Poole [109] and define it as “a process that changes a system to increase its personal relevance to an individual or a category of individuals”. Furthermore, to enhance personalisation, we use CluStream-GT (standing for: CluStream for Growing Time-series). CluStream-GT was chosen for this RA as it is the state-of-the-art clustering algorithm for time-series data (especially within the Health domain). CluStream-GT works in two phases: offline and online. First, the *offline phase* initialises the algorithm with a small initial dataset; this is done either at design time or at the start of runtime. After, during the *online phase* the algorithm clusters the data that is being collected at runtime. Clustering allows the RA to group similar users together; where similarity is determined by the data gathered from the apps. This gives the RA a more sustainable and scalable method of personalisation, without requiring to create individual personalisation strategies but maintaining a suitable degree of personalisation [146; 177]. For a more in depth explanation of the algorithm and how clustering can aid personalisation we refer the reader back to Chapters 3 and 4.

The *methodology* used for the design of our RA is the one presented by Angelov *et al.* [14] (see Fig. 7.1), where the authors present their RA Framework to facilitate software architects in the design of *congruent RAs*, i.e., RAs where the design, context and goals are explicit and coherent (adapted from [14]).

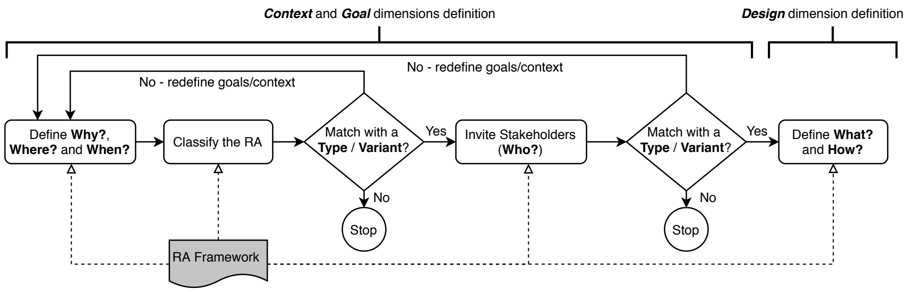


Figure 7.1: *Methodology for the design of our RA[14]*

The RA Framework (or framework for short) consists of two elements: a multi-dimensional classification space, and a set of predefined RA types (and variants of these types). The former, through the use of strict questions and answers, supports software architects in classifying RAs according to their *context* (**Where?**, **Who?** and **When?** questions in Fig. 7.1), *goals* (**Why?** in Fig. 7.1), and *design* (**How?** and **What?** in Fig. 7.1) dimensions. The latter consists of specific combinations of values from the multi-dimensional space. These types, and variants of, are used to evaluate the congruence of the RA being designed. If a RA is congruent (i.e. matches a type or variant) it has a greater chance of becoming a success, where by success the authors mean *"... the acceptance of the architecture by its stakeholders and its usage in multiple projects"*[14]. For each dimension, the authors have defined sub-dimensions with respective questions and answers. During the design of our RA we have worked with each dimension and, with the use of the framework, classified our RA according to the possible values available for each sub-dimension. As knowledge of our RA and its components is necessary to understand the design process, we further explain the use of the framework in Sec. 7.7.

In recent years a larger body of software engineering and software architecture works address sustainability. Sustainability can be divided into four dimensions: technical, economical, environmental, and social [188]. Within this work we present an RA for the e-Health domain with the main goal of better addressing the social dimension of sustainability, whilst the technical contributions of this work include the combination of AI and self-adaptation. In this work we build on the following definition of social sustainability: "focusing on supporting current and future generations to have the same or greater access to social resources by pursuing generational equity. For software-intensive systems, this dimension encompasses the direct support of social communities, as well as the support of activities or processes that indirectly create benefits for such communities"[188].

7.3 Related Work

Several RAs for IoT can be found in the literature [30; 28; 2; 122]. In particular, Bauer *et al.* [30] present several abstract *architectural views* and *perspectives*, which can be differently instantiated. The adaptation of the system's configuration is also envisioned, at an abstract level. IoT-A [28] aims to be easily customised to different needs, and it makes use of *axioms* and *relationships* to define connections among IoT entities. IIRA [2] is particularly tailored for industrial IoT systems. WSO2 [122] presents a layered structure and targets scalability and security aspects too. All of the above RAs are abstract and domain independent. As such, they do not address required features specific to the IoT-based e-Health domain. Moreover, they lack the needed integration with AI for personalisation used to tailor interventions to the user's health-related characteristics; an important technique used by the RA to address social sustainability.

Other works providing service oriented architectures (SOAs) focused on adaptation but neglected user-based personalisation. E.g., Feljan *et al.* [111] defined a SOA for planning and execution (SOA-PE) in Cyber Physical Systems (CPS), and Mohalik *et al.* [245] proposed a MAPE-K autonomic computing framework to manage adaptivity in service-based CPS. Morais *et al.* [92] present RAH, a RA for IoT-based e-Health apps. RAH has a layered structure, and it provides components for the prevention, monitoring and detection of faults. Differently from RAH, our RA explicitly manages the self-adaptation of the e-Health mobile app, both at users- and architectural levels. Mizouni *et al.* [240] propose a framework for designing and developing context-aware adaptive mobile apps. Their framework lacks other types of adaptation, i.e., adaptation for user personalisation and adaptation with other IoT devices – which is possible with our RA.

Lopez and Condori-Fernandez [210] propose an architectural design for an adaptive persuasive mobile app with the goal of improving medication adherence. Accordingly, the adaptation is here focused only on the messages given to the user and lacks the other levels of adaptation (environment adaptation, etc.) that our RA covers. Kim [176] proposes a general RA that can be used when developing adaptive apps and implements a e-Health app as an example. However, being it general, the RA lacks the level of detail present in our work, the integration of AI for personalisation, and a way for involving domain experts in the app design and operation, which is essential in adaptive e-Health.

In summary, to the best of our knowledge, ours is the first RA for e-Health mobile apps that simultaneously supports (i) *personalisation* for the different users, by exploiting the users' smart objects and preferences to dynamically get data about e.g., their mood and daily activities, and (ii) *runtime adaptation* to the user-needs and context in order to keep them engaged and active, so that we can better address social sustainability.

7.4 Reference Architecture

Fig. 7.2 shows our RA with the following stakeholders and components. Section 7.8 defines the corresponding viewpoint.

Users provide and generate the Data gathered by the e-Health app. At the first installation, the users are asked to input information to better understand their aptitudes. After an initial usage phase and data collection, the system has enough information to assign them to a cluster.

Smartphone is the host where the self-adaptive e-Health app is installed. In the mobile app, four components, namely *User Driven Adaptation Manager*, *Environment Driven Adaptation Manager*, *Smart Objects Manager* and *Inter-*

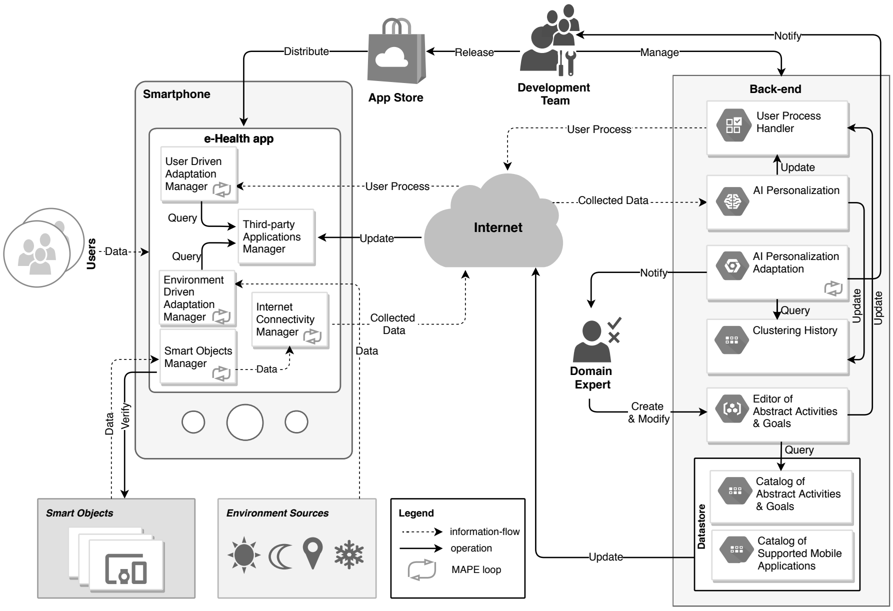


Figure 7.2: *Reference architecture for Personalised and Self-adaptive e-Health Apps*

net Connectivity Manager implement a MAPE loop to dynamically perform adaptation. The *Third-party Applications Manager*, in turn, is responsible for the communication with third-party apps supported by the RA that can be exploited by the e-Health app both during its nominal execution and when adaptation is performed. It is also responsible for storing the user's preferences. Further details on these components are given in Sec. 7.5.

Smart Objects are devices, other than the smartphone, that the app can communicate with. They are used to gather additional data about the users as well as augmenting the data collected by the smartphone sensors. For instance, a smart-watch would be used by the app to track the user's heart-rate, therefore adding extra information on the real-time performance of the user.

Environment is the physical location of the user, and its measurable properties. It is used by the e-Health app to make runtime adaptations according to its current operational context and to the user's scheduled activities, as described in Sec. 7.5.5.

The back-end of our RA (right-hand side of Fig. 7.2) is Managed by a *Development team*. It additionally exposes an interface to the *Domain Expert* that is also involved in the e-Health app design and operation. The back-end contains the components needed to store the collected user data and to manage the user clusters. It also hosts components supporting the general functioning of the app.

User Process Handler is in charge of sending User Processes to the users, by taking care of sending the same User Process to all users of the same cluster. A *User Process* is composed of one or more *Abstract Activities*. These activities are inspired by the ones introduced in [47], although they differ both in the structure and in the way they are refined, as later explained. An *Abstract Activity* is defined by a vector of one or more *Activity categories* and an associated goal, with each vector entry representing a day of the week. Examples of Abstract Activities are discussed later in section 7.9.

Each *Abstract Activity* is defined by the Domain Expert via the *Editor of Abstract Activities & Goals* and later stored in the *Catalog of Abstract Activities & Goals*. Each Activity category identifies the kind of activity the user should perform. As an example, the user can receive either a *Cardio* or *Strength* Activity category and so should perform an activity of that kind. More precisely, for each user, the Activity categories are converted to *Concrete Activities* at run-time via the use of the *User Driven Adaptation Manager* and based on the user's preferences. For instance, a cardio Activity category can be instantiated into different *Concrete Activities* such as running, swimming and walking. Moreover, if an *Abstract Activity* is composed of multiple Activity categories, all or some of type Cardio, they can be converted into different *Concrete Activities*. This implies that users who receive the same User Process will still be likely to have different *Concrete Activities*, therefore personalising the experience to the individual user (this is further discussed in Sec. 7.5.2).

The goals associated with an *Abstract Activity* are also important for distinguishing between Abstract Activities, besides for converting them into *Concrete Activities*. Two Abstract Activities containing the same vector of Activity categories can be different solely based on their associated goal. More details on the goal model are given in Sec. 7.6.

The *User Process Handler* receives Updates from (i) the *AI Personalization* and (ii) the *Editor of Abstract Activities & Goals* in order to send User Processes to their associated users. The AI Personalization Updates the *User Process Handler* every time a user moves from one cluster to another, while the *Editor of Abstract Activities & Goals* Updates it every time new clusters are analyzed by the Domain Expert (along with the new associated User Process). These updates guarantee that the *User Process Handler* remains up to date about the User Processes and their associated users.

AI Personalization sends an Update to the *Clustering History* component whenever a change occurs in the clusters. The *AI Personalization* component uses the CluStream-GT algorithm to cluster users into clusters in a real-time and online fashion [146]. It receives the input data from the e-Health app (see Collected Data in Fig. 7.2). More than one instance of CluStream-GT can be running at the same time. In fact, there is one instance per category of data. E.g., if the e-Health app is recording both ecological momentary assessment [318] and biometric data, one for the purpose of monitoring **mood** and the other for **fitness**, there will be two running instances of the algorithm.

AI Personalization Adaptation is in charge of monitoring the evolution of clusters and detecting if any change occurs. Examples include the merging of two clusters or the generation of a new one. To do so, it periodically Queries the *Clustering History* database. If one or more new clusters are detected, this component will Notify both the Development Team and the Domain Expert. The Domain Expert will examine the new information and add the appropriate User Process to the *Catalog of Abstract Activities & Goals* via the dedicated editor. In turn, the Development Team is notified just as a precaution so that it can verify if the new cluster is not an anomaly. The specifics of the corresponding MAPE loop are described in Sec. 7.5.1.

The role played by AI via the CluStream-GT algorithm is relevant in our RA as it strongly supports both personalisation and self-adaptation, thus guaranteeing a continuous user engagement that is crucial in e-Health apps. Specifically, personalisation is achieved by clustering the users based on their preferences and their physical and mental condition. This supports the RA in assigning appropriate *User Processes* to each user, and further adapt them to continuously cope with the current status of the user and by doing so better addressing social sustainability concerns.

Clustering History is a database of all the clusters created by the *AI Personalization* component. For each cluster it keeps all of the composing micro-clusters with all of their contained information.

Editor of Abstract Activities & Goals allows the Domain Expert to create and modify *Abstract Activities* (and their associated goals) and to combine them as User Processes. This is achieved via a web-based interactive UI and the editor's ability to Query the *Catalog of Abstract Activities & Goals*. It is also the editor's responsibility to update the *User Process Handler* if any new User Process has been created and is currently in use.

Catalog of Abstract Activities & Goals is a database of all User Processes that the Domain Expert has created for each unique current and past cluster. When a new cluster is defined, the Domain Expert can assign to it an existing User Process from this catalog, or create a new one and store it.

Catalog of Supported Mobile Applications is a database containing the metadata needed for interacting with supported third-party mobile apps installed on users' devices. This database stores information such as the specific types of Android intents (and their related extra data) needed for launching each third-party app, the data it produces after a tracking session, etc. Indeed, our e-Health app does not provide any specific functionality for executing the activities suggested to the user (e.g., running, swimming); rather, it brings up third-party apps (e.g., Strava[1] for running and cycling, Swim.com[2] for swimming) and collects the data produced by the apps after the user performs the physical activities. The main reasons for this design decision are: (i) we do not want to disrupt the users' habits and preferences in terms of apps used for tracking their activities, (ii) we want to *build on* existing large user bases, (iii) we do not want to reinvent the wheel by re-implementing functionalities already supported by development teams with years-long experience.

Whenever the e-Health app evolves by supporting new applications (or no longer supporting certain applications), the *Catalog of Supported Mobile Applications* Updates, through the *Datastore*, the *Third-party Applications Manager*. The *Third-party Applications Manager* responsibility is to keep the list of supported mobile apps up to date and provide the corresponding metadata to the *User Driven Adaptation Manager* and the *Environment Driven Adaptation Manager*, when needed.

The e-Health app and back-end communicate via the Internet. Specifically, the communication from the e-Health app to the back-end is REST-based and it is performed by the *Internet Connectivity Manager*, which is responsible for sending the Collected Data to the *AI Personalization* component in the back-end. Communication from the back-end to the e-Health app is performed by the *User Process Handler* which is in charge of sending the User Process to the e-Health app via push notifications.

[1] http://strava.com [2] http://swim.com

7.5 Components supporting Self-adaptation

The RA has five components used for self-adaptation. To accomplish its responsibilities, each of these components implements a MAPE loop.

7.5.1 AI Personalization Adaptation

The main goal of the AI Personalization Adaptation is to keep track of the clusters evolution and to enable the creation of new User Processes. It does it through its MAPE loop depicted in Fig. 7.3.

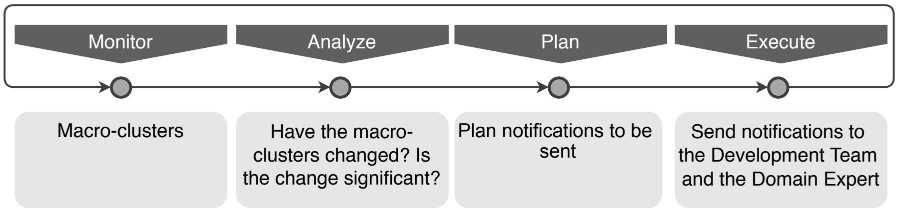


Figure 7.3: *AI Personalization Adaptation MAPE loop.*

During its Monitor phase, the AI Personalization Adaptation monitors the macro-clusters. In its Analyze phase it determines if there are changes in the monitored macro-clusters. To do so, the AI Personalisation Adaptation periodically queries the Clustering History database. It compares the current clusters with the previously saved ones. If any of the current ones are significantly different, then the AI Personalization Adaptation enters its Plan phase. The Plan phase gathers the IDs of the users and macro-clusters involved in these significant changes. Since this change involves the need of the creation of new User Processes for all of the users belonging to the new clusters the Domain Expert must be involved in this adaptation. To achieve this we have exploited the type of adaptation described in [131] which considers the involvement of humans in MAPE loops. In particular, in [131] the authors describe various cases in which a human can be part of a MAPE loop. AI Personalization Adaptation falls under what the authors refer to as: ‘System Feedback (Proactive/foreground)’. This type of adaptation is initiated by the system which may send information to the human. The human (i.e. Domain Expert) uses this information to execute the adaptation (by creating the new User Processes necessary). To send the needed information to the Domain Expert, AI Personalization Adaptation takes the gathered knowledge from the Plan phase and gives it to Execute. Execute notifies (Fig. 7.2) both the Development Team and the Domain Expert about the detected cluster change(s) and relays the gathered information.

To determine if a cluster is significantly different from another we use a parameter ***delta***. This parameter is set by the Development Team at design time and determines how different the stored information of one cluster has to be from another one to identify them as unique. The Development Team is notified as a precaution, to double check the change and verify that no errors occurred.

7.5.2 User Driven Adaptation Manager

The main responsibility of the User Driven Adaptation Manager is to receive the User Process from the back-end and convert the contained Abstract Activities into Concrete Activities. A Concrete Activity represents a specific activity that the user can perform, also with the support of smart objects and/or corresponding mobile apps. As an example, *running* is a concrete activity during which the user can exploit a smart-bracelet to monitor their cardio rate as well as a dedicated mobile app to measure the run distance and the estimated burned calories. A Concrete Activity is designed as a class containing multiple attributes that is stored on the smartphone. The attributes are:

- **Selectable**: is True if the User Driven Adaptation Manager or the Environment Driven Adaptation Manager can choose this Concrete Activity, when dynamically refining Abstract Activities; False otherwise. It is set by the user via the user preferences.
- **Location**: it specifies if the activity is performed indoors or outdoors. This attribute is used by the Environment Driven Adaptation Manager to choose the appropriate Concrete Activity according to weather conditions (see Sect. 7.5.5).
- **Activity category**: it defines what type of category does the Concrete

Activity fall under. E.g., for a fitness activity, it specifies a cardio or strength training.

• **Recurrence**: it tracks how many times the user has performed the Concrete Activity in the past. It allows the User Driven Adaptation Manager to have a preference ranking system within all the selectable Concrete Activities.

For each user, the Concrete Activities are derived from their preferences stored in the Third-party Applications Manager. During its nominal execution, the User Driven Adaptation Manager is in charge of refining the Abstract Activities in the User Process into Concrete ones. To do this, it queries the Third-party Applications Manager and exploits its knowledge of the Concrete Activities and their attributes. After completing the task, the User Driven Adaptation Manager presents the personalised User Process to the user as a schedule, where each slot in the vector of Activity categories corresponds to a day. Therefore creating the personalised user schedule of Concrete Activities.

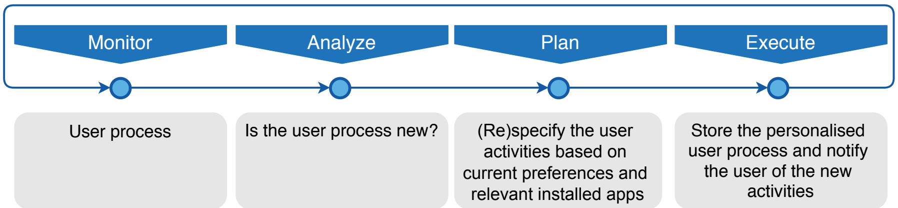


Figure 7.4: *User Driven Adaptation Manager MAPE loop.*

Refining a User Process is required every time that the user is assigned with a new process, to keep up with its improvements and/or cluster change. To this aim, a dynamic User Process adaptation is needed to adapt at run-time the personalised user schedule, in a transparent way and without a direct user involvement. Fig. 7.4 depicts the MAPE loop of the User Driven Adaptation Manager.

Once it accomplishes its main task of refining the User Process, the User Driven Adaptation Manager enters the Monitor phase of its MAPE loop, by monitoring the User Process. The Analyze phase receives the monitored User Process from Monitor. Analyze is now responsible to determine if the user has been assigned a new User Process. If so, the User Driven Adaptation Manager converts the Abstract Activities in this new User Process into Concrete ones, taking into account the user preferences. It makes this conversion by finding suitable Concrete activities during the Plan phase. As all of the Abstract Activities have been matched with a corresponding Concrete activity, the Execute phase makes the conversion, storing this newly created personalised User Process and notifying the user about the new activity schedule.

7.5.3 Smart Objects Manager

This component aims to maintain the connection with the user's smart objects and, if not possible, find alternative sensors to make the e-Health app able to continuously collect user's data, thus to perform optimally. To this aim, it implements a MAPE loop, shown in Fig. 7.5, supporting the dynamic adaptation at the architectural level of the smart objects.

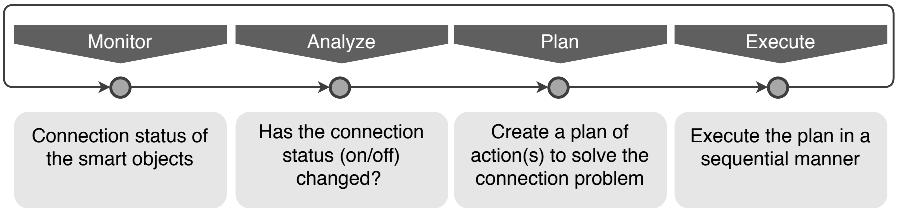


Figure 7.5: *Smart Objects Manager MAPE loop.*

The Monitor phase is devoted to the run-time monitoring of the connection status with the smart objects. Connection problems can be due to either the smart objects themselves, which can be out of battery, or to missing internet, Bluetooth or Bluetooth low energy connectivity. The Analyze phase is in charge of verifying the current connection status (received by Monitor) and see if the connection status with any of the smart objects has changed. During the Plan phase the MAPE will create a sequential plan of actions that the Execute will have to perform. All of the actions are aimed at re-establishing the lost connection or at finding a new source of data (e.g. reconnect, notify the user, find a new source of data). For instance, if the smart-watch connected to the smartphone runs out of battery and the attempts to reconnect to it fail, the Smart Objects Manager will switch to sensors inbuilt in the smartphone (such as the accelerometer).

7.5.4 Internet Connectivity Manager

The main purposes of the Internet Connectivity Manager are to (i) send the Collected Data to the back-end and store them locally when the connection is missing, and (ii) provide resilience to the e-Health app's internet connectivity.

As shown in the MAPE loop in Fig. 7.6, during the Monitor phase the Internet Connectivity Manager runtime monitors the quality of the smartphone's internet connection.

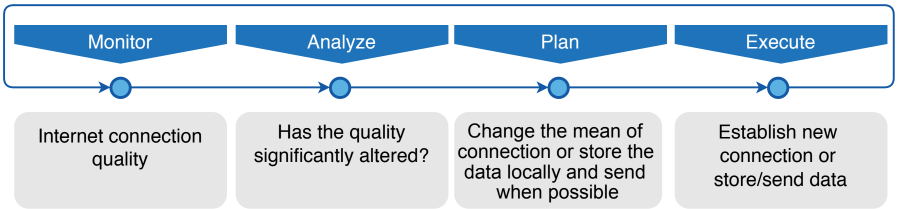


Figure 7.6: *Internet Connectivity Manager MAPE loop.*

Analyze is then in charge of detecting whether a significant connection quality alteration is taking place. If so, the Internet Connectivity Manager enters the Plan phase and it plans for an alternative. The alternative can include switching the connection type or storing the currently collected data locally on the smartphone. As a new connection can be established, the component sends the data to the back-end to be used by the AI Personalization.

7.5.5 Environment Driven Adaptation Manager

One of the objectives of the e-Health app is keeping the users constantly engaged, to ensure that they execute their planned schedule of activities. To this aim, the Environment Driven Adaptation Manager plays an important role, which is essentially supported by its MAPE loop, depicted in Fig. 7.7.

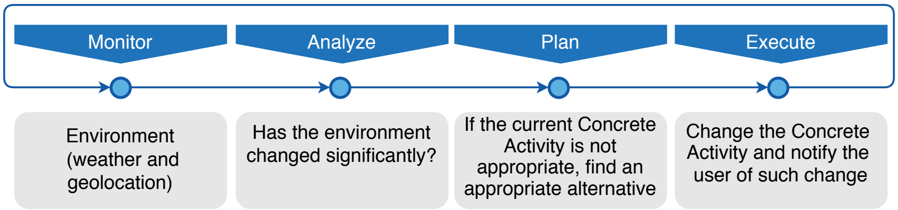


Figure 7.7: *Environment Driven Adaptation Manager MAPE loop.*

The purpose of this component is to constantly check whether the currently scheduled Concrete Activity best matches the runtime environment (i.e., weather conditions) the user is located in. To do so, the Environment Driven Adaptation Manager monitors in run-time the user's environment. The Monitor phase periodically updates the Analyze phase by sending the environment data. This phase establishes if the environment significantly changed. If so, it triggers the Plan phase that verifies whether the currently planned Concrete Activity is appropriate for the user's environment. If it is not, it finds an appropriate alternative and sends the information to Execute. Execute swaps the planned Concrete Activity with the newly found one and notifies the user of this change.

7.6 Goal Model

Goals have been used in many areas of computer science for a long time. For instance, in AI planning they are used to describe desirable states of the world (e.g., [86]) whereas in goal-oriented requirements engineering (GORE [254]) they are used to model non-functional requirements (e.g., [307]). Goals have been also used in self-adaptive systems to express the desired runtime behaviour of systems execution [249; 47]. More recently, goals are used to model personal objectives at users level [286], as done in our work.

As stated before, a User Process is composed of one or more Abstract

Activities, each defined as a vector of Activity categories with an associated *goal*. For each cluster, the Domain Expert defines its User Process and corresponding goals, through the *Editor of Abstract Activities & Goals*.

Table 7.1: *Goal model syntax.*

$$\begin{array}{rl}
\boldsymbol{G_a}::= & \boldsymbol{m_g} \,|\boldsymbol{f_g} \\
\boldsymbol{m_g}::= & \textbf{one_of } \boldsymbol{STRING_SET}\ ?(\boldsymbol{FREQ})\ |<or \leq\ or >or \geq\ or \\
 & = \text{value } \textbf{in } [\boldsymbol{1,\ldots,n}]\ ?(\boldsymbol{FREQ}) \\
\boldsymbol{f_g}::= & \boldsymbol{INTENSITY_{time}}\ ?(\boldsymbol{FREQ})\ |\boldsymbol{INTENSITY_{value}}\ ?(\boldsymbol{FREQ}) \\
 & |<or \leq\ or >or \geq\ or = \boldsymbol{f_g}\ | \\
 & \boldsymbol{f_g} \textbf{ and } \boldsymbol{f_g}\ |\boldsymbol{f_g} \textbf{ or } \boldsymbol{f_g}\ |\textbf{one_of } \boldsymbol{seq}\ \boldsymbol{f_g}\ |\top\ |\bot \\
\boldsymbol{INTENSITY_{time}} ::= & \boldsymbol{seconds}\ |\boldsymbol{minutes}\ |\boldsymbol{hours} \\
\boldsymbol{INTENSITY_{value}} ::= & \boldsymbol{Kcal}\ |\boldsymbol{Km}\ |\boldsymbol{step_count} \\
\boldsymbol{FREQ} ::= & \boldsymbol{TIMES} \textbf{ per day } |\boldsymbol{TIMES} \textbf{ per week } |\boldsymbol{TIMES} \textbf{ per month} \\
\boldsymbol{TIMES} ::= & [\boldsymbol{1,\ldots,n}]\, \forall \boldsymbol{n} \in \mathbb{N}
\end{array}$$

The syntax of our *goal model* is presented in Table 7.1. A goal of an Abstract Activity, namely $\boldsymbol{G_a}$, refers to the type of feature that the Abstract Activity represents (e.g., mood, fitness). At the current stage of our work, we have *mood-based goals* – $\boldsymbol{m_g}$ and *fitness-based goals* – $\boldsymbol{f_g}$.

A mood-based goal defines as objective a desirable mood that the user should reach, considering their specific pathology. A mood-based goal can be specified in two different ways: as a numerical value belonging to a given discrete range, such as $[\boldsymbol{1,\ldots,n}]$, or as a string value belonging to a specific string set, such as *[very sad, sad, neutral, happy, very happy]*. This goal type establishes the target mood that users are expected to reach when performing mood-related activities. Specifically, we use the **one_of** $\boldsymbol{STRING\_SET}$ construct to allow the Domain Expert to define as goal one of the mood among the ones listed in the set $\boldsymbol{STRING\_SET}$, as for instance in (7.1):

$$\boldsymbol{G_a := m_g}\ \textbf{one\_of}\,[\boldsymbol{neutral, happy, very\,happy}] \tag{7.1}$$

When a numerical range is used to describe the user mood, we use relational operators to specify a goal as a value in a subset of the given discrete range. Moreover, for both mood-based goals, the expert can optionally specify the frequency with which the user is asked to register their mood, through the

$\boldsymbol{?(FREQ)}$ construct. The frequency can be expressed in terms of $\boldsymbol{TIMES}$ *per day*, *per week* or *per month*, where $\boldsymbol{TIMES}$ belongs to a discrete range of values, as given in (7.2):

$$\boldsymbol{G_a := m_g \geq 7\,\mathbf{in}\,[1,\ldots,10]\,3\,\mathbf{per\ day}} \tag{7.2}$$

A mood-based goal $\boldsymbol{m_g}$ *succeeds* if it satisfies the relation expressed by the goal. In the presence of a frequency, instead, the user enters more than one mood. In this case, the mood-based goal succeeds if the average computed among the registered mood satisfies the relation expressed by the goal $\boldsymbol{m_g}$. It *fails* otherwise.

A fitness-based goal specifies the required *intensity* and *frequency* with which users should perform fitness-related activities. In particular, the goal model provides two constructs to indicate the intensity, namely $\boldsymbol{INTENSITY_{time}}$ and

$\boldsymbol{INTENSITY_{value}}$. The former is used to express the intensity in terms of duration of the activity (e.g., $\boldsymbol{seconds}$, $\boldsymbol{minutes}$ and $\boldsymbol{hours}$). The latter is used to express the intensity in non time-based terms. Our goal model foresees the use of values such as $\boldsymbol{Kcal}$, $\boldsymbol{Km}$ and $\boldsymbol{step\_count}$. As for mood-based goals, the Domain Expert can optionally specify the frequency with witch the user is asked to perform the suggested activities, via the $\boldsymbol{?(FREQ)}$ construct. Relational operators can be used to specify thresholds values over intensity-based goals. Moreover, *control-flow constructs*, namely **and**, **or** and **one_of**, can also be specified to combine fitness-based goals. These constructs allow us to recursively combine elementary goals, of $\boldsymbol{INTENSITY_{time}}$, $\boldsymbol{INTENSITY_{value}}$ and threshold types, thus to create goals of different complexity. An example is given in (7.3):

$$\boldsymbol{G_a := f_g \geq 1000\,Kcal\,1\,\mathbf{per\ day\ or} f_g > 5\,Km} \tag{7.3}$$

A fitness-based goal $\boldsymbol{f_g}$ of type intensity or threshold *succeeds* if the user performs the suggested activities with the required time-based or value-based intensity. It *fails* otherwise. Goals of type **and** and **or** represent combination of goals and they *succeed*, respectively *fail*, as per the rule defined by the involved logical operators. A goal **one_of** ***seq*** $\boldsymbol{f_g}$ specifies the need of achieving one of the goals in the given sequence. The choice of the goal to target among the available ones can depend on a utility function or a user's choice.

The presented goal model is open and easy to extend. If a new feature different from *mood* and *fitness* is envisaged, it is sufficient to extend the rule related to $\boldsymbol{G_a}$ with a further non-terminal term on the right-hand side of the rule, referring to the new feature, along with one or more associated rules. The ease of use of the goal model, as well as the *Editor of Abstract Activities & Goals* are designed as tools that allow *Domain Experts* to make changes in the tailoring of the app to better meet the interests and needs of the users. This is an important feature of the RA that allows it to better address social sustainability.

7.7 Methodology

As introduced in Sect. 7.2, to design our RA we used the framework and the methodology of Angelov *et al.* [14]. In Table 7.2 we illustrate all questions for each dimension (i.e., context, goals, and design), the answers we gave whilst designing our RA and the rationale for each answer.

In the **goal dimension**, the aim of our RA is providing guidelines for the design of personalised and self-adaptive e-Health apps, as to the best of our knowledge no RA of this type exists in this domain (**G1**).

In the **context dimension**, our RA is devoted to any organisation in the e-Health domain who can benefit from it (**C1**). Particularly, during the design of our RA we have used our collected experience from multiple collaborations with psychologists and e-Health app providers to formulate the requirements needed to be addressed. In the design of the RA, we were the sole designers of the RA (**C2**). The main objective of the RA is to be designed so that it can utilise, in the same architecture, relevant techniques needed to achieve both personalisation and self-adaptation within this domain (**C3**).

Table 7.2: *RA according to the three dimensions: context, goals, design*

Dimension	Values	Rationale
G1: Why is it defined?	Facilitation	Our aim with this RA is to provide guidelines for the design of personalised and self-adaptive e-Health apps.
↓	↓	↓
C1: Where will it be used?	Multiple organisations	Multiple organisations within the e-Health domain.
C2: Who defines it?	Research centres (D),	The RA was designed by the authors who are all researchers.
	User organisations (R), Software organisations (R)	Requirements for this RA were derived by collaborations with domain experts and e-Health app providers.
C3: When is it defined?	Preliminary	The algorithms, goal model and MAPE-loops do not exist in practice yet.
↓	↓	↓
D1: What is described?	Components, algorithms, protocols, etc	Components, CluStream-GT, MAPE-loops, domain model.
D2: How detailed is it described?	Semi-detailed architecture, detailed algorithms and aggregated protocols	The goal model and the software components are semi-detailed, CluStream-GT is detailed, and the MAPE-loops are aggregated.
D3: How concrete is it described?	Abstract elements	At the time of design, our RA mainly abstracts from concrete technologies.
D4: How is it represented?	Semi-formal architecture representation and a formal algorithm	The RA is described according to 42010, CluStream-GT is implemented.

In the **domain dimension**, the main ingredients of our RA are: software components and their connectors, the CluStream-GT algorithm, the MAPE-loops, and the goal model (**D1**). Specifically, the software components and goal model are semi-detailed as they demonstrate implementation-feasibility and a clear objective but are not yet implemented. CluStream-GT is detailed as it is previously published and tested work. The MAPE-loops only demonstrate the general communication and are specified at an aggregated level (**D2**). As our RA is described, we mainly abstract from concrete technologies (**D3**); in fact, the majority of the RA is currently presented in a semi-formal manner with the exception of CluStream-GT (**D4**).

In Table 7.3 we present our final match of the RA with respect to the types/variants (T/V) presented by Angelov *et al.* [14]. In particular, X denotes a match of the architecture values with those in the T/V. As shown, our RA fits one of the architecture variants identified and described by Angelov *et al.* (specifically variant 5.1); this demonstrates its congruence w.r.t. its context, goals, and design. As stated in [14], if a RA can be classified into one of their identified types it has a better chance of being successful (i.e., "accepted by its stakeholders and used in multiple projects"[14]).

Table 7.3: *Final match of our RA to one of the five types identified in [14].*

	T/V	G1	C1	C2	C3	D1	D2	D3	D4
RA	5.1	X	X	X	X	X	X	X	X

7.8 Viewpoint Definition

This Section describes the essential elements of the viewpoint defined to represent Mobile-enabled Self-adaptive Personalised Systems (or MSaPS Viewpoint for short).

Table 7.4: *Elements of the MSaPS Viewpoint*

ELEMENT	DESCRIPTION
Viewpoint description	This viewpoint captures the essential architectural and contextual elements supporting the design of mobile-enabled self-adaptive and personalised systems.
Typical stakeholders	Domain Experts, Software Architects, Members Development Teams, User.

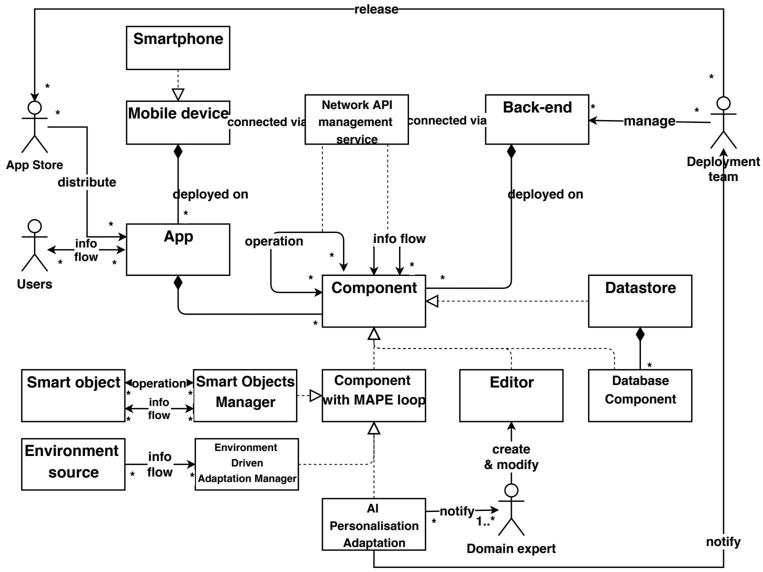
Continued on next page

Table 7.4 – *Continued from previous page*

ELEMENT	DESCRIPTION
Concerns	**C1:** How to extend a mobile app with personalisation and self-adaptation? **C2:** How to integrate external smart objects and environmental information flows? **C3:** How to integrate Domain Expert knowledge into the mobile app's personalisation? **C4:** How to integrate third-party apps as part of the mobile app's personalisation? **C5:** What are the components with MAPE loops and how do they interact? **C6:** Where is the user data stored?

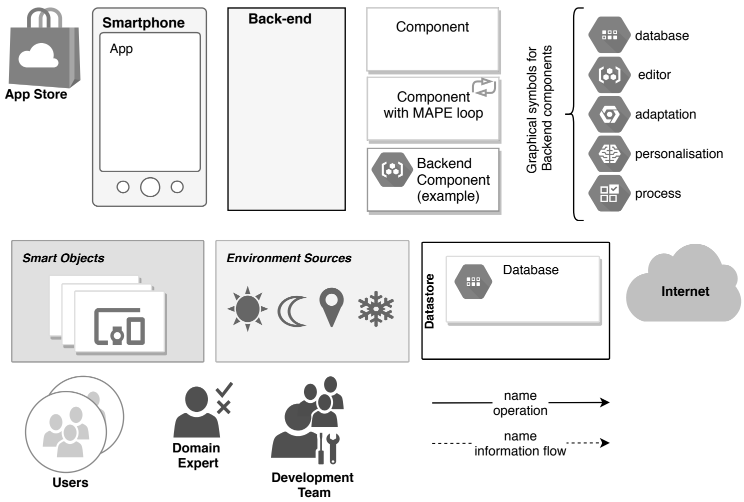
Continued on next page

Table 7.4 – *Continued from previous page*

Element	Description
Meta-model	

Continued on next page

Table 7.4 – *Continued from previous page*

Element	Description
Conforming notation	

We have used it to create the view of our RA for personalised and self-adaptive e-Health Apps as described in Fig. 7.2. It must be noted, however, that the MSaPS Viewpoint is not limited to reference architecture use: one could use it to design specific e-Health mobile-enabled systems, as well as to describe mobile-enabled systems not targeted at e-Health but involving personalisation and self-adaptation.

The MSaPS Viewpoint relies on the guidelines provided in the ISO/IEC/IEEE 42010 Standard [165]. Accordingly, after a short description it frames (cf. Table 7.4) the typical stakeholders, their concerns, the meta-model and the related conforming visual notation. The indication of which stakeholders may have which concerns is further shown in Table 7.5.

CONCERNS/STAKEHOLDERS	User	Domain expert	Developer	Software Architect
C1:Extend App w/ Pers/Adapt			✓	✓
C2:Integrate External Elements			✓	✓
C3:Integrate Domain Knowledge		✓	✓	✓
C4:Integrate Apps			✓	✓
C5:MAPE Interactions			✓	
C6:User Data	✓			

Table 7.5: *Stakeholders and Related Concerns*

7.9 Scenario-based Evaluation

To evaluate how our RA would cover typical usage scenarios, we used the domain expertise learnt from our industrial collaborations and have defined the example case and associated scenarios described in this Section (see Figs. 7.8 and 7.9). For each scenario, we challenged how the RA can be used. Throughout the example we use a hypothetical user named Connor and focus on fitness-based goals.

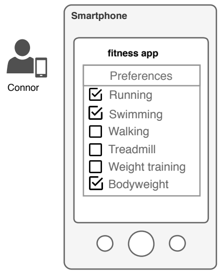
Scenario 1 (Fig. 7.8a). Connor downloads a fitness app that uses our proposed RA. As a first step, he has to input some preferences about the kind of activities he likes the most, complete a questionnaire used to understand his fitness level and give consent for his data to be tracked and used by the app. The fitness app decides on his first weekly schedule of activities. This is a default schedule created by the Domain Expert, in accordance with the information provided by Connor. The default schedule, represented as an Abstract Activity, is adapted by the User Driven Adaptation Manager in accordance with Connor's preferences and supported third-party applications. *This scenario highlights how our RA supports both user level adaptation (where the Abstract Activities* *assigned to Connor are adapted by the User Driven Adaptation Manager), and architecture level adaptation (where the Third-party Applications Manager realises the Concrete Activities by dynamically integrating the specific apps Connor uses on his mobile device).*

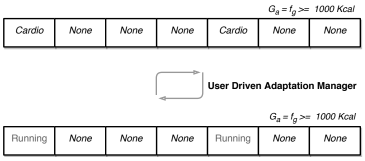
Scenario 2 (Fig. 7.8b). During the first week Connor performs the Concrete Activities assigned to him. This first week is needed by the app to gather enough data from Connor so that the AI Personalization can determine to which macro-cluster Connor belongs. After successfully clustering Connor, the AI Personalization sends an update to the User Process Handler, which is now able to send the appropriate User Process to Connor. By querying the Third-party Applications Manager the Abstract Activity is adapted by the User Driven Adaptation Manager into appropriate Concrete Activities. Like with the default schedule, the two *Cardio* entries are converted into running, whilst the newly given *Strength* one is converted into weight lifting. Furthermore, the new goal he receives is more challenging. *This scenario illustrates the same levels of adaptation as scenario 1, completed by the same components. Additionally, the user level adaptation is further personalised by clustering Connor and the User Process Handler sending him his cluster-related User Process.*

Scenario 3 (Fig. 7.8c). On Monday Connor goes running as suggested by the app. Whilst he is running outdoors, both the WiFi and 4G have no connection. The Internet Connectivity Manager detects this and so decides to store the data locally. As Connor gets back home after completing his run, the WiFi connection is re-established. Aware of this, the Internet Connectivity Manager sends the locally stored Collected Data to the back-end. *This scenario illustrates an architectural level adaptation – performed by the Internet Connectivity Manager by storing the data locally and sending it to the back-end when the internet connection is re-established.*

Scenario 4 (Fig. 7.9a). On Wednesday as Connor is in the gym doing the assigned weight training, the connection with the smart-watch is interrupted. The disconnection is detected by the Smart Objects Manager that at run-time reconnects to the smart-watch allowing the app to resume collecting the data about Connor via the smart object. *This scenario describes an example of*

(a) Scenario 1: Connor installs the application

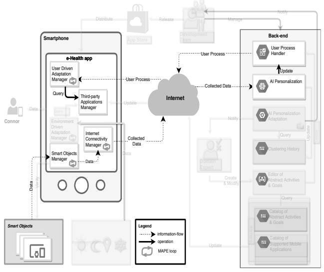


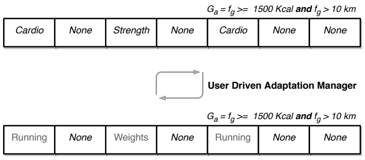


Scenario 1 steps:

1. Connor downloads and installs the e-Health app on his smartphone. He then sets his preferred types of exercise, which the phone stores as selectable Concrete Activities.
2. From the inputted preferences the app selects the appropriate Abstract Activity.
3. The User Driven Adaptation Manager converts the Abstract Activity into Concrete Activities and notifies Connor.

(b) Scenario 2: Connor gets assigned to a macro-cluster and sent the respective User Process.

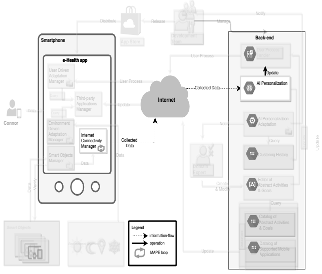


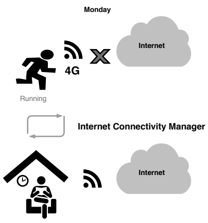


Scenario 2 steps:

1. Connor completed the default Abstract Activity. With the resulting collected data, he gets assigned to a macro-cluster.
2. The User Process Handler sends to Connor his assigned macro-cluster User Process.
3. The User Process is received and processed by the User Driven Adaptation Manager, which then notifies Connor of the new Concrete Activities.

(c) Scenario 3: Connor has no WiFi or 4G connection so the Internet Connectivity Manager saves the data to send at a later point.



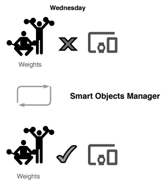


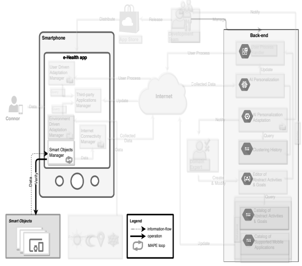
Scenario 3 steps:

1. Connor is starting to run so to complete his first given Concrete Activity.
2. The Internet Connectivity Manager detects that the WiFi and 4G is not connecting to the Internet.
3. The Internet Connectivity Manager saves the `Collected Data` locally during the run. It then sends the `Collected Data` as Connor gets back home and has WiFi connection.

Figure 7.8: *Scenarios 1-3*

(a) Scenario 4: The e-Health app loses connection to Connor's smart-watch. The Smart Objects Manager re-establishes the connection.

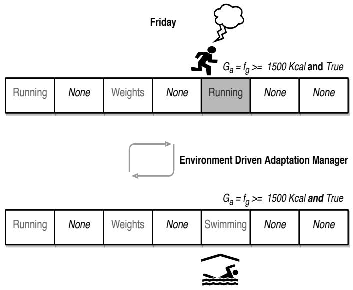


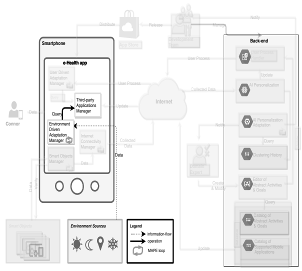


Scenario 4 steps:

1. The e-Health app looses connection to Connor's smart-watch.
2. The Smart Objects Manager detects the changed state of the connection to the smart-watch.
3. The Smart Objects Manager autonomously re-establishes connection to Connor's smart-watch, allowing the e-Health app to resume collecting the data via that smart object.

(b) Scenario 5: The Environment Driven Adaptation Manager detects a change of weather forecast and so adapts the Concrete Activity

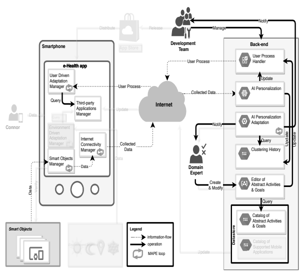


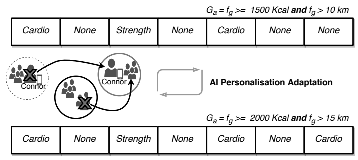


Scenario 5 steps:

1. The Environment Driven Adaptation Manager detects that the weather forecast predicts rain.
2. During the time in which it is forecasted to rain Connor has scheduled Running, an outdoors Concrete Activity.
3. Given this clash the Environment Driven Adaptation Manager adapts Running to Swimming, since it is an indoor activity and doesn't clash with the rain forecast.

(c) Scenario 6: Connor moves to a newly created macro-cluster.





Scenario 6 steps:

1. A new macro-cluster is formed prompting the AI Personalization Adaptation to Notify the Domain Expert.
2. One of the users moving to the new macro-cluster is Connor.
3. Connor therefore gets assigned a new User Process, that the Domain Expert has created for this new macro-cluster.

Figure 7.9: *Scenarios 4-6*

architectural level adaptation - performed by the Smart Objects Manager when Connor's smart-watch is no longer detected by the app.

Scenario 5 (Fig. 7.9b). On Friday, the Environment Driven Adaptation Manager detects that the weather forecast predicts rain for the day. As Connor's scheduled Concrete Activity is running, an outdoor activity, the Environment Driven Adaptation Manager needs to make a run-time adaptation. It queries the Third-party Applications Manager for Cardio activities suitable for indoors. As swimming is the best alternative, it switches running with swimming and notifies Connor of the change, saying that the activity will be carried out via the *swim.com* app. *This scenario focuses on both user level adaptation (when the Concrete Activity is adapted by the Environment Driven Adaptation Manager), and architectural level adaptation (when the Third-party Applications Manager accesses the third-party app).*

Scenario 6 (Fig. 7.9c). Connor has now finished his second week and has successfully reached his assigned goal. In order to maintain the goal engaging and challenging, Connor's success, along with the success of other users, causes the AI Personalization to create a new macro-cluster for them. As the new macro-cluster is one that has never occurred in the system's history, the AI Personalization Adaptation deems this change significant and so notifies the Domain Expert to analyse the new macro-cluster and associate to it a new User Process. The notified Domain Expert makes the new User Process via the web-based Editor of Abstract Activities and Goals. Given the users of the new macro-cluster's success (including Connor), the Domain Expert makes the User Process goal more challenging increasing the amount of $\boldsymbol{Kcal}$ to 2000 and the $\boldsymbol{Km}$ to fifteen (as shown in the figure). This new User Process is sent to the members of the new macro-cluster via the User Process Handler. *This scenario illustrates all three levels of adaptation: the cluster level adaptation to the new macro-cluster done by the AI Personalization Adaptation, the user level adaptation done by the User Driven Adaptation Manager when adapting a new User Process, and the architectural level adaptation done by associating third-party apps to Concrete Activities done by the Third-party Applications Manager.*

7.10 Discussion

It is important to note that our RA is *extensible so to support other domains* beyond fitness and mood. Specifically, the goal model has been designed such that supporting an additional domain can be achieved by adding (i) a new non-terminal term in the root rule $\boldsymbol{G_a}$ and (ii) one or more rules describing the goal within the new domain. Also, many of the existing rules (e.g., $\boldsymbol{FREQ}$) are generic enough to be reused by newly-added rules. On the client side no changes are required, whereas the only components which may need to be customised to a new application domain are: (i) the Editor of Abstract Activities & Goals, so that it is tailored to the different domain experts and the extended goal model; and (ii) the Catalog of Supported Mobile Applications, so that it now describes the interaction points with different third-party apps.

Abstract Activities allow Domain Experts to define *incremental goals* spanning over the duration of the whole User Process. In addition, User Processes are defined at the cluster level (potentially including thousands of users) and can cover large time spans (e.g., weeks or months). Those features make the operation of the RA sustainable from the perspective of Domain Experts, who are not required to frequently intervene for defining new goals or User Processes. Furthermore, these features make the apps adopting our RA socially sustainable on multiple levels. The cluster level defined User Processes allow for tailoring to a 'community' of similar users, empowering them to achieve a better life. On an individual level, the app fine tunes the User Processes to better suite the user's needs and interests; this allows the individual user to better achieve their goals both in the immediate and in the systemic (as defined in [187]). Lastly, these features allow for the larger group of users utilising this RA to all reach the same level of health benefits, as the interventions have been specifically tailored for them for this goal.

Through the conversion from Activity Categories to Concrete Activities, which takes place during the dynamic Abstract Activities refinement, we accommodate both *Type-to-Type* adaptation (e.g., from the *Cardio* Activity Category to the *Running* Concrete Activity) and the most common *Type-to-Instance* adaptation (e.g., by using the Strava mobile app as an instance of the *Running* Concrete Activity). Similarly, a *Type-to-Type* adaptation is reported by Calinescu *et al.* [55] presenting an approach where elements are replaced with other elements providing the *same* functionality but showing a superior quality to deal with changing conditions (e.g., dynamic replacement of service instances in service-based systems). However, we go beyond, by replacing activities with others providing *different* functionality to deal with changing conditions. To the best of our knowledge, this adaptation type is uncommon in self-adaptive architectures, despite quite helpful.

The components of the RA running on the smartphone can be deployed in two different ways, each leading to a different business case. Firstly, those components can be integrated into an existing e-Health app (e.g., Endomondo[3]) so to provide personalisation and self-adaptation capabilities to its services. In this case the development team of the app just needs to deploy the client-side components of the RA as a third-party library, suitably integrate the original app with the added library, and launch the server-side components. The second business case regards the creation of a new meta-app integrating the services of third-party apps, similarly to what apps like IFTTT[4] do. In this case, the meta-app makes an extensive usage of the Third-party Applications Manager component and orchestrates the execution of the other apps already installed on the user device.

Finally, we are aware that our RA is responsible for managing highly-sensitive user data, which may raise severe *privacy* concerns. In order to mitigate potential privacy threats, the communication between the mobile app and the back-end is TLS-encrypted and the payload of push notifications is encrypted as well, e.g., by using the Capillary Project [159] for Android apps, which supports state-of-the-art encryption algorithms, such as RSA and Web Push encryption. Eventually, according to the privacy level required, the components running in the back-end can be deployed either on premises or in the Cloud, e.g., by building on public Cloud services like Amazon AWS and execute them behind additional authentication and authorisation layers.

[3] http://endomondo.com [4] http://ifttt.com

7.11 Conclusions and Future Work

The aim of this chapter was to answer ***T.RQ*4**, namely: *How can AI-based personalisation and self-adaptation be used to create e-Health apps that dynamically adapt to the user and their context?* We do so by proposing a RA for personalised and self-adaptive e-Health mobile apps. The RA achieves self-adaptation on three levels: (i) adaptation to the users and their environment, (ii) adaptation to smart objects and third-party applications, and (iii) adaptation according to the data of the AI-based personalisation, ensuring that users receive personalised activities that evolve with the users' run-time changes in behaviour. This work emphasises how personalisation and self-adaptation within the e-Health domain can be beneficial in addressing social sustainability. By tailoring user interventions we empower mobile app developers to better help their users in achieving better physical and mental health; this leads to increased support for the community of people who suffer from mental and physical illness and are working on increasing their health. The RA therefore achieves what is defined as the core principal of social sustainability in the realm of software-intensive systems. In the next chapter we implement a prototype app using the RA as guidance. We then design and execute two controlled experiments to evaluate its effects on users and users' mobile devices.

8
Empirical Evaluation

Chapter 8 is under review as a journal paper:

Grua, E. M., De Sanctis, M., Malavolta, I., Hoogendoorn, M., & Lago, P. (2021). An Evaluation of the Effectiveness of Personalization and Self-Adaptation for e-Health Apps. Elsevier. Under review (journal).

Abstract - *Context.* There are many e-Health mobile apps on the apps store, from apps to improve a user's lifestyle to mental coaching. Whilst these apps might consider user context when they give their interventions, prompts, and encouragements, they still tend to be rigid *e.g.,* not using user context and experience to tailor themselves to the user.

Objective. To better engage and tailor to the user, in the previous chapter we proposed a Reference Architecture for enabling self-adaptation and AI-based personalisation in e-Health apps. In this chapter we will answer ***T.RQ*5**, specifically: *How do dynamically adaptive e-Health apps affect users and their mobile devices?* To answer this research question, we evaluate the end users' perception, usability, performance impact, and energy consumption contributed by this Reference Architecture.

Method. We do so by implementing a Reference Architecture compliant app and conducting two experiments: a user study and a measurement-based experiment.

Results. Although limited in the number of participants, the results of our user study show that usability of the Reference Architecture compliant app is similar to the control app. Users' perception was found to be positively influenced by the compliant app when compared to the control group. Results of our measurement-based experiment showed some differences in performance and energy consumption measurements between the two apps. The differences are, however, deemed minimal.

Conclusions. Our experiments show promising results for an app implemented following our proposed Reference Architecture. This is preliminary evidence that the use of personalization and self-adaptation techniques can be beneficial within the domain of e-Health apps.

8.1 Introduction

E-health apps have some components that make them unique compared to other health-related systems *i.e.,* (i) can take advantage of smartphone sensors, (ii) can reach an extremely wide audience with low infrastructural investments, and

(iii) can leverage the intrinsic characteristics of the mobile medium (*i.e.,* being always-on, personal, and always-carried by the user) for providing timely and in-context services [119]. However, even with all of these tools available to them, e-Health apps still tend to be *rigid* and not tailored in their interventions and prompts to the user, *e.g.,* the apps are using a fixed rule set to construct their interventions and not considering unique traits and behaviours of the individual user. To solve this problem, we previously proposed a *reference architecture (RA) that combines data-driven personalisation with self-adaptation* [143; 144]. In this paper we extend on this research line by:

- utilising our RA to guide the implementation of an app.
- designing and conducting a user study to investigate end users' concerns related to both usability and perception of an app complying to our RA.
- designing and conducting a measurement-based experiment to investigate the impact on performance and energy consumption that an app complying to our RA has.
- discussing the newly found results and frame them in the broader context of e-Health mobile apps and the usage of personalisation and self-adaptation techniques in this domain.

We conduct two experiments that investigate some concerns that developers and end users of our implemented app would have. To this end, we have formulated four main research questions to empirically assess the impact of personalisation and self-adaptation from (i) the users' perspective and (ii) the system perspective. Our experiment results show that for the user perspective personalisation and self-adaptation techniques have an overall positive impact on the end users' perception of e-Health mobile apps. We saw no apparent impact of these techniques on usability of e-Health mobile apps. From the system perspective our results have found some statistically significant differences in app performance. These differences are too small to realistically impact the user experience of an Android app. Furthermore, our experiments provide evidence that the impact of personalisation and self-adaptation on energy consumption of e-Health mobile apps is negligible.

The chapter is structured as follows.

Section 8.2 describes our study design, with Section 8.2.1 showing how an app was implemented following the guidelines of our RA, and Sections 8.2.2 and 8.2.3 describing the design of our experiments. Section 8.3 explains the results for both experiments. In Section 8.4 we discuss the results. Section 8.5 explains the threats to validity. Section 8.6 describes the related work. Lastly, Section 8.7 concludes the chapter.

8.2 Study Design

As shown in Figure 8.1, our study is composed of three main phases, namely: the instantiation of the RA, the user study (Experiment 1), and the measurement-based experiment (Experiment 2). We describe each step of all phases, its objective, expected input and output, and number of involved researchers.

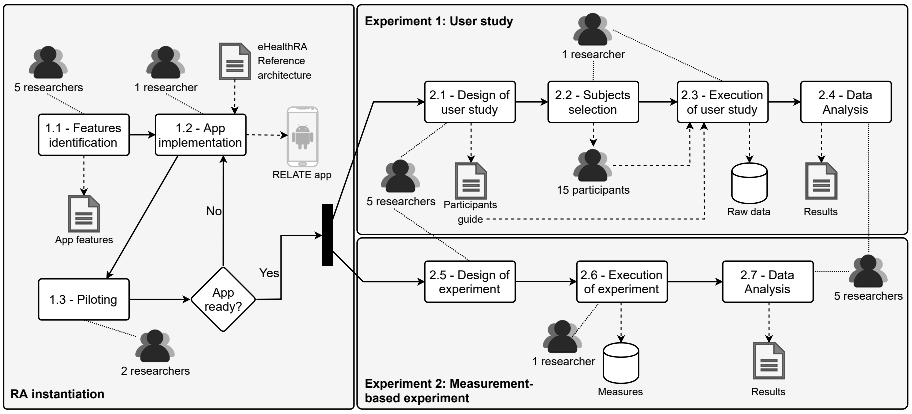


Figure 8.1: *Study Design*

The goal of the **RA instantiation** phase is to design and develop an instance of the RA, which is used in the two experiments. This phase is composed of

three main steps: the identification of the features of the app (step 1.1), its implementation for the Android platform (step 1.2), and its piloting (step 1.3).
– *Features identification (step 1.1).* This step is conducted by all five researchers and has the goal of identifying the features that need to be present in the app implementation. This activity is carried out by taking into consideration our need of keeping the app reasonably simple (so to be used by multiple participants without requiring extensive training), while still having room for personalisation and self-adaptation at runtime. The main output of this step is the list of the app features:

- **F1.** The app needs to provide a list of weekly physical activities to the user.
- **F2.** The user is able to selected their preferred physical activities from a list of available ones.
- **F3.** The app has to be able to determine the environment of the user.
- **F4.** The app needs to change recommended physical activities in accordance to the user environment.

– *App implementation (step 1.2).* The implementation is named RELATE, standing for peRsonalized sELf-AdapTive E-health. RELATE is implemented in Android and its back-end is implemented in Python. The details of the implementation process are discussed in Section 8.2.1.
– *Piloting (step 1.3).* We pilot the implemented app in order to ensure that it can be successfully used in the two experiments. Two researchers different from the one implementing RELATE are involved in this step and they carry out the piloting activities independently from each other. Each researcher installs the app on their mobile device and simulates typical usage scenarios according to the features identified in step 1.1. During usage, they note down apparent bugs and problems and discuss them with the researchers implementing the app. The app would then go back to implementation, to correct the found issues. This cycle continues until the app is deemed ready to be used for the experiments. This step took a total of 14 days.

Once the implementation of the RELATE app is completed and piloted, we can proceed with the design and execution of the two experiments. The complementary nature of the two experiments allows us to carry them out in parallel.

We organise the **user study** into four main phases: the design of the user study, subjects selection, the execution, and the data analysis. Below we describe how the four phases fit together, whereas their detailed description is given in Section 8.2.2.

– *Design of user study (step 2.1).* The goal of this phase is to design a user study that would allow us to understand the impact of personalisation and self-adaptation techniques on the usability and end users' perception of our e-Health mobile app. The design is carried out collaboratively by all five researchers. In addition to the formulation of the goal, research questions and other details, the main observable output of this phase is the *Participants guide*. We hand out the Participants guide to each participant. The Participants guide contains instructions on how to install RELATE on their own personal smartphones, links and instructions on how to fill in the participant surveys, where RELATE can be downloaded from, contact e-mail for participants in need of help.

– *Subjects selection (step 2.2).* After completing the study design we conduct our subjects selection. This step is further detailed in Section 8.2.2.2.

– *Execution of user study (step 2.3).* As we are interested in understanding the influence of the introduction of self-adaptation and personalisation techniques in our e-Health mobile app, we split the set of participants into two groups. One group uses a baseline version of our RELATE app, whilst the other group uses a version containing the aforementioned techniques. We ask both groups to use their app for *four consecutive weeks*. During this user study, each participant completes three different types of surveys, namely: (i) an initial one-time survey for the demographics, (ii) a daily survey reporting their activities and their perception with respect to their app during the whole four-weeks period, and (iii) a final one-time survey about the overall usability and perception of the two versions of the RELATE app. The details about the structure and contents of the surveys are reported in Section 8.2.2.3 and Section 8.2.2.5, respectively.

– *Data Analysis (step 2.4).* The data analysis is carried out once the user study is complete. This phase entails (i) cleaning and organisation of all the raw data produced in the previous step and (ii) its qualitative analysis in order to properly answer the research questions. The analysis of the data is further explained in Section 8.2.2.4.

We organise the **measurement-based experiment** into three main phases: the design of the experiment, its execution, and the data analysis. Below we describe how the three phases fit together and their details are provided in Section 8.2.3.

– *Design of Experiment (step 2.5).* The goal of this step is to precisely define the details of the measurement-based experiment, such as its goal, research questions, dependent and independent variables, hypotheses, statistical tests, etc. The experiment is designed as a one-factor-two-treatments experiment, where the main factor is the presence of personalisation and self-adaptation techniques. The dependent variables are: the energy consumption, the CPU usage, and the memory consumption of the RELATE app. Similarly to step 2.1, this step is carried out collaboratively by all five researchers.

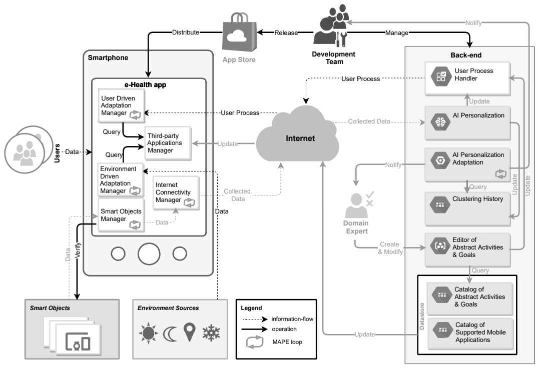
– *Execution of Experiment (step 2.6).* In this phase we execute the experiment according to its design. All runs of the experiment are orchestrated automatically and are carried out in a controlled setting. This allows us to isolate the potential effect of the treatments of the main factor of the experiment on the values of the dependent variables, while having minimal bias from external confounding factors. Further details of the experiment execution are reported in Section 8.2.3.4.

– *Data Analysis (step 2.7).* In this phase we firstly explore the collected measures by graphically visualising them and by performing descriptive analyses. Then, we proceed to check for normality and test for statistical significance, so to answer the statistical hypotheses of the experiment. The detailed explanation of the data analysis is given in Section 8.2.3.3.

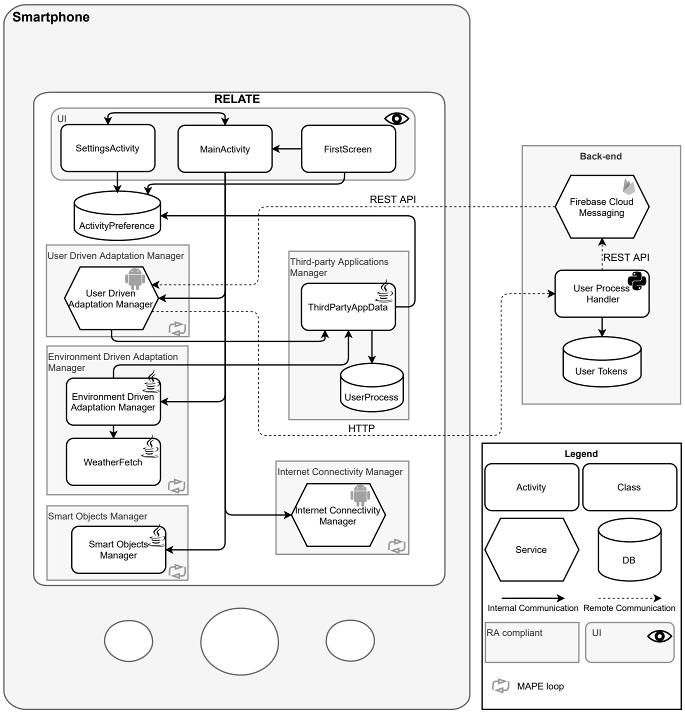
A complete **replication package** is publicly available[1] for allowing independent replication and verification of both the experiments presented above.

[1] https://github.com/S2-group/self-adaptive-ehealth-apps-replication-package

8.2.1 Implementation of the e-Health app



(a) *All of the RA components implemented by RELATE (the gray ones have been omitted)*



(b) *RELATE architecture*

Figure 8.2: *Figures describing the components of the RA used and the RELATE architecture*

In this section we describe the implementation of the RA compliant e-Health app, named RELATE, that we used for our experiments. RELATE is implemented in Android, as Android mobile devices cover the majority of the mobile device sector and the majority of scientific research on mobile software engineering is done on Android [25; 223]. For an explanation of the app's flow the reader is directed to the online material in the **replication package**.

Figure 8.2b shows RELATE's architecture, whose mapping with the RA components is shown in Figure 8.2a. There are three activities in RELATE that together form the UI: the MainActivity, the SettingsActivity and the FirstScreen activity.

– *FirstScreen*. This is the first activity displayed to the user. The sole responsibility of this activity is to present the user with the list of available physical activities and have them choose their preferred ones. After they have made their preference they are redirected to the Main Screen, which is managed by the MainActivity.

– *MainActivity*. This activity is in charge of displaying the Main Screen to the user, as well as instantiating and communicating to most other components present in RELATE. It is also from here that the user can choose to access the settings.

– *SettingsActivity*. This activity is in charge of displaying the app's settings to the user and redirecting them to either adjust their preferred physical activities, read the about page or go back to the Main Screen. Whenever the user makes a change to their preferred activities, the SettingsActivity stores the preferences locally, so that they are available even after the application has been closed by the user.

RELATE contains two services on the user side: the User Driven Adaptation Manager and the Internet Connectivity Manager.

– *User Driven Adaptation Manager*. This service has two main responsibilities: it creates a unique identifier token at installation which it sends to the Back-end and it converts each User Process received from the Back-end. The unique token is used by the Back-end to send the User Process to the correctly paired user. The conversion of the User Process is done by the User Driven Adaptation

Manager in accordance to the self-adaptive loop described in Section 7.2.

The responsibility of sending the token to the Back-end is a deviation from the RA. In the RA the only component to send information to the Back-end is the Internet Connectivity Manager. This change was made to optimize the information flow of RELATE. As with the current implementation of RELATE we don't have the AI Personalization in the Back-end, we decided to use the already created information flow from the User Process Handler to the User Driven Adaptation Manager and add the task of receiving the user token.

– *Internet Connectivity Manager.* This service is started by the MainActivity whenever the app is opened in the foreground. Its main purpose is to monitor and manage the connection to the internet via its self-adaptive loop as described in Section 7.2.

RELATE contains four classes: Smart-Objects Manager, Environment Driven Adaptation Manager, ThirdPartyAppData, and WeatherFetch.

– *Smart Objects Manager.* This class is initialized by the MainActivity whenever the app is launched and has two main purposes: ask the user for the runtime permission for the Bluetooth usage and, monitor and manage the connection to external devices via the self-adaptive loop, in accordance to the RA's description.

– *Environment Driven Adaptation Manager.* This class is also initialized by the MainActivity whenever the app is launched by the user. It has two main purposes: to check what the weather forecast is for the current day and to convert the daily suggested activity if the current user environment calls for it. The class determines the daily weather forecast with the help of WeatherFetch. The Environment Driven Adaptation Manager can also perform a change in suggested activity, as described in the RA, which it then displays to the user via a push-notification.

– *WeatherFetch.* The main resposibility of this class is to determine the weather forecast and deliver that information to the Environment Driven Adaptation Manager. To determine the forecast, it uses the OpenWeather API[2] to retrieve a .json file containing information on the weather forecast for that day. It processes the file and sends the parsed information to the Environment Driven

[2] https://openweathermap.org/api

Adaptation Manager (*e.g.,* Sunny, Rain, Windy, Storm, etc.).
– *ThirdPartyAppData*. This class is a helper class to the User Driven Adaptation Manager and the Environment Driven Adaptation Manager in the conversion of the received UserProcess from the Back-end to a schedule of concrete activities displayed to the user. In this version of RELATE, this class did not interact with other third party apps as described in the RA.

Lastly, the Back-end *User Process Handler* was implemented using Flask in Python. The Flask server would receive the initial unique identifier token sent by the User Driven Adaptation Manager and store it in the *User Tokens* database. The User Process Handler then sends the weekly user process to the app via the use of Google's Firebase Cloud Messaging[3]. As we did not have a Domain Expert involved in these Experiments, the User Process was fixed and saved in the same class file as the User Process Handler. Furthermore, the other components of the Back-end were not implemented in this version of RELATE. Whilst in the future we would want to include all components to our Experiments, for these Experiments we focused our efforts on the application side of the RA, as it is most relevant to our current research questions.

For both Experiments we also used a BaseApp as a comparison to RELATE. It is identical to RELATE apart from not including: the Environment Driven Adaptation Manager, the Internet Connectivity Manager, and the Smart Objects Manager. The BaseApp is therefore not able to provide the functionalities offered by those components. By not having the Environment Driven Adaptation Manager the BaseApp can't adapt the daily physical activity to better suit the user's current environment and will not show the related push notification. Without the Internet Connectivity Manager, the BaseApp is unable to detect and automatically resolve a failure to connect to the internet, as well as notify the user of such failure. By excluding the Smart Object Manager, the BaseApp is unable to verify the current status of the Bluetooth connection or amend problems that may occur with said connection. As most of the excluded functionalities work without the user's direct involvement, the two applications are aesthetically identical as they contain all of the same screens.

[3] https://firebase.google.com/docs/cloud-messaging

8.2.2 Design and execution of Experiment 1 (User study)

8.2.2.1 Goal and Research Questions

We formulate the goal of this experiment by using the goal template provided by Basili *et al.* [27].

> *Analyse* personalisation and self-adaptation techniques *for the purpose of* assessing their impact *with respect to* end users' perception *from the viewpoint of* users, developers, and researchers *in the context of* our Android implementation of RA.

The *goal* of this experiment is to study our RA from the end users' perception. In order to gain a better understanding of the combination of AI and self-adaptation, we identified the types of usability concerns that users have whilst using a system complying to our RA, as opposed to an identical, non dynamically tailored, system. As described in Section 8.2.1, RELATE is implemented by following our RA and so, in our experiments, is our RA compliant system. As our comparison, non dynamically tailored, system we use the implemented BaseApp (described in Section 8.2.1). Specifically, for the scope of these experiments we define dynamic tailoring to be the utilisation of the Environment Driven Adaptation Manager, the Smart Objects Manager and the Internet Connectivity Manager. As the two systems are identical, a part from one factor, using them in our experiments allows us to make a fair comparison of systems, that isolates our one investigated factor: dynamic tailoring.

In the following we present and discuss the research questions we translated from the above mentioned overall goal.

> ***RQ*1.1** – What is the impact of personalisation and self-adaptation techniques on **end users' perception** of an e-Health mobile app?

The main objective of ***RQ*1.1** is to investigate how the inclusion of the aforementioned techniques (personalisation and self-adaptation) can influence the perception that a user has on such an app as compared to their perception

Table 8.1: *Table showing the initial subject selection, number of participants whom dropped-out and final group numbers for each user study.*

	Participants Enrolled	Participant Drop out	End size of Group R	End size of Group B
First user study	20	11	5	4
Second user study	9	3	4	2
Total Participants	29		9	6

of apps who are not personalised and self-adaptive. The knowledge gained by answering this question can be of important use to developers as they design and introduce these techniques in their own applications.

> ***RQ*1.2** – What is the impact of personalisation and self-adaptation techniques on the **usability** of an e-Health mobile app?

The main objective of ***RQ*1.2** is to investigate whether the use of personalisation and self-adaptation techniques can influence the usability of an app, as compared to one that does not use such techniques. By studying how users perceive RELATE, we can better understand the usability concerns specifically related to dynamically tailored systems. These findings will allow researchers and developers in this field to have increased awareness on what usability concerns a user of a dynamically tailored system has.

8.2.2.2 Subjects Selection

As shown in Table 8.1 we recruited 20 participants in the first user study and 9 in the second one. Participants were split into two groups: Group R used RELATE, group B utilised the BaseApp. During the course of the first user study 11 participants dropped out, whilst 3 participants left in the second user study. The first user study was conducted with weekly reminders to complete the given daily survey. For the second study, as a way to try and diminish participant drop out, the reminders were sent out daily. Another factor that might have played an important role in participant drop out was the ongoing lock-down imposed by the government due to the Covid-19 pandemic. During the first user study, the lockdown was a lot stricter and was, in some cases, the main factor in participants dropping out. This was discovered as we contacted the inactive participants after the trail to inquire on the reasons why they did not complete the study. The Covid-19 restrictions were partially lifted during the run of the second user study. We believe that the combination of less restrictive Covid-19 related policies and the daily reminders, aided to diminish the participants' drop out numbers.

8.2.2.3 Design of the Surveys

The initial survey was given to the participants in order to collect general information about them and is formed as shown in Figure 8.3a.

The daily survey is presented on Figure 8.3b. The goal of the daily survey is to understand the level of engagement of the participants and what their opinion on their daily suggested activity was.

The final questionnaire focuses on usability concerns (Figure 8.3c). This questionnaire uses the System Usability Scale (SUS) [45] together with tailored made questions for our particular experiment. All of the questions/state-ments in the daily and final survey, apart from Q13, Q14, Q16 and Q17, are evaluated on a likert scale ranging from 1-5 (1 being strongly disagree and 5 being strongly agree).

8.2.2.4 Data Analysis

For both studies we will be focusing our analysis on the data collected from the final survey, as that most directly addresses RQ1.1 and RQ1.2. For each of the statements we will count and classify each of the categorical responses given on the likert scale. We will then present this analysis in the form of tables. Thereafter, we will analyse if a difference was recorded between the users of the BaseApp and those of RELATE.

Evaluation Goal	Question ID	Question text	RQ
Demographic information	Q1	e-mail address	N/A
	Q2	Name	N/A
	Q3	Surname	N/A
	Q4	Age	N/A
	Q5	Mother-tongue	N/A
	Q6	What job do you do?	N/A
	Q7	How many hours a week do you work-out?	N/A
	Q8	Gender	N/A
	Q9	Android software Version	N/A
	Q10	Phone model/brand	N/A

(a) *Initial survey given to the participants*

Evaluation Goal	Question ID	Question text	RQ
Daily activity	Q11	e-mail address	N/A
	Q12	I am happy with the daily activity suggested to me today	1.1
	Q13	I performed the daily activity suggested to me today	1.1
	Q14	Provide here if you have performed any other physical activity today	N/A

(b) *The daily survey given to the participants*

Evaluation Goal	Question ID	Question text	RQ
Usability	S1	Whilst using it, the app changed to better fit my needs and preferences	1.1
	S2	The changes that the app performed influenced my perception of it for the better	1.1
	S3-12	As defined in the System Usability Scale	1.2
	Q15	I am happy with the daily activity suggested to me today	1.1
	Q16	I performed the daily activity suggested to me today	1.1
	Q17	Do you have any further comments or suggestions?	N/A

(c) *The end survey given to the participants*

Figure 8.3: *Tables listing all of the surveys used in our user study and each question's relationship to the specified research questions*

8.2.2.5 Experiment Execution

Following the initial stage of recruitment, we had the interested participants fill in the initial survey. After, we randomly divided the participants into two groups. Group R used RELATE and Group B used the BaseApp. The participants were then sent an e-mail with attached the .apk file for their system, as well as instructions on how to install it on their own Android devices. Once all participants informed us of the successful installation of the app, we sent them access to the daily survey and sent them their first weekly activity schedule. During the course of the study, we sent e-mail remainders to the participants to fill in their daily surveys. After one month the study was ended. At the end of the study the participants completed the final survey. This designed experiment was executed twice. Once over the month of December 2020 and the second from mid January to mid February 2021. Both executions of the experiment lasted 4 weeks and in both cases the participants were selected via convenience sampling. Due to the Covid-19 pandemic, both studies were conducted with no physical interaction between us and the participants. All correspondence was done via e-mail.

8.2.3 Design and execution of Experiment 2 (Measurement-Based Experiment)

8.2.3.1 Goal and Research Questions

Similarly to the previous experiment, we formulate the goal of this experiment by using the goal template provided by Basili *et al.* [27].

> *Analyse* personalisation and self-adaptation techniques *for the purpose of* assessing their impact *with respect to* resource consumption at runtime *from the viewpoint of* users, developers, and researchers *in the context of* our Android implementation of RA.

In the following we present and discuss the research questions we translated from the above mentioned overall goal.

> ***RQ*2.1** – What is the impact of personalisation and self-adaptation techniques on the **performance** of an e-Health mobile app?

The main objective of ***RQ*2.1** is to investigate how the use of personalisation and self-adaptation could impact the performance of a e-Health mobile app as opposed to one that does not include such techniques. For the purpose of our experiment, we measure performance impact by measuring the CPU usage and memory consumed by the mobile device whilst operating one of the tested systems (either RELATE or the BaseApp). This knowledge can help developers and users of personalised and self-adaptive e-Health mobile apps as performance problems are easily noticed by a user and impact to their experience. These performance problems can be perceived by the user as app sluggishness and non-responsiveness which can lead to user abandonment as they uninstall the app due to frustration and dissatisfaction.

> ***RQ*2.2** – What is the impact of personalisation and self-adaptation techniques on the **energy consumption** of an e-Health mobile app?

The main objective of ***RQ*2.2** is to investigate if the use of personalisation and self-adaptation can have a significant impact on the energy consumption that a e-Health mobile app draws as compared to an identical e-Health app that does not use these techniques. Answering this research question gives important insight to the developer and the user of such apps. The amount of energy consumed by a single app can have great impact on the users' experience, as it could potentially hinder the users' ability of utilising their mobile device all together. If the introduction of these techniques would lead to a high enough energy consumption, it could potentially discourage users from choosing e-Health mobile apps containing personalisation and self-adaptation, something that would be undesirable to a developer.

8.2.3.2 Variables and Hypotheses

This section explains both the independent and dependent variables present in our experiment.

The independent variables in this experiment are two: the type of smartphone used and the type of system installed on it. The type of smartphone used has two treatments: low-end and middle-end. For our low-end device we used a LG Nexus 5X and for the middle-end device we used a Samsung Galaxy J7 Duo (further details on the two smartphones are reported in Section 8.2.2.5). The type of system installed on the smartphone has also two treatments: the system with no dynamic tailoring (the BaseApp) and the system with dynamic tailoring (RELATE). For each execution of one of these systems we measure the below reported dependent variables.

The dependent variables in this experiment are the energy consumed (reported in Joules), the cpu usage (reported as the percentage amount used over the total amount available) and the memory consumed (reported in kilobytes) by either the BaseApp or RELATE.

For each of the above listed dependent variables we formulate the following hypotheses:

- **H1:** We define $\boldsymbol{CPU_B}$ to be the measured CPU usage of the BaseApp and $\boldsymbol{CPU_R}$ to be the measured CPU usage of RELATE. The null and the alternative hypotheses are formulated as follows:
  $\boldsymbol{H1_0 : CPU_B = CPU_R}$
  $\boldsymbol{H1_1 : CPU_B \neq CPU_R}$

- **H2:** We define $\boldsymbol{MEM_B}$ to be the measured memory consumption of the BaseApp and $\boldsymbol{MEM_R}$ to be the measured consumption of RELATE. The null and alternative hypotheses are formulated as follows:
  $\boldsymbol{H2_0 : MEM_B = MEM_R}$
  $\boldsymbol{H2_1 : MEM_B \neq MEM_R}$

- **H3:** We define $\boldsymbol{EC_B}$ to be the measured energy consumption of the BaseApp and $\boldsymbol{EC_R}$ to be the measured energy consumption of RELATE.

The null and alternative hypotheses are formulated as follows:

$\boldsymbol{H3_0 : EC_B = EC_R}$

$\boldsymbol{H3_1 : EC_B \neq EC_R}$

H1 and **H2** investigate the dependent variables for answering RQ2.1, while **H3** aims at answering RQ2.2. All three hypotheses are separately assessed for each of the two smartphones used.

8.2.3.3 Data Analysis

In this experiment we are going to answer each of our research questions in four phases: exploration, normality checks, hypotheses testing, effect size estimation.

Exploration. In this first phase we get an indication of the data collected via the use of descriptive statistics (*i.e.,* mean, median and standard deviation) and boxplots.

Normality checks. We measure the distribution of each data type collected to understand whether we can apply parametric or non-parametric statistical tests. We check whether the data is normally distributed by first visually analysing it with a Q-Q plot and the applying a Shapiro-Wilks statistical test [315] with an $\boldsymbol{\alpha = 0.05}$. As we report in Section 8.3.2, the collected data is not normally distributed.

Hypotheses testing. Given the non normal distribution of the data collected we test our hypotheses by the use of the Mann Whitney U test (with an $\boldsymbol{\alpha = 0.05}$). The Mann Whitney U (also known as the Wilcoxon rank-sum test) is a non-parametric statistical test used to check whether the population of two distributions are statistically equal [234].

Effect size estimation. To statistically test the effect size of the difference found between samples we use the Cliff's Delta statistical test [79]. Cliff's Delta is a non-parametric statistical tool used to calculate the effect size without making assumptions on the distributions compared.

8.2.3.4 Experiment Execution

In this subsection we explain how we conducted our experiments to measure CPU, memory and energy consumption. As shown in Figure 8.4, for each repetition

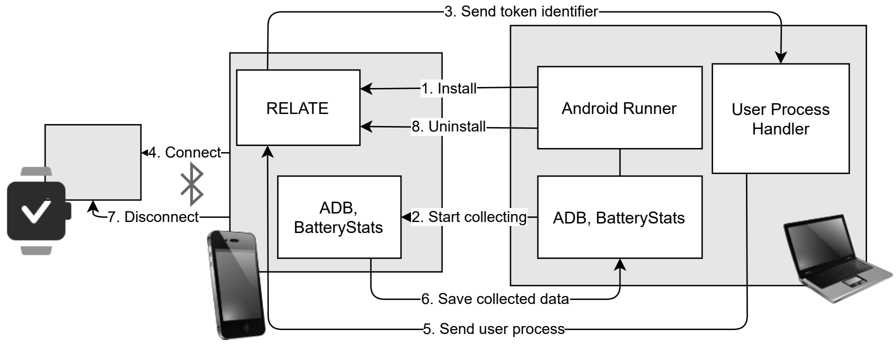


Figure 8.4: *Execution of one repetition of the experiment*

of our experiments we used: a laptop, one of the two chosen smartphones, a smartwatch, and an internet connection.

– *The Laptop.* It is running Ubuntu 16.04LTS and had the following hardware specifications: RAM 16GB, CPU i7-6700HQ @ 2.60GHz * 8, Intel HD Graphics 530. In order to automate our repetitions we installed Android Runner (AR) on the laptop [223]. AR is a framework that allows users to automatically execute measurement-based experiments on both native and web apps running on Android devices.

– *The Smartphones.* The Android devices used are two: a LG Nexus 5X smartphone and a Samsung Galaxy J7 Duo. The LG smartphone has a 1.8 Ghz hexacore ARM Cortex A53 & Cortex A57 cpu with 2 GB of RAM running Android 6.0.1. This model is chosen to represent the possible performance impact that these systems can have on an older android smartphone. The Samsung smartphone has a 1.6 Ghz octacore ARM Cortex A73 & Cortex A53 cpu with 4 GB of RAM running Android 8.0.0. This smartphone is chosen to represent a mid level android smartphone.

Each repetition starts with installing either the BaseApp or RELATE on the smartphone (step 1). Once installed, AR starts measuring the system's consumption of CPU, memory and energy (step 2). For measuring the CPU and memory consumption AR uses Android Debug Bridge[4] (adb). For measuring the energy consumption it instead uses the Android Batterystats profiler [101]. AR then follows a series of screen taps and gestures that are engineered to be indicative of a worst case scenario. Within the profiling session, the scenario goes through the completion of the initial screen, giving of the necessary run-time permissions, the re-connection to the paired smartwatch (step 4), the receiving of the weekly user activities (step 5). When the scenario is terminated, AR stops profiling the Android device (step 6). After profiling, AR makes the necessary steps to set the device back to how it was before the installation of the system (step 7 and 8). Lastly, AR waits 2 minutes before running another repetition of the experiment. This wait is introduced to allow the device to 'cool-off' and go back to an idle state; this break minimises inconsistencies between repetitions. We ran 50 repetitions for each combination of system and smartphone, leading to a total of 200 repetitions.

8.3 Results

In this section we report on the results for both Experiments 1 and 2.

8.3.1 Results of Experiment 1 (User study)

We will now discuss the results of our user studies, organised by research question and following the method and tools described in Section 8.2.2.4. The results of the final survey for the first user study are shown in Figure 8.5a. The participants answering the final survey could rate each statement from agreement (by rating it a 5) to disagreement (by rating it a 1). It is important to note that the scoring can mean something different per statement. For some statements disagreement is desired and for other statements we are looking for user agreement. Participants in Group R used RELATE and Group B

[4] https://developer.android.com/studio/command-line/adb

participants used the BaseApp. Figure 8.5b illustrates the final survey results for the second user study.

8.3.1.1 Investigating end users' perception (RQ1.1)

The statements related to RQ1.1 are S1 and S2 shown in Figure 8.5, and Q12, Q15, Q13, and Q16 defined in Figures 8.3c and 8.3b.
– *Whilst using it, the app changed to better fit my needs and preferences (S1).* In both user studies participants in Group R tended to agree more with this statement. Participants in Group B did instead agree less. This means that generally participants in Group B did not find the app to change for the better, implying that they either did not think the app changed or that it changed for the worse.
– *The changes that the app performed influenced my perception of it for the better (S2).* In the first user study participants in Group R rated this statement with either disagreement or neutrality. Whilst members of Group B showed more agreement to the statement. In the second user study members of both groups showed their opinion to be either neutral or in agreement. These findings imply that in the first user study participants using RELATE did not believe that the changes performed by the app influenced their opinion of it for the better. This could mean that either the changes they perceived didn't impact their opinion of RELATE or shifted their opinion for the worse. On the contrary, members of Group B as well as all participants in the second user study seemed to have received the perceived changes positively. This indicates that, especially for users of RELATE in the second study, the changes offered and perceived by the users are considered a positive aspect of the app.
– *I am happy with the daily activity suggested to me today (Q12 and Q15).* We have grouped Q12 and Q15 as they are the same question but posed in two different surveys (*i.e.,* the daily and the final surveys). For the first user study, we gathered the following average (median) and standard deviation for Group B: 4.290 (5) and 1.062. Whilst, for Group R we gathered the following average (median) and standard deviation of 3.9256 (4) and 1.039. In the second user study, we gathered the following results of average (median) and standard

Final Survey, First User Study (9 participants in total)		BaseApp (B)					RELATE (R)				
		1	2	3	4	5	1	2	3	4	5
S1	Whilst using it, the app changed to better fit my needs and preferences	1	2	2	0	0	1	0	2	1	0
S2	The changes that the app performed influenced my perception of it for the better	1	0	1	2	0	2	1	2	0	0
S3	I think that I would like to use this app frequently.	1	0	2	1	0	3	2	0	0	0
S4	I found the app unnecessarily complex.	4	0	0	0	0	4	1	0	0	0
S5	I found the app easy to use.	0	0	0	3	1	0	0	0	2	3
S6	I think that I would need the support of a technical person to be able to use this app.	4	0	0	0	0	5	0	0	0	0
S7	I found the various functions in this system were well integrated.	1	0	2	1	0	1	2	1	0	1
S8	I thought there was too much inconsistency in this app.	4	0	0	0	0	0	2	3	0	0
S9	I would imagine that most people would learn to use this app very quickly.	0	0	0	1	3	0	0	1	1	3
S10	I found the app very cumbersome to use.	4	0	0	0	0	2	0	2	1	0
S11	I felt very confident using the app	0	0	1	1	2	0	0	1	2	2
S12	I needed to learn a lot of things before I could get going with this app.	3	1	0	0	0	4	1	0	0	0

(a) *The final survey answers entered by the participants of the first user study (1 - disagreement, 5 - agreement).*

Final Survey, Second User Study (6 participants in total)		BaseApp (B)					RELATE (R)				
		1	2	3	4	5	1	2	3	4	5
S1	Whilst using it, the app changed to better fit my needs and preferences	0	0	1	1	0	0	0	1	3	0
S2	The changes that the app performed influenced my perception of it for the better	0	0	1	1	0	0	0	2	2	0
S3	I think that I would like to use this app frequently.	0	0	2	0	0	1	1	0	2	0
S4	I found the app unnecessarily complex.	2	0	0	0	0	3	1	0	0	0
S5	I found the app easy to use.	0	0	0	1	1	0	0	0	2	2
S6	I think that I would need the support of a technical person to be able to use this app.	0	2	0	0	0	4	0	0	0	0
S7	I found the various functions in this system were well integrated.	0	0	1	1	0	0	0	2	2	0
S8	I thought there was too much inconsistency in this app.	0	2	0	0	0	0	2	2	0	0
S9	I would imagine that most people would learn to use this app very quickly.	0	0	0	1	1	0	0	0	2	2
S10	I found the app very cumbersome to use.	1	1	0	0	0	2	1	1	0	0
S11	I felt very confident using the app	0	0	1	1	0	0	1	1	2	0
S12	I needed to learn a lot of things before I could get going with this app.	1	1	0	0	0	4	0	0	0	0

(b) *The final survey answers entered by the participants of the second user study (1 - disagreement, 5 - agreement).*

Figure 8.5: *Recorded ratings for the final survey for both user studies.*

deviation from Group B: 3.706 (4) and 0.47. For Group R we got an average (median) and standard deviation of 3.655 (4) and 0.971. In both user studies the results gathered indicate a minimal difference in happiness with the suggested daily activities, with the participants in Group B seemingly happier.

– *I performed the daily activity suggested to me today (Q13 and Q16).* For these questions we calculated the percentage of times that participants of a group reported performing their suggested physical activity of the day. In the first user study participants of Group B reported following the suggested daily activity 82.26% of the time. Participants of Group R reported performing their suggested activity 73.4% of the time. In the second study, the participants of Group B reported performing their daily suggested activity 58.82% of the time, whilst participants of Group R agreed with the suggested activity 66.37% of the time. These findings show that in the first user study, participants using RELATE recorded performing their suggested activities less often than those not using this app. The opposite can be seen from the results of the second user study.

8.3.1.2 Investigating usability (RQ1.2)

In this section we report on the data obtained to answer RQ1.2. The statements related to this research question are S3 to S12 shown in Figure 8.5. We have grouped the statements together per overarching topic: ease of use of the app, app cohesiveness, and likely hood of using the app in the future. – *Ease of use of the app.* This topic groups the following statements:

- I found the app unnecessarily complex. (S4)
- I found the app easy to use. (S5)
- I think that I would need the support of a technical person to be able to use this app. (S6)
- I would imagine that most people would learn to use this app very quickly. (S9)

  - I felt very confident using the app. (S11)
  - I needed to learn a lot of things before I could get going with this app. (S12)

For this group of statements, we find that the participants rated their utilised apps easy to use (S5, S6, S9, S11) and understand (S4, S6, S12). This is true no matter if the participants are from Group R or Group B. Furthermore, there is no significant difference in opinion between the participants of the first user study as compared to those of the second user study. As participants from both groups had no significant difference in scoring their version of the app, we also understand that dynamic tailoring did not make it harder for the user to use and understand the app.

– *App cohesiveness.* This group is comprised of the following statements:

  - I found the various functions in this system were well integrated. (S7)
  - I thought there was too much inconsistency in this app. (S8)

The findings for S7 in the first user study show that, no matter which group the participants belonged to, they were split across the scale. The opposite is seen for the results of the second user study, here the opinion on the integration of system functionalities was more focused and seen more positively. More cohesive are the results of S8. For both user studies the participants found their apps to not have too much inconsistency. In particular users of RELATE were more neutral towards this statement than the users of the BaseApp, which all rated the statement with disagreement.

– *Likelihood of using the app in the future.* Lastly, in this topic we group the following statements:

  - I think that I would like to use this app frequently. (S3)
  - I found the app very cumbersome to use. (S10)

The findings for S3 are non homogeneous and inconsistent across the two user studies. In the first one, most participants either disagreed with the statement of were neutral about it. In the second one, half of Group R disagreed with the statement, whilst the other half agreed; the members of Group B all rated the statement neutrally. These findings show a non homogeneous consent in the opinions of the participants. We, instead, find cohesion in the results given for S10, as in both user studies the majority of participants did not find their version of the app to be cumbersome to use. We further elaborate on these results in Section 8.5.4.

8.3.2 Results of Experiment 2 (Measurement-Based Experiment)

In this section we report on the results obtained for the measurement-based experiment. We will be discussing the results per research question answered, following the procedure reported in Section 8.2.3.3.

8.3.2.1 Impact on performance (RQ2.1)

Exploration. The performance data measured for the LG smartphone are shown on Figure 8.6. For the CPU measurements, we see no clear difference between the two apps. The mean (median) and standard deviation for the BaseApp are: 14.24 % (10.55 %) and 7.90 %, whilst for RELATE they are: 13.69 % (11.00 %) and 7.15 % respectively.

We can observe a difference between the distribution of the memory usage of the two apps, with RELATE consuming more memory. The mean (median) and standard deviation of the BaseApp are 65266.40 kB (64675.35 kB) and 1140.62 kB, respectively, and the descriptive statistics of RELATE are: 67912.26 kB (67859.79 kB) and 519.67 kB, respectively.

The performance measured for the Samsung smartphone are shown in Figure 8.7. As shown in subfigure 8.7a, we have found no difference in CPU usage by the two systems. The mean (median) and standard deviation of the BaseApp are: 13.90 % (11.96 %) and 9.34 %. Similarly, the descriptive statistics for the CPU consumption of RELATE are: 12.61 % (11.00 %) and 7.93 %. Similar to the LG smartphone, in this case RELATE tends to use more memory than the

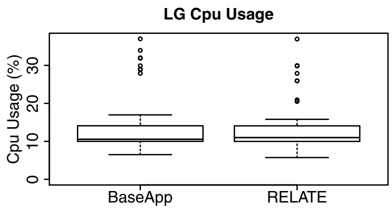
BaseApp (seen in subfigure 8.7b). The mean (median) and standard deviation for the BaseApp are: 47963.58 kB (44531.24 kB) and 6363.78 kB, respectively. Differently, RELATE reported a mean (median) and standard deviation of: 55492.90 kB (54478.22 kB) and 2898.77 kB, respectively.

Check for normality. Figure 8.6 shows the Q-Q plot against the normal distribution for both the CPU and the memory consumption data measured on the LG smartphone. Several measures fall far away from the reference line, indicating that the collected measures are not normally distributed. To further confirm our observation we carried out the Shapiro-Wilks test on all four datasets. For RELATE CPU measurements the test returned a p-value of 3.545e-08 and, for the BaseApp we achieved a p-value of 4.72e-09. For the memory measurements of RELATE we obtained a p-value = 0.03345 and for the BaseApp the p-value is 1.89e-05. Therefore, in all cases, we can reject the null hypothesis stating that these samples come from a normal distribution.

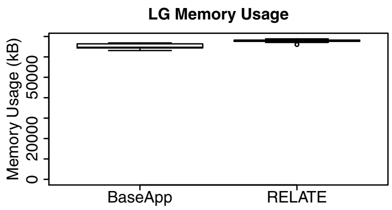
With Figure 8.7 we illustrate the Q-Q plots for the performance measurements taken on the Samsung smartphone.

Our Shapiro-Wilks tests confirm the Q-Q plots. With a returned p-value of 6.262e-10 for the CPU measurements of RELATE and a p-value of 4.161e-09 for the BaseApp. For the memory usage RELATE had a p-value = 1.285e-11 and the BaseApp returned a p-value of 4.661e-07. We can therefore reject the null hypothesis stating that the Samsung smartphone CPU usage data comes from a normal distribution.

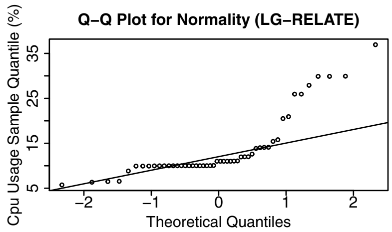
Hypothesis testing. As stated in subsection 8.2.3.3, we utilise the non-parametric Mann–Whitney U test to determine whether we can reject our stated null hypotheses (formulated in subsection 8.2.3.2). Starting by examining the measurements collected for the LG smartphone; the p-value returned for the comparison between BaseApp and AdaptiveSystem on the CPU consumption is equal to 0.48. As this p-value is above the significance threshold ($\boldsymbol{\alpha}$=0.05), we cannot reject the null hypothesis $\boldsymbol{H1_0}$. When applying the statistical test to the memory consumption values of the two apps we obtain a returned p-value of 2.54814e-17. As this value is smaller than our chosen $\boldsymbol{\alpha}$, we can reject our null hypothesis $\boldsymbol{H2_0}$. The above findings are similar for the Samsung



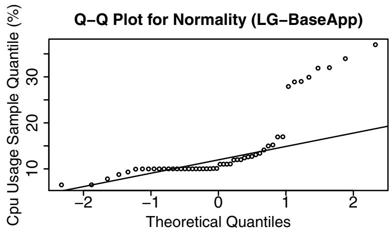
(a) *CPU usage for the LG smartphone*



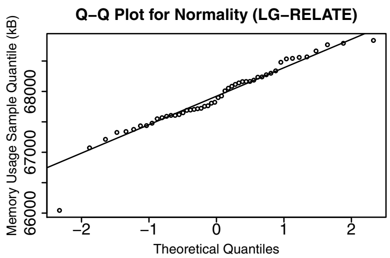
(b) *Memory usage for the LG smartphone*



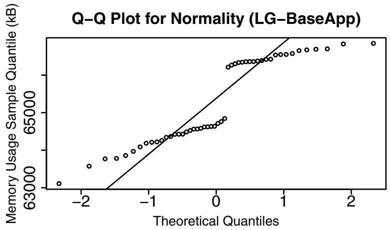
(c) *Q-Q plot for the CPU usage on the LG smartphone for RELATE*



(d) *Q-Q plot for the CPU usage on the LG smartphone for the BaseApp*

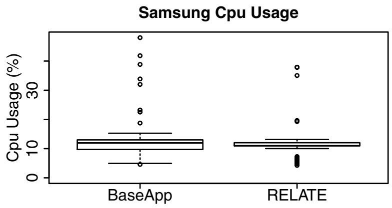


(e) *Q-Q plot for the memory usage on the LG smartphone for RELATE*

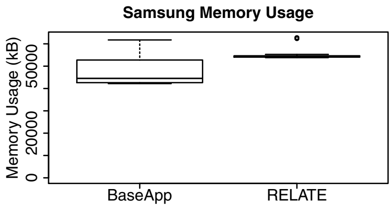


(f) *Q-Q plot for the memory usage on the LG smartphone for the BaseApp*

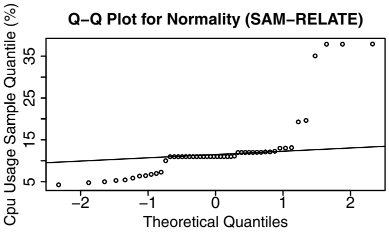
Figure 8.6: *All plots related to performance measurements for the LG smartphone*



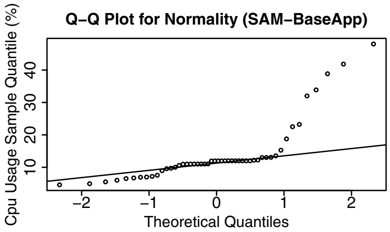
(a) *CPU usage for the Samsung smartphone*



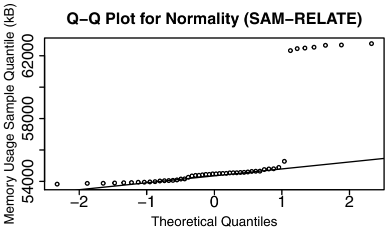
(b) *Memory usage for the Samsung smartphone*



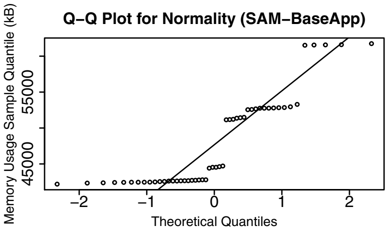
(c) *Q-Q plot for the CPU usage on the Samsung smartphone for RELATE*



(d) *Q-Q plot for the CPU usage on the Samsung smartphone for the BaseApp*



(e) *Q-Q plot for the CPU usage on the Samsung smartphone for RELATE*



(f) *Q-Q plot for the memory usage on the Samsung smartphone for the BaseApp*

Figure 8.7: *All plots related to performance measurements for the Samsung smartphone*

smartphone, where the returned p-values for the comparison of the two apps on CPU consumption and memory usage are 0.36 and 9.67143e-13, respectively. This means that we cannot reject the null hypothesis $\boldsymbol{H1_0}$, but we can reject the null hypothesis $\boldsymbol{H2_0}$.

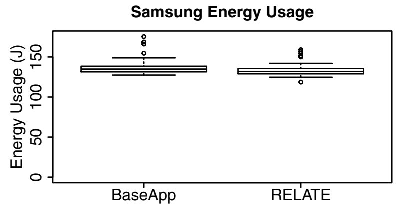
Effect size estimation. As a follow up to the use of the Mann-Whitney U test, we determine the effect size of the differences found. As stated in subsection 8.2.3.3, we use Cliff's Delta to do so. A **large** effect size is found when investigating the difference in memory consumption between the BaseApp and AdaptiveSystem for both the Lg smartphone (0.98) and the Samsung smartphone (0.83).

8.3.2.2 Impact on energy consumption (RQ2.2)

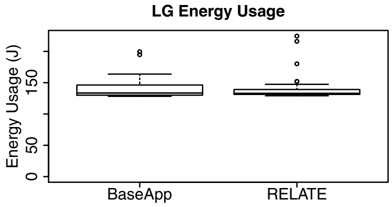
Exploration. Figure 8.8b shows the distribution of the energy consumption of the two apps running on the LG smartphone. We see no apparent difference in the energy consumption between the two apps. Indeed the mean (median) and standard deviation for the BaseApp are: 139.54 J (133.53 J) and 15.34 J. For RELATE the mean (median) and standard deviation are: 139.32 J (132.95 J) and 18.78 J.

For the Samsung smartphone we observe a slight difference in energy consumption between the two systems (shown in Figure 8.8a). We can observe that RELATE consumes less energy than the BaseApp; we will be further discussing this finding in Section 8.4. The mean (median) and standard deviation of the energy consumption for the baseline app are: 137.51 J (134.83 J) and 9.99 J; whilst the descriptive statistic for RELATE are: 133.96 J (131.85 J) and 8.28 J respectively.

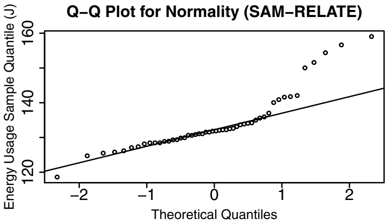
Check for normality. Figures 8.8d and 8.8f show the Q-Q plots against the normal distribution for the energy consumption measured on the LG smartphone. Both plots show that the data collected is not normally distributed. To further corroborate our finding, the Shapiro-Wilks test done on RELATE's dataset returns a p-value of 6.115e-12 and the BaseApp case gives a p-value of 5.904e-09. Therefore we can reject the null hypothesis of the data belonging to a normal distribution.



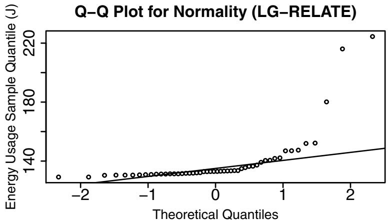
(a) *Energy usage for the Samsung smartphone*



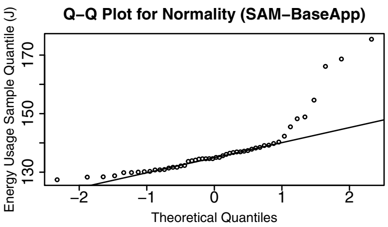
(b) *Energy usage for the LG smartphone*



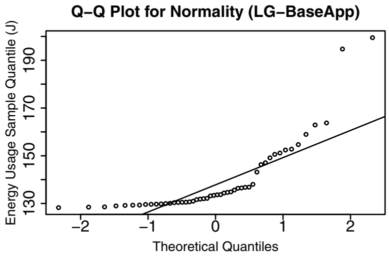
(c) *Q-Q plot for the energy usage on the Samsung smartphone for RELATE*



(d) *Q-Q plot for the energy usage on the LG smartphone for RELATE*



(e) *Q-Q plot for the energy usage on the Samsung smartphone for the BaseApp*



(f) *Q-Q plot for the energy usage on the LG smartphone for the BaseApp*

Figure 8.8: *All plots for the energy usage measured*

Figures 8.8c and 8.8e illustrate the Q-Q plots against the normal distribution for the energy consumption measured on the Samsung smartphone. As the plots indicates, the data is not normally distributed. This is confirmed by the Shapiro-Wilks test, as RELATE returned a p-value of 1.258e-05 and the BaseApp returned a p-value equal to 3.329e-08. We can therefore reject the null hypothesis that the energy consumption measured from the Samsung smartphone comes from a normal distribution.

Hypothesis testing. We start by using the Mann–Whitney U test on the energy consumption data collected on the Lg smartphone. The p-value returned by the test is 0.73, as this value is above our chosen $\boldsymbol{\alpha}$, we cannot reject the null hypothesis $\boldsymbol{H3_0}$ and therefore we find that the difference in energy consumption between the BaseApp and RELATE on the Lg smartphone is not statistically significant. When running the test on the energy consumption data for the Samsung smartphone we obtain a p-value of 0.009. As the p-value is below our $\boldsymbol{\alpha}$ threshold, we can reject the null hypothesis $\boldsymbol{H3_0}$ and find that the difference in energy consumption between the BaseApp and RELATE is statistically significant.

Effect size estimation. Here we use Cliff's Delta to follow up on the findings gathered in our hypothesis testing. The difference found on the Samsung smartphone can be classified as small (*i.e.,* -0.30).

8.4 Discussion

8.4.1 Discussion on Experiment 1 (User study)

We start by discussing the results on the *end users' perception* (RQ1.1). Participants using RELATE tend to agree more than those using the BaseApp that the app changed to better fit their needs and preference (S1). This result is interesting, as it suggests that users of RELATE (i) noticed the adaptation of the app and (ii) found those changes to be useful.

Most participants rated the statement "the changes that the app performed influenced my perception of it for the better" neutrally or approvingly (S2).

Only the users of RELATE in the first user study also stated disagreement with the statement. The disagreement with the statement can either mean that the participants found the changes to modify their perception of RELATE for the worse or that they did not make a difference in their perception of the app. Given the agreement recorded for the previous statement (S1), we find it unlikely for the disagreement on this statement, S2, to have a negative connotation: as this would contradict the positive implications found with S1. Therefore, we can conclude that the changes performed by the app were overall seen as either non impactful to the users' perception of the app or as a positive influence.

Regarding how happy the users were with their daily activities, our results found little difference between users using the BaseApp and those using RELATE in both user studies. The only difference seems to be the fact that participants in Group B appeared to be somewhat happier. This, however, is not reflected in the adherence to performing the suggested activities. Here, the two user studies show opposite results with the first one showing participants in Group B performing their suggested activities more often, whereas the second user study showed Group R more often performing their daily activities. Therefore, these last results seem to be inconclusive. This could be due to the simplicity of the suggested daily activities and the minimal dynamic tailoring that is done with them in this current version of the implemented apps. As future work, it would be important to include all of the Back-end components in the RA in order to be able to better personalize the daily suggested activities to the participants using RELATE. This further implementation of dynamic tailoring could lead to a wider observed difference between the two groups of participants as the two apps will be further distinguished from each other.

In summary, the results we have obtained indicate that **personalisation and self-adaptation techniques have an overall positive impact on the end users' perception of e-Health mobile apps**. Therefore, developers and researchers whom are interested in end users' perception, can successfully adopt these techniques in their own e-Health apps.

Lastly, privacy was a relevant concern during our experiments. We addressed privacy concerns by having all of our participants give us their personal information willingly and understand that it would be saved and used for the purposes of this work. To respect privacy regulations, the data presented to the public, via the replication package, has been anonymised.

We will discuss now the results related to our investigation on the *usability* of our e-Health mobile app (RQ1.2). Across all of the statements analysed, we see a pattern of agreement between all of the participants, no matter the group they were assigned to. This is interesting, as it points to dynamic tailoring not being a determining factor to how the participants responded to the survey. We observed only with statements S3 (*i.e.,* "I think that I would like to use this app frequently") and S7 (*i.e.,* "I found the various functions in this system were well integrated") that the participants did not show a clear trend or consensus, and were instead more distributed along the Likert scale. Given these results, we can conclude that **there seems to be no apparent impact caused by personalisation and self-adaptation techniques on usability of e-Health mobile apps**. As discussed previously, a future implementation of RELATE containing the complete Back-end components from our RA might help surface differences that were not recorded in this study, as this new version of RELATE would include the full range of dynamic tailoring advocated by our RA and would therefore increase the difference between our two tested systems (RELATE and our BaseApp).

8.4.2 Discussion on Experiment 2 (Measurement-Based Experiment)

We start the discussion by elaborating on our results for RQ2.1, namely: "What is the impact of personalisation and self-adaptation techniques on the *performance* of e-Health mobile apps?". For both devices, RELATE tends to use more memory than the BaseApp. This is understandable, as RELATE contains the adaptive components that the BaseApp does not (*i.e.,* Environment Driven Adaptation Manger, Smart Objects Manager, and Internet Connectivity Manager). These components require the utilisation of the smartphone's memory in order to carry out their business logic. As an example, the Environment Driven Adaptation Manager needs to assess what current day of the week it is, what the weather forecast for that day is and if it needs to change the currently recommended daily activity. Having said so, the difference of the amounts of used memory is negligible when put into the context of the total amount of memory that these devices have. The difference between the averages for the LG smartphone is 2645.9 kB (over a total of 2 GB available) and for the Samsung smartphone is 7529.3 kB (over a total of 4 GB available). This difference, whilst shown to be statistically significant, has no practical implication over the user experience of the apps. We can therefore conclude that the 'price' paid in terms of memory consumption for the benefits of adding dynamic tailoring is worthwhile.

Our results also show a difference in the CPU usage levels between the two examined apps on the Samsung smartphone. However, RELATE seems to be the one consuming the least amount of CPU. If we take the difference between RELATE CPU usage average and the one of the BaseApp we get a difference of -1.288 (the CPU measurement is quantified as a percentage of the total amount of CPU). Whilst our analysis has shown this difference to be statistically significant, we argue that such a small difference in CPU usage would have no impact on the user experience. In conclusion, **whilst our results have found some statistically significant differences in app performance, these differences are too small to realistically impact the user experience of an Android app**.

Regarding RQ2.2, namely: "What is the impact of personalisation and self-adaptation techniques on the *energy consumption* of e-Health mobile apps?". Only for the Samsung smartphone we found a statistically significant difference in energy consumption between the two apps. This difference is, however, not expected as it shows RELATE to be consuming *less* energy than the BaseApp. Upon closer inspection, we notice that the difference in average energy consumption between the BaseApp and RELATE is equal to 3.6 J. Just like with the differences found for the memory and the CPU, this discovered difference is so small that it will not impact the user experience in a practical sense. In conclusion, **our experiments provide evidence that the im-**

pact of personalisation and self-adaptation techniques on the energy consumption of e-Health mobile apps is negligible.

Overall, our findings for RQ2.1 and RQ2.2 show that using personalisation and self-adaptation techniques in e-Health mobile apps has no adverse effect on both performance an energy consumption. This should encourage app developers and researchers working in this field to adopt these techniques in their own e-Health apps as they provide a great range of extra functionalities with little to no impact on the resources of the user's smartphone.

8.5 Threats to Validity

8.5.1 External Validity

– *Experiment 1 (User study).* There is a threat to generalisability as the sample of participants for our experiment was limited. Because of this, the presented results are not meant to be final but rather as an exploration of these topics. Further work with a larger sample of participants would be needed to draw more conclusive results.

– *Experiment 2 (Measurement-Base Experiment).* To minimise the threat to external validity we ran our experiment on two different types of smartphone. The smartphones chosen are intended to be a representation of a low-end and a middle-end device. This diversification of devices should better capture the real world scenario. Having said so, the use of a newer smartphone could possibly lead to different results and conclusions. We therefore encourage further experiments to further minimise this threat to validity.

8.5.2 Internal Validity

– *Experiment 1 (User study).* To mitigate the threat to internal validity we implemented the two applications to be as close as possible, leaving dynamic tailoring as the sole difference. Furthermore, the participants for both groups R and B were recruited in the same manner and are all of a comparable demographic (therefore mitigating possible selection bias).

– *Experiment 2 (Measurement-Base Experiment).* There are a number of factors that can influence the measurements we have collected in our experiments *i.e.,* brightness of the screen, distance to the internet router, distance to the Bluetooth smartwatch and background processes. We designed our experiments so to minimise as much as possible these factors. We maintained the brightness of the screen, the distance to the internet router and the distance to the Bluetooth smartwatch fixed across all repetitions. To mitigate the impact of uncontrollable background processes we performed 50 repetitions for each experiment case, mitigating the bias that one spike in background processes can have over our overall readings. Lastly, maturation can influence the data collected in the experiments. In our case maturation is the changes that occur in the smartphone as the experiment is running (*e.g.,* memory usage, CPU heat generation and impact on its performance). In order to mitigate it, we imposed a waiting time of 2 minutes between each repetition. We also cleared any data that was gathered during the course of a repetition, to maintain the status of the smartphone identical across experiments.

8.5.3 Construct Validity

– *Experiment 1 (User study).* To minimise the threat to construct validity, we defined all of the details regarding our experiment design a priori (*e.g.,* research questions, data analysis methodology, variables).
– *Experiment 2 (Measurement-Base Experiment).* Here we also defined everything regarding our experiment design and methodology a priori.

8.5.4 Conclusion Validity

– *Experiment 1 (User study).* To mitigate the threat to conclusion validity, we had all 5 researchers involved in the data analysis of the results obtained for this experiment. This mitigates an individual bias and interpretation of the results. Furthermore, we offer a complete replication package to the public. Allowing for independent replication of our experiment.
– *Experiment 2 (Measurement-Base Experiment).* To minimise the threat to

conclusion validity we have used statistical analysis to more objectively draw our conclusions on the experiment. Lastly, we offer a complete replication package to the public. Allowing for independent replication of our experiment.

8.6 Related Work

The literature does not provide studies like ours about the empirical evaluation of self-adaptive (e-Health) mobile apps. However, we could identify several works close to our research area, namely *self-adaptive mobile apps in eHealth.*

Self-adaptation represents a suitable method to detect and deal with (potentially impactful) unexpected context changes. In the field of *mobile* apps, it is even more challenging due to, *i.e.,* mobile phones resource constraints (*e.g.,* battery level, network traffic). Grua *et al.* [148] give an overview of self-adaptability for mobile apps by providing a classification framework for understanding, classifying, and comparing approaches. In the same field, Grassi and Mirandola [140] proposed a new perspective for looking at known approaches, to make mobile apps resilient and anti-fragile to changes. They suggest to look at distinct and complementary approaches (*e.g.,* self-adaptation and cyber foraging [196]), often considered separately, from the Tao perspective (*e.g.,* Yin and Yang strategies), to fully exploit their respective potential.

The need for *runtime adaptation* is exacerbated in the eHealth domain where adapting to the user-needs and context may be of crucial importance, *i.e.,* to properly and promptly react to monitored patients activities. Ballesteros *et al.* [23] present a wearable patient-monitoring system for tele-rehabilitation, supporting traditional rehabilitation therapies by providing valuable information for the evaluation, monitoring, and treatment of patients. The system follows a goal-oriented self-adaptation approach based on dynamic software product lines (DSPL) and it makes use of a set of self-adaptation policies enabling it to dynamically self-configure its internal behaviour to the current context of the patient, while maintaining the system efficiency (*e.g.,* optimising battery consumption). Differently than our RELATE app, the used adaptation policies do not influence the usability or end users' perception, since end users do not directly interact with the system. They only make use of the system's devices and wearable (*e.g.,* knee motion sensor), used to monitor and recognise the activity the user is performing and to collect data to trigger re-configuration.

Mizouni *et al.* [240] focus on the design and development of self-adaptive applications that sense and react to contextual changes (*e.g.,* environment, device status) to provide a *value-added user experience*. The authors present a framework defining a systematic approach to model dynamic adaptation of mobile apps behaviour at runtime by using SPL concepts and offering feature priority based dynamic adaptability. The framework is evaluated through an application supporting doctors on the move to have access to patients' files, report medical conditions, prepare for intervention and advise hospital about patient needs and arrival. A similar application targeting doctors on the move is proposed by Preuveneers *et al.* [284]. In this study, the authors focus on how to deliver the right patient's information at the right time under variable connectivity and limited resource availability. Probabilistic models and dynamic decision networks are used to improve the user experience and on-device resource utilisation. Differently than [240], we have defined a reference architecture for personalised and self-adaptive e-health apps, by leaving a certain degree of freedom to developers about design decisions and adaptation strategies. Moreover, in contrast to both [240] and [284], our RELATE app is not intended for healthcare professionals and caregivers, but for end-users. For this reason, usability and users' perception concerns are quite relevant, since they might impact the constant and active commitment of end users.

Lopez *et al.* [211] make use of non-obtrusive monitoring technology in their context-aware mobile app delivering self-adaptive persuasive messages that stimulate the medication adherence, by exploiting real-time physiological data (*e.g.,* heart rate). In our e-Health app, in contrast, runtime adaptation and personalisation are used to create the best possible conditions for users to keep active in their activities, by considering their current context and preferences (*e.g.,* by suggesting their preferred indoor activities if it is raining), by guaranteeing a good level of usability.

Usability of e-Health mobile apps is also a matter of the user experience provided by the apps' user interface. In this context, Raheel [290] proposes a set of adaptive mechanisms able to monitoring the user's behaviour w.r.t. a mobile phone (*e.g.,* determining the distance between the user and the screen) and adapting the interface accordingly. For evaluation purposes, the author presents a medical adaptive mobile app aiming to help the elderly remember taking their medicine at specific times. Experiments show the effectiveness of adaptive user interface in improving usability and acceptance of the mobile app. Improving usability is also one of the aim of our RELATE app. However, our RA brings other instruments in support of usability that go beyond the usability of the user interface (*e.g.,* the goal model, user process adaptation, architectural adaptation) and, simultaneously, it aims to guarantee that the personalisation and self-adaptation techniques we use do not degrade the app usability.

All the above reviewed studies share with our work the exploitation of self-adaptation techniques in eHealth mobile apps. Similarly to us, some of them especially focus on the usability of the apps, from the perspectives of the end users (*e.g.,* [211; 290]) and the experts (*e.g.,* [240; 284]). However, in the context of *mobile* apps, adaptation engines must satisfy the energy efficiency requirement. According to Cañete *et al.* [56], energy consumption also depends on the execution context (environment, devices status) and how the user interacts with the application. Indeed, despite the hardware consumes the energy, the software (*e.g.,* adaptation mechanisms) is responsible for managing hardware resources and its functionality, thus affecting the energy consumption. This demands for energy-efficient adaptation. Although some of the reviewed work (*e.g.,* [23; 284]) aim to maintain the system efficiency, differently from them, our study investigates the impact of the used personalisation and self-adaptation techniques on the performance and energy consumption of an e-Health app. This is made through an empirical evaluation and by comparing two instances of our RELATE app, with and without personalisation and self-adaptation, respectively. Results clearly show that applications built on top of our RA, exploiting several MAPE loops and dynamic, personalised user processes, guarantee energy-efficient adaptation.

8.7 Conclusions

The goal of this chapter was to answer ***T.RQ*5**, namely: *How do dynamically adaptive e-Health apps affect users and their mobile devices?* We do so by building upon the RA and test an implemented prototype app that complies to it. We call this prototype RELATE, standing for peRsonalized sELf-AdapTive E-health. We designed and executed two experiments: a user study, to test user concerns regarding usability and app perception, and a measurement-based experiment to test concerns related to performance and energy consumption. Both experiments focused on studying the impact of self-adaptation and personalisation on their respective independent variables. To be able to isolate these variables, RELATE was tested again an identical app, lacking dynamic tailoring, which we named BaseApp. In our user study, our results show that end users' usability and perception is not harmed by the introduction of dynamic tailoring and is instead made better for the case of usability whilst for user perception we couldn't find any significant difference. In our measurement-based experiment, we concluded that for both performance and energy consumption the differences measured were never at such a scale to cause real world usage consequences.

9 Conclusion

The research presented in this thesis aimed at investigating how we can help overcome the rigidity and partial tailoring currently seen in e-Health mobile applications (e-Health apps). We proposed that AI-based personalisation and software self-adaptation can be used together to achieve this research goal. We started by analysing the current state of the art in reinforcement learning (RL) for personalisation and classified the used techniques, their properties and their shortcomings. We identified two shortcomings relevant to e-Health: the need to obtain large amounts of data to reach an optimal policy and the possibility of user disengagement during the exploration phase. We addressed the first limitation by proposing that RL is used together with clustering. To solve the current lack of online clustering algorithms for e-Health data, we created and evaluated our own state of the art algorithm (*i.e.,* CluStream-

GT). To tackle the second limitation of RL, we investigated which machine learning models are better suited in predicting user engagement. We then explored and categorised the current state of the art of software self-adaptation techniques. After choosing to use the MAPE (Monitor-Analyze-Plan-Execute) loop technique we propose a reference architecture (RA) for personalised and self-adaptive e-Health apps. This RA utilises several self-adaptive components, in combination with the previously reported AI techniques, to provide state of the art dynamic adaptation. We lastly explore the social sustainability impact of our proposed RA, as well as empirically evaluating the effectiveness of an app implemented following the guidance of our RA.

9.1 Discussion

In this section we are going to discuss and reflect on the work done throughout this thesis. We divide our discussion to reflect the contributions brought forth.

A rigorous map into the current state of the art use of RL for personalisation: We started by conducting a systematic literature review (SLR) of RL applications for personalisation used in different applications domains. The goal of the SLR was to present an overview and categorisation of the state of the art in RL for personalisation, the settings, solution architectures and evaluation strategies used. Overall, our results found an increase in the use of RL for personalisation over time. Furthermore, we found RL to be a suitable paradigm for personalisation of systems on an individual level, using collected user data. These findings are valid across domains, including e-Health. Our findings also shed a light on some of the existing short-comings in the use of RL for personalisation. Relevant to this thesis, was the discovery that the majority of RL models can be classified as one of two ways: 1) one RL model used on all of the users and their data (*i.e.,* one-size-fits-all) or 2) using one model per user (*i.e.,* on an individual level). The one-size-fits-all model is faster at finding a suitable policy, as it has more data to use, however it does it by trading the level of personalisation that it can deliver to each user. This can be problematic in e-Health as a high level of individual personalisation is desired in order to better help the user achieve their health goals. The opposite method found (*i.e.,* one model per user) could achieve a long term higher level of personalisation, which is desired in e-Health. However, as the model is trained only on the information of the individual user, it could then take too long to collect enough data to reach the desired policy: possibly disengaging the user before it can reach the desired level of personalisation. To tackle this shortcoming, we propose to use clustering together with RL creating a cluster-based RL model.

Data-efficient and effective techniques for personalisation: We designed a study to compare various clustering algorithms and distance metrics for the use of cluster-based RL for personalisation in e-Health. The goal of this study was to find which combination resulted in the most effective and efficient way of performing cluster-based RL. We then compared the best found combinations with the one-size-fits-all method and the per individual RL model. We compared these techniques on data created with the use of a simulation environment, generating data imitating the daily schedule of an individual. The goal of the RL personalisation was to send interventions at the correct moment in the schedule of the individuals, so that they would accept and perform them. The results of our study found that derived features as a distance metric normally gave us the best results, with some times the use of dynamic time warping performing equally if not better. This result was consisted, no matter the clustering algorithm that was used. Furthermore, in our study we demonstrate how the best found cluster-based RL methods can outperform the level of personalisation offered by the current state of the art. This result is achieved by also tackling the user disengagement that could occur with the use of a RL model per individual as we are able to reach an optimal personalisation policy faster. More testing would be needed to understand if, over longer periods of time, cluster-based RL would still perform better. It is possible, as time progresses, that using on RL model per individual could surpass the level of personalisation offered by cluster-based RL. If that were to be the case, it would be interesting to explore the use of cluster-based RL as a first step to a later switch to one RL model per individual.

To expand on our cluster-based RL findings, we decided to address the discovered lack of clustering algorithms tailored for e-Health. We did so by creating an online clustering algorithm capable of clustering growing timeseries and so suited for the e-Health domain. The algorithm was created by modifying the already existing data stream clustering algorithm CluStream and so named ours CluStream-GT (CluStream for *G*rowing *T*imeseries). We empirically evaluated CluStream-GT against other state of the art online clustering algorithms, by using them on both artificially generated and real-life datasets. Our results found CluStream-GT capable of addressing the shortcomings of the current state of the art, but also able to cluster data more efficiently whilst remaining comparatively effective. With the creation of CluStream-GT, we now have a better suited clustering method for the domain of e-Health and one that could help further improve cluster-based RL for personalisation in e-Health. In order to further validate our findings it will be important to expand our testing on other datasets and analyse what any potential differences in performance may arise. Furthermore, whilst we have evidence that CluStream-GT will be a good fit in combination with RL for personalisation in e-Health, empirical experiments are needed in order to validate this claim. These types of experiments would also allow us to test how well CluStream-GT aided cluster-based RL performs compared to other state of the art methods (*e.g.,* deep reinforcement learning)

Machine Learning models to predict user engagement in mobile apps: To combat the potential user disengagement that can occur when using RL based personalisation, we studied how to predict user engagement. We explored if and which machine learning techniques are most effective at predicting user engagement, as we can use user engagement as a method to understand if the given personalisation is liked by the user. The understanding is that, with better liked personalisation the user will remain engaged with the app. If the personalisation is not performing as intended, the user will consequently disengage from the app. In our investigation we used collected data from a real world app. We tested and evaluated four different machine learning approaches whom were all aided by the use of clustering. Our results showed that the random forest and boosted-tree algorithms were the best in predicting user engagement. We also observed that we achieve a desirable level of accuracy utilising usage dynamics and features that can be obtained by apps from various domains. A significant feature for the models, was location. We are therefore confident that our findings could apply also to apps from domains where location plays an important role, such as e-Health (*e.g.,* fitness). This however, will have to be tested as feature importance may shift according to the domain. Furthermore, the best performing models in our tested setting might not remain such in other domains.

A rigorous map into the current state of the art use of self-adaptation for mobile apps: As the goal of this thesis is to investigate the use of AI for personalisation and software self-adaptation, we conducted a SLR on self-adaptation in the context of mobile applications. The purpose of this SLR was to identify the state of the art techniques used for self-adaptation in apps. We did so by creating a classification framework that allowed us to categorise the state of the art algorithms and methods found. From this SLR we identified a few important shortcoming in the current state of the art. We found few approaches of self-adaptation that, as a goal, targeted non-technical goals (*e.g.,* promoting user behavioural change and lifestyle improvements). This type of adaptation would be crucial to develop for e-Health apps, as their main goal is the betterment of the users' health and well being. We also identified a lack of adaptation techniques adapting because of changes occurring in third-party apps and smart-objects surrounding the user. Being able to trigger adaptation from changes coming from these devices would be an important step in further adapting to the users' context, as these devices are part of, and capture, the current state of the users' context. Lastly, we found a lack of empirical testing done within the field of self-adaptation for apps.

An RA for personalised self-adaptive e-Health apps: In order to overcome the identified shortcomings in the the state of the art self-adaptation for apps and combine these solutions with the personalisation techniques developed, we propose a Reference Architecture (RA). The goal of this RA is to combine AI personalisation and self-adaptation under one architecture that can be used to guide the development of dynamically adapting e-Health apps. Within our RA we achieve self-adaptation on three levels: adapting to connected smart object and third-party applications, adapting to the users and their environments, and adapting to the output of the AI achieved personalisation. With this work we show how the combination of self-adaptation and personalisation can be beneficial to users in the e-Health domain and how it can address social sustainability. By providing smarter and more well-rounded tailoring, e-Health apps can help users achieve better mental and physical health. We are therefore increasing the support offered to the community of people whom suffer from physical and mental illness.

In order to empirically evaluate some of the benefits that our RA could achieve, we implemented and then tested a prototype app using the RA as a guide for the implementation. For the evaluation we designed and performed two experiments: a user study and a measurement-based experiment. In both experiments, our prototype app (which we refer to as RELATE), was compared to an almost identical app (called BaseApp) lacking of adaptation techniques. For the user study we recruited human participants which we randomly assigned into one of two groups. Group R used RELATE, the app with adaptation techniques, and group B used the BaseApp. We asked the participants to completed a number of questionnaires, over the course of a month of use. As a result of our study we found no difference in the rated end users' perception between the two groups and also recorded a better end users' usability for RELATE. In our second experiment, the measurement-based one, we concluded that the differences in performance impact (*i.e.,* CPU and memory consumption) and energy consumption between the two apps were so small that the user wouldn't be able to perceive a difference. Whilst further work is needed to validate our results on other aspects affected by the use of RELATE, our results encourage the use of AI-base personalisation with self-adaptation in e-Health apps.

9.2 Thesis Research Questions Answered

The goal of this thesis is to understand how AI-based personalisation and self-adaptation can be used together for designing and developing e-Health mobile apps. To achieve this goal we defined a number of research questions. In this section we answer each one of them.

T.RQ*1** ***How can RL-based personalisation for e-Health be improved? We propose the use of cluster-based RL to improve the efficiency and efficacy of current state of the art RL. We compare a number of clustering algorithms and distance metrics to find which variation of cluster-based RL performed best. Our results find the use of derived features from a user's timeseries data lead to the best RL policy, regardless of the clustering algorithm used. This finding also applied when comparing this cluster-based RL approach to the current state of the art. Our results therefore show that the use of cluster-based RL can lead to improvements to RL based personalisation for e-Health.

T.RQ*2** ***How can online-clustering be used to efficiently and effectively cluster e-Health data? We propose a novel online clustering algorithm (*i.e.,* CluStream-GT) tailor made for e-Health data. We test CluStream-GT against the state of the art, by using both artificial and real life e-Health timeseries datasets. Our results show that CluStream-GT can cluster up to 95% faster, whilst being comparably effective in the clusters generated. Furthermore, CluStream-GT can cope with specific difficulties of clustering e-Health data, that no other state of the art algorithm can do.

T.RQ*3** ***How can we predict user engagement in apps? We compare multiple machine learning models aided by clustering. We show how, on our collected dataset, two of the models (*i.e.,* random forest and boosted tree) can predict user engagement with good accuracy using usage dynamics and features relate to apps (*e.g.,* user location).

T.RQ*4** ***How can AI-based personalisation and self-adaptation be used to create e-Health apps that dynamically adapt to the user and their context? We propose a unique RA for personalised and self-adaptive e-Health apps. With this RA we show how AI-based personalisation and self-adaptation can be used together to create e-Health apps that are capable of adapting to the user and their context whilst personalising to their needs and end goals. To achieve this, the RA is comprised of several components that address the shortcomings identified in both the fields of AI-based personalisation as well as software self-adaptation, in the context of e-Health.

T.RQ*5 *How do dynamically adaptive e-Health apps affect users and their mobile devices? We conduct two experiments to empirically test how the use of dynamic adaptation on an e-Health app effects users and their mobile devices. For the first experiment we conduct a user study and analyse the effects that dynamic adaptation has on end users' usability and perception. For our second experiment we conduct a measurement-based experiment to empirically evaluate how dynamic adaptation impacts energy consumption and performance. In both experiments, the dynamically adaptive e-Health app was compared to an identical non dynamically adaptive e-Health app. Our results show how the use of dynamic adaptation on an e-Health app does not negatively impact it in terms of end users' perception, energy consumption and performance. Furthermore, we find that dynamic adaptation can improve end users' usability.

9.3 Future Work

Whilst we found promising results in the combination of AI-based personalisation and self-adaptation in tackling the current shortcomings of e-Health apps, we believe there is still room for improvement and future testing. We hereby list some points for future work:

1. ***Test the identified machine learning models for user engagement on datasets from other app domains, including e-Health apps.*** In this work we concluded that our findings will likely remain valid across domains were user location plays an important role. For future work, it will be important to test our conclusions and better understand

how our models perform to these types of domains, especially e-Health.

2. ***Expand our testing done on cluster-based RL by including real world e-Health datasets.*** In our original experiments we tested cluster-based RL with the use of data generated from a simulation environment. For future work, it is important to expand our testing by using real world e-Health datasets to better understand how cluster-based RL will perform in a real world scenario when compared to other state of the art RL models.

3. ***Investigate how our proposed RA can be expanded to other e-Health sub-domains.*** Our proposed RA is currently designed for the e-Health sub-domains of fitness and mental health. We have, however, designed the most critical components of the RA (such as the Goal Model) to be easily expanded to other e-Health sub-domains. For future work, it is important to test our design and how our RA can guide the implementation of dynamically adaptive apps across the whole domain of e-Health.

4. ***Replicate our empirical experiments with a dynamically adaptive app that uses all of our RA's components.*** In the future, it will be important to replicate our empirical experiments on an app that uses all of the components present in our RA. This will allow researchers and developers working in the field of e-Health apps to get a better understanding of the advantages and potential drawbacks of using an RA such as the one we propose.

Bibliography

[1] (2018). *Android connectivity.* Available at https://developer.android.com/training/basics/network-ops/connecting.

[2] (2019). The industrial internet of things volume G1: reference architecture. Industrial Internet Consortium.

[3] Abe, N., N. Verma, C. Apte, and R. Schroko (2004). Cross channel optimized marketing by reinforcement learning. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*.

[4] Abowd, G., A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles (1999). Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pp. 319. Springer.

[5] Aggarwal, C. C., J. Han, J. Wang, and P. S. Yu (2003). A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pp. 81–92. VLDB Endowment.

[6] Ahrndt, S., M. Lützenberger, and S. M. Prochnow (2016). Using personality models as prior knowledge to accelerate learning about stress-coping preferences: (demonstration). In *AAMAS*.

[7] Almardini, M., A. Hajja, Z. W. Raś, L. Clover, D. Olaleye, Y. Park, J. Paulson, and Y. Xiao (2015). Reduction of readmissions to hospitals based on actionable knowledge discovery and personalization. In *Beyond Databases,*

Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery, pp. 39–55. Springer.

[8] Anderlucci, L. (2012). *Comparing different approaches for clustering categorical data*. Ph. D. thesis, University of Bologna.

[9] Andersson, J., R. De Lemos, S. Malek, and D. Weyns (2009). Modeling dimensions of self-adaptive software systems. In *Software engineering for self-adaptive systems*, pp. 27–47. Springer.

[10] Andrade, G., G. Ramalho, H. Santana, and V. Corruble (2005). Automatic computer game balancing. *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05*.

[11] Andreu-Perez, J., C. C. Poon, R. D. Merrifield, S. T. Wong, and G.-Z. Yang (2015). Big data for health. *IEEE journal of biomedical and health informatics 19*(4), 1193–1208.

[12] Android (2017). Android developer portal.

[13] Andrzejak, R. G., K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger (2001). Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E 64*(6), 061907.

[14] Angelov, S., P. Grefen, and D. Greefhorst (2012). A framework for analysis and design of software reference architectures. *Information and Software Technology 54*(4).

[15] Asimakopoulos, S., G. Asimakopoulos, and F. Spillers (2017). Motivation and user engagement in fitness tracking: Heuristics for mobile healthcare wearables. In *Informatics*, Volume 4, pp. 5. Multidisciplinary Digital Publishing Institute.

[16] Aspinall, M. G. and R. G. Hamermesh (2007). Realizing the promise of personalized medicine. *Harvard business review 85*(10), 108.

[17] Atrash, A. and J. Pineau (2008). A bayesian reinforcement learning approach for customizing human-robot interfaces. *Proceedingsc of the 13th international conference on Intelligent user interfaces - IUI '09*.

[18] Attfield, S., G. Kazai, M. Lalmas, and B. Piwowarski (2011). owards a science of user engagement (position paper). In *WSDM Workshop on User Modelling for Web Applications*.

[19] Auer, P., N. Cesa-Bianchi, and P. Fischer (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning 47*(2-3), 235–256.

[20] Ávila-Sansores, S., F. Orihuela-Espina, and L. Enrique-Sucar (2013). Patient tailored virtual rehabilitation. In *Converging Clinical and Engineering Research on Neurorehabilitation*, pp. 879–883. Springer.

[21] Awad, N. F. and M. S. Krishnan (2006). The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS quarterly*, 13–28.

[22] Bagdure, N. and B. Ambudkar (2015). Reducing delay during vertical handover. *2015 International Conference on Computing Communication Control and Automation*.

[23] Ballesteros, J., I. Ayala, J. R. Caro-Romero, M. Amor, and L. Fuentes (2020). Evolving dynamic self-adaptation policies of mhealth systems for long-term monitoring. *Journal of Biomedical Informatics 108*, 103494.

[24] Baniya, A., S. Herrmann, Q. Qiao, and H. Lu (2017). Adaptive interventions treatment modelling and regimen optimization using sequential multiple assignment randomized trials (smart) and q-learning. In *IIE Annual Conference. Proceedings*, pp. 1187–1192. Institute of Industrial and Systems Engineers (IISE).

[25] Baresi, L., W. G. Griswold, G. A. Lewis, M. Autili, I. Malavolta, and C. Julien (2020). Trends and challenges for software engineering in the mobile domain. *IEEE Software 38*(1), 88–96.

[26] Barto, A., P. Thomas, and R. Sutton (2017). Some recent applications of reinforcement learning.

[27] Basili, V. R., G. Caldiera, and H. D. Rombach (1994). The goal question metric approach. In *Encyclopedia of Software Engineering*. Wiley.

[28] Bassi, A., M. Bauer, M. Fiedler, T. Kramp, R. van Kranenburg, S. Lange, and S. Meissner (2016). *Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model* (1st ed.). Springer Publishing Company.

[29] Bates, D. W., S. Saria, L. Ohno-Machado, A. Shah, and G. Escobar (2014). Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs 33*(7), 1123–1131.

[30] Bauer, M. and et. al (2013). IoT Reference Architecture. In *Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model*.

[31] Bazzan, A. L. C. (2017). Synergies between evolutionary computation and multiagent reinforcement learning. *Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '17*.

[32] Bellemare, M. G., Y. Naddaf, J. Veness, and M. Bowling (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research 47*, 253–279.

[33] Bellera, C. A., G. MacGrogan, M. Debled, C. T. de Lara, V. Brouste, and S. Mathoulin-Pélissier (2010). Variables with time-varying effects and the Cox model: Some statistical concepts illustrated with a prognostic factor study in breast cancer. *BMC Medical Research Methodology 10*(1), 1–12.

[34] Bellman, R. E. (2015). *Adaptive control processes: a guided tour*, Volume 2045. Princeton university press.

[35] Berndt, D. J. and J. Clifford (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, Volume 10, pp. 359–370. Seattle, WA.

[36] Bi, H., O. J. Akinwande, and E. Gelenbe (2015). Emergency navigation in confined spaces using dynamic grouping. *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*.

[37] Biegel, G. and V. Cahill (2004). A framework for developing mobile, context-aware applications. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pp. 361–365. IEEE.

[38] Block, R. (2007). *Live from Macworld 2007: Steve Jobs keynote.* Available at https://www.engadget.com/2007/01/09/live-from-macworld-2007-steve-jobs-keynote/.

[39] Bodas, A., B. Upadhyay, C. Nadiger, and S. Abdelhak (2018). Reinforcement learning for game personalization on edge devices. *2018 International Conference on Information and Computer Technologies (ICICT)*.

[40] Bonino, D., M. T. D. Alizo, C. Pastrone, and M. Spirito (2016). Wasteapp: Smarter waste recycling for smart citizens. In *2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pp. 1–6. IEEE.

[41] Bouneffouf, D., A. Bouzeghoub, and A. L. Gançarski (2012). Hybrid-ϵ-greedy for mobile context-aware recommender system. *Lecture Notes in Computer Science*, 468–479.

[42] Bragg, J., Mausam, and D. S. Weld (2016). Optimal testing for crowd workers. In *AAMAS*.

[43] Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

[44] Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

[45] Brooke, J. (1996). Sus: a "quick and dirty'usability. *Usability evaluation in industry 189*.

[46] Brusilovski, P., A. Kobsa, and W. Nejdl (2007). *The adaptive web: methods and strategies of web personalization*, Volume 4321. Springer Science & Business Media.

[47] Bucchiarone, A., A. Lluch-Lafuente, A. Marconi, and M. Pistore (2009). A formalisation of adaptable pervasive flows. In *WS-FM*, pp. 61–75.

[48] Budgen, D. and P. Brereton (2006). Performing systematic literature reviews in software engineering. In *Proceedings of the 28th international conference on Software engineering*, pp. 1051–1052. ACM.

[49] Buduru, A. B. and S. S. Yau (2015). An effective approach to continuous user authentication for touch screen smart devices. *2015 IEEE International Conference on Software Quality, Reliability and Security*.

[50] Buehlmann, P. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics 2*(34), 559–583.

[51] Buehlmann, P. and T. Hothorn (2007). Boosting algorithms: regularization, prediction and model fitting. *Statistical Science 4*(22), 477–505.

[52] Buehlmann, P. and B. Yu (2003). Boosting with the l2 loss: regression and classification. *Journal of the American Statistical Association 1*(98), 324–339.

[53] Cafazzo, J. A., M. Casselman, N. Hamming, D. K. Katzman, and M. R. Palmert (2012, may). Design of an mHealth app for the self-management of adolescent type 1 diabetes: A pilot study. *J Med Internet Res 14*(3), e70.

[54] Caldiera, G., V. R. Basili, and H. D. Rombach (1994). Goal question metric paradigm. *Encyclopedia of software engineering 1*, 528–532.

[55] Calinescu, R., D. Weyns, S. Gerasimou, M. U. Iftikhar, I. Habli, and T. Kelly (2018). Engineering trustworthy self-adaptive software with dynamic assurance cases. *IEEE Trans. Software Eng. 44*(11), 1039–1069.

[56] Cañete, A., J.-M. Horcas, I. Ayala, and L. Fuentes (2020). Energy efficient adaptation engines for android applications. *Information and Software Technology 118*, 106220.

[57] Cao, F., M. Estert, W. Qian, and A. Zhou (2006). Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, pp. 328–339. SIAM.

[58] Casanueva, I., T. Hain, H. Christensen, R. Marxer, and P. Green (2015). Knowledge transfer between speakers for personalised dialogue management. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 12–21.

[59] Casquina, J. C., J. D. S. Eleuterio, and C. M. Rubira (2016). Adaptive deployment infrastructure for android applications. In *Dependable Computing Conference (EDCC), 2016 12th European*, pp. 218–228. IEEE.

[60] Castro-Gonzalez, A., F. Amirabdollahian, D. Polani, M. Malfaz, and M. A. Salichs (2011). Robot self-preservation and adaptation to user preferences in game play, a preliminary study. *2011 IEEE International Conference on Robotics and Biomimetics*.

[61] Cella, L. (2017). Modelling user behaviors with evolving users and catalogs of evolving items. *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17*.

[62] Chakraborty, B. and S. A. Murphy (2014). Dynamic treatment regimes. *Annual Review of Statistics and Its Application 1*(1), 447–464.

[63] Chan, J. and G. Nejat (2011). A learning-based control architecture for an assistive robot providing social engagement during cognitively stimulating activities. *2011 IEEE International Conference on Robotics and Automation*.

[64] Chatzis, S. P., V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications 112*, 353 – 371.

[65] Chellappa, R. K. and R. G. Sin (2005). Personalization versus privacy: An empirical examination of the online consumer's dilemma. *Information technology and management 6*(2-3), 181–202.

[66] Chen, J. and Z. Yang (2003). A learning multi-agent system for personalized information fiftering. *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*.

[67] Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM.

[68] Chen, T. and T. He (2015). Higgs boson discovery with boosted trees. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pp. 69–80.

[69] Chen, X., Y. Zhai, C. Lu, J. Gong, and G. Wang (2017). A learning model for personalized adaptive cruise control. *2017 IEEE Intelligent Vehicles Symposium (IV)*.

[70] Chen, Y. and L. Tu (2007). Density-based clustering for real-time stream data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142. ACM.

[71] Cheng, B. H. C., R. d. Lemos, H. Giese, P. Inverardi, and J. Magee (Eds.) (2009). *Software Engineering for Self-Adaptive Systems*. Lecture Notes in Computer Science.

[72] Cheng, Z., Q. Zhao, F. Wang, Y. Jiang, L. Xia, and J. Ding (2016). Satisfaction based q-learning for integrated lighting and blind control. *Energy and Buildings 127*, 43–55.

[73] Chi, C.-Y., R. T.-H. Tsai, J.-Y. Lai, and J. Y.-j. Hsu (2010). A reinforcement learning approach to emotion-based automatic playlist generation. *2010 International Conference on Technologies and Applications of Artificial Intelligence*.

[74] Chi, M., K. VanLehn, D. Litman, and P. Jordan (2010). Inducing effective pedagogical strategies using learning context features. *Lecture Notes in Computer Science*, 147–158.

[75] Chiang, Y.-S., T.-S. Chu, C. D. Lim, T.-Y. Wu, S.-H. Tseng, and L.-C. Fu (2014). Personalizing robot behavior for interruption in social human-robot interaction. *2014 IEEE International Workshop on Advanced Robotics and its Social Impacts*.

[76] Cho, Y. H., J. K. Kim, and S. H. Kim (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert systems with Applications 23*(3), 329–342.

[77] Chu, W., L. Li, L. Reyzin, and R. Schapire (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214.

[78] Claeys, M., S. Latre, J. Famaey, and F. De Turck (2014). Design and evaluation of a self-learning http adaptive video streaming client. *IEEE Communications Letters 18*(4), 716–719.

[79] Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological bulletin 114*(3), 494.

[80] Cooray, D., E. Kouroshfar, S. Malek, and R. Roshandel (2013). Proactive self-adaptation for improving the reliability of mission-critical, embedded, and mobile software. *IEEE Transactions on Software Engineering 39*(12), 1714–1735.

[81] Cortellese, F., M. Nalin, A. Morandi, A. Sanna, and F. Grasso (2009). Personality diagnosis for personalized ehealth services. In *International Conference on Electronic Healthcare*, pp. 157–164. Springer.

[82] Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological) 34*(2), 187–220.

[83] Cox, D. R. and D. Oakes (1984). *Analysis of survival data* (1st ed.). Chapman and Hall.

[84] Cugola, G., C. Ghezzi, L. S. Pinto, and G. Tamburrelli (2012). Adaptive service-oriented mobile applications: A declarative approach. In *International Conference on Service-Oriented Computing*, pp. 607–614. Springer.

[85] Da Silveira, G., D. Borenstein, and F. S. Fogliatto (2001). Mass customization: Literature review and research directions. *International journal of production economics 72*(1), 1–13.

[86] Dal Lago, U., M. Pistore, and P. Traverso (2002). Planning with a language for extended goals. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*, pp. 447–454.

[87] Daltayanni, M., C. Wang, and R. Akella (2012). A fast interactive search system for healthcare services. *2012 Annual SRII Global Conference*.

[88] Daskalaki, E., P. Diem, and S. G. Mougiakakou (2013a). An actor–critic based controller for glucose regulation in type 1 diabetes. *Computer Methods and Programs in Biomedicine 109*(2), 116–125.

[89] Daskalaki, E., P. Diem, and S. G. Mougiakakou (2013b). Personalized tuning of a reinforcement learning control algorithm for glucose regulation. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).*

[90] Daskalaki, E., P. Diem, and S. G. Mougiakakou (2016). Model-free machine learning in biomedicine: Feasibility study in type 1 diabetes. *PLOS ONE 11*(7), e0158722.

[91] Day, W. H. E. and H. Edelsbrunner (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification 1*(1), 7–24.

[92] de Morais Barroca Filho, I., G. S. A. Junior, and T. V. Batista (2019). Extending and instantiating a software reference architecture for iot-based healthcare applications. In *Int. Conf. on Computational Science and Its Applications*, pp. 203–218.

[93] De Paula, M., G. G. Acosta, and E. C. Martínez (2015). On-line policy learning and adaptation for real-time personalization of an artificial pancreas. *Expert Systems with Applications 42*(4), 2234–2255.

[94] De Paula, M., L. O. Ávila, and E. C. Martínez (2015). Controlling blood glucose variability under uncertainty using reinforcement learning and gaussian processes. *Applied Soft Computing 35*, 310–332.

[95] De Pessemier, T., S. Dooms, and L. Martens (2014). Context-aware recommendations through context and activity recognition in a mobile environment. *Multimedia Tools and Applications 72*(3), 2925–2948.

[96] den Hengst, F., E. Grua, A. el Hassouni, and M. Hoogendoorn (2020, January). Release of the systematic literature review into Reinforcement Learning for personalization.

[97] den Hengst, F., M. Hoogendoorn, F. van Harmelen, and J. Bosman (2019). Reinforcement learning for personalized dialogue management. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 59–67.

[98] Deng, K., J. Pineau, and S. Murphy (2011). Active learning for personalizing treatment. *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*.

[99] Deshmukh, A. A., Ürün Dogan, and C. Scott (2017). Multi-task learning for contextual bandits. In *NIPS*.

[100] Dey, S., Y. Kumar, S. Saha, and S. Basak (2016, 10). Forecasting to classification: Predicting the direction of stock market price using xtreme gradient boosting.

[101] Di Nucci, D., F. Palomba, A. Prota, A. Panichella, A. Zaidman, and A. De Lucia (2017). Software-based energy profiling of android apps: Simple, efficient and reliable? In *2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER)*, pp. 103–114. IEEE.

[102] Domingos, P. (2012, oct). A few useful things to know about machine learning. *Communications of the ACM 55*(10), 78.

[103] Duan, Y., X. Chen, R. Houthooft, J. Schulman, and P. Abbeel (2016). Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pp. 1329–1338.

[104] Durand, A. and J. Pineau (2015). Adaptive treatment allocation using sub-sampled gaussian processes. In *2015 AAAI Fall Symposium Series*.

[105] El Fouki, M., N. Aknin, and K. E. El. Kadiri (2017). Intelligent adapted e-learning system based on deep reinforcement learning. *Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems - ICCWCS'17*.

[106] El Hassouni, A., M. Hoogendoorn, A. E. Eiben, M. van Otterlo, and V. Muhonen (2019). End-to-end personalization of digital health interventions using raw sensor data with deep reinforcement learning: A comparative study in digital health interventions for behavior change. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 258–264. IEEE.

[107] el Hassouni, A., M. Hoogendoorn, M. van Otterlo, and E. Barbaro (2018a). Personalization of health interventions using cluster-based reinforcement learning. In *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 467–475. Springer.

[108] el Hassouni, A., M. Hoogendoorn, M. van Otterlo, and E. Barbaro (2018b). Personalization of health interventions using cluster-based reinforcement learning. *arXiv preprint arXiv:1804.03592*.

[109] Fan, H. and M. S. Poole (2006). What is personalization? perspectives on the design and implementation of personalization in information systems. *Journal of Organizational Computing and Electronic Commerce 16*(3-4), 179–202.

[110] Fawcett, T. (2006, jun). An introduction to ROC analysis. *Pattern Recognition Letters 27*(8), 861–874.

[111] Feljan, A. V., S. K. Mohalik, M. B. Jayaraman, and R. Badrinath (2015). SOA-PE: A service-oriented architecture for planning and execution in cyber-physical systems. In *2015 International Conference on Smart Sensors and Systems (IC-SSS)*, pp. 1–6.

[112] Feng, J., H. Li, M. Huang, S. Liu, W. Ou, Z. Wang, and X. Zhu (2018). Learning to collaborate. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*.

[113] Fernandez-Gauna, B. and M. Grana (2014). Recipe tuning by reinforcement learning in the sands ecosystem. *2014 6th International Conference on Computational Aspects of Social Networks*.

[114] Ferretti, S., S. Mirri, C. Prandi, and P. Salomoni (2014a). Exploiting reinforcement learning to profile users and personalize web pages. *2014 IEEE 38th International Computer Software and Applications Conference Workshops*.

[115] Ferretti, S., S. Mirri, C. Prandi, and P. Salomoni (2014b). User centered and context dependent personalization through experiential transcoding. *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*.

[116] Ferretti, S., S. Mirri, C. Prandi, and P. Salomoni (2016a). Automatic web content personalization through reinforcement learning. *Journal of Systems and Software 121*, 157–169.

[117] Ferretti, S., S. Mirri, C. Prandi, and P. Salomoni (2016b). On personalizing web content through reinforcement learning. *Universal Access in the Information Society 16*(2), 395–410.

[118] Ferretti, S., S. Mirri, C. Prandi, and P. Salomoni (2017). On personalizing web content through reinforcement learning. *Universal Access in the Information Society 16*(2), 395–410.

[119] Fling, B. (2009). *Mobile design and development: Practical concepts and techniques for creating mobile sites and Web apps.* O'Reilly Media, Inc.

[120] Flores, H. and S. Srirama (2013). Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning. In *Proceeding of the fourth ACM workshop on Mobile cloud computing and services*, pp. 9–16. ACM.

[121] Fournier, L. (1994). Learning capabilities for improving automatic transmission control. *Proceedings of the Intelligent Vehicles '94 Symposium.*

[122] Fremantle, P. (2015). A Reference Architecture for the Internet of Things. WSO2 White paper.

[123] Gao, A. Y., W. Barendregt, and G. Castellano (2017). Personalised human-robot co-adaptation in instructional settings using reinforcement learning. In *IVA Workshop on Persuasive Embodied Agents for Behavior Change: PEACH 2017, August 27, Stockholm, Sweden.*

[124] Garcıa, J. and F. Fernández (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research 16*(1), 1437–1480.

[125] Garivier, A. and E. Moulines (2011). On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pp. 174–188. Springer.

[126] Gaweda, A., M. Muezzinoglu, G. Aronoff, A. Jacobs, J. Zurada, and M. Brier (2005a). Incorporating prior knowledge into q-learning for drug

delivery individualization. *Fourth International Conference on Machine Learning and Applications (ICMLA'05).*

[127] Gaweda, A. E. (2009). Improving management of anemia in end stage renal disease using reinforcement learning. *2009 International Joint Conference on Neural Networks*.

[128] Gaweda, A. E., M. K. Muezzinoglu, G. R. Aronoff, A. A. Jacobs, J. M. Zurada, and M. E. Brier (2005b). Individualization of pharmacological anemia management using reinforcement learning. *Neural Networks 18*(5), 826–834.

[129] Gentile, C., S. Li, and G. Zappella (2014). Online clustering of bandits. In *International Conference on Machine Learning*, pp. 757–765.

[130] Ghahfarokhi, B. S. and N. Movahhedinia (2013). A personalized qoe-aware handover decision based on distributed reinforcement learning. *Wireless Networks 19*(8), 1807–1828.

[131] Gil, M., V. Pelechano, J. Fons, and M. Albert (2016). Designing the human in the loop of self-adaptive systems. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pp. 437–449. Springer.

[132] Ginsburg, G. S. and J. J. McCarthy (2001). Personalized medicine: revolutionizing drug discovery and patient care. *TRENDS in Biotechnology 19*(12), 491–496.

[133] Glavinic, V., S. Ljubic, and M. Kukec (2008). Transformable menu component for mobile device applications: Working with both adaptive and adaptable user interfaces. *International Journal of Interactive Mobile Technologies (iJIM) 2*(3), 22–27.

[134] Global Industry Analysts, I. (2019). mhealth (mobile health) services - market analysis, trends, and forecasts.

[135] Glowacka, D., T. Ruotsalo, K. Konuyshkova, k. Athukorala, S. Kaski, and G. Jacucci (2013). Directing exploratory search. *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13*.

[136] Goldberg, Y. and M. R. Kosorok (2012). Q-learning with censored data. *The Annals of Statistics 40*(1), 529–560.

[137] Gordon, G., S. Spaulding, J. K. Westlund, J. J. Lee, L. Plummer, M. Martinez, M. Das, and C. Breazeal (2016). Affective personalization of a social robot tutor for children's second language skills. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[138] Goutte, C. and E. Gaussier (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In D. E. Losada and J. M. Fernández-Luna (Eds.), *Advances in Information Retrieval*, Berlin, Heidelberg, pp. 345–359. Springer Berlin Heidelberg.

[139] Gower, J. C. (1971, dec). A general coefficient of similarity and some of its properties. *Biometrics 27*(4), 857.

[140] Grassi, V. and R. Mirandola (2021). The tao way to anti-fragile software architectures: the case of mobile applications. In *IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE.

[141] Greenhalgh, T. and R. Peacock (2005). Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources. *Bmj 331*(7524), 1064–1065.

[142] Grua, E., I. Malavolta, and P. Lago (2019a). *Replication package of the study*. Available at http://s2group.cs.vu.nl/seams-2019-replication-package/.

[143] Grua, E. M., M. De Sanctis, and P. Lago (2020). A reference architecture for personalized and self-adaptive e-health apps. In *Software Architecture: 14th European Conference, ECSA 2020 Tracks and Workshops, L'Aquila, Italy, September 14–18, 2020, Proceedings*, pp. 195–209. Springer.

[144] Grua, E. M., M. De Sanctis, I. Malavolta, M. Hoogendoorn, and P. Lago (2021). Social sustainability in the e-health domain via personalized and self-adaptive mobile apps. In C. Calero, M. Moraga, and M. Piattini (Eds.), *Software Sustainability*. Springer. To appear.

[145] Grua, E. M. and M. Hoogendoorn (2018). Exploring clustering techniques for effective reinforcement learning based personalization for health and wellbeing. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 813–820. IEEE.

[146] Grua, E. M., M. Hoogendoorn, I. Malavolta, P. Lago, and A. Eiben (2019). Clustream-GT: Online clustering for personalization in the health domain. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 270–275. ACM.

[147] Grua, E. M., I. Malavolta, and P. Lago (2019b). Self-adaptation in mobile apps: A systematic literature study. In *IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 51–62.

[148] Grua, E. M., I. Malavolta, and P. Lago (2019). Self-adaptation in mobile apps: a systematic literature study. In *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 51–62.

[149] Harper, F. M., X. Li, Y. Chen, and J. A. Konstan (2005). An economic model of user rating in an online recommender system. *Lecture notes in computer science 3538*, 307.

[150] Heagerty, P. J., T. Lumley, and M. S. Pepe (2000). Time-dependent roc curves for censored survival data and a diagnostic marker. *Biometrics 56*(2), 337–344.

[151] Hemminghaus, J. and S. Kopp (2018). Adaptive behavior generation for child-robot interaction. *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction - HRI '18*.

[152] Henning, C. and T. Liao (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society 62*(3), 309–369.

[153] Herland, M., T. M. Khoshgoftaar, and R. Wald (2014). A review of data mining using big data in health informatics. *Journal of Big data 1*(1), 1–35.

[154] Hester, T. and P. Stone (2012). Learning and using models. In M. Wiering and M. Van Otterlo (Eds.), *Reinforcement learning*, Volume 12, pp. "120". Springer.

[155] Hill, D. N., H. Nassif, Y. Liu, A. Iyer, and S. Vishwanathan (2017). An efficient bandit algorithm for realtime multivariate optimization. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*.

[156] Hiraoka, T., G. Neubig, S. Sakti, T. Toda, and S. Nakamura (2016). Learning cooperative persuasive dialogue policies using framing. *Speech Communication 84*, 83–96.

[157] Hochberg, I., G. Feraru, M. Kozdoba, S. Mannor, M. Tennenholtz, and E. Yom-Tov (2016a). Encouraging physical activity in patients with diabetes through automatic personalized feedback via reinforcement learning improves glycemic control. *Diabetes Care 39*(4), e59–e60.

[158] Hochberg, I., G. Feraru, M. Kozdoba, S. Mannor, M. Tennenholtz, and E. Yom-Tov (2016b). Encouraging physical activity in patients with diabetes through automatic personalized feedback via reinforcement learning improves glycemic control. *Diabetes Care 39*(4), e59–e60.

[159] Hogben, G. and M. Perera (2018). Project capillary: End-to-end encryption for push messaging, simplified.

[160] Hoogendoorn, M. and B. Funk (2017). *Machine Learning for the Quantified Self: On the Art of Learning from Sensory Data*, Volume 35. Springer.

[161] Huajun, Z., Z. Jin, W. Rui, and M. Tan (2008). Multi-objective reinforcement learning algorithm and its application in drive system. *2008 34th Annual Conference of IEEE Industrial Electronics*.

[162] Huang, S.-l. and F.-r. Lin (2005). Designing intelligent sales-agent for online selling. *Proceedings of the 7th international conference on Electronic commerce - ICEC '05*.

[163] IBM (2006). An architectural blueprint for autonomic computing. Technical report, IBM.

[164] Ie, E., C.-w. Hsu, M. Mladenov, V. Jain, S. Narvekar, J. Wang, R. Wu, and C. Boutilier (2019). Recsim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*.

[165] International Organization for Standardization (2011). ISO/IEC/IEEE 42010:2011 - Systems and Software Engineering – Architecture Description. Technical report, International Organization for Standardization (ISO).

[166] internationale de normalisation, O. (2011). *Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE): System and Software Quality Models*. ISO/IEC.

[167] Jaradat, S., N. Dokoohaki, M. Matskin, and E. Ferrari (2016). Trust and privacy correlations in social networks: A deep learning framework. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

[168] Jawaheer, G., M. Szomszor, and P. Kostkova (2010). Comparison of implicit and explicit feedback from an online music recommendation service. In *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, pp. 47–51. ACM.

[169] Jiang, N. and L. Li (2016). Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 652–661.

[170] Jin, X. and J. Han (2016). K-medoids clustering. In *Encyclopedia of Machine Learning and Data Mining*, pp. 1–3. Springer.

[171] Jin, Z. and Z. Huajun (2011). Multi-objective reinforcement learning algorithm and its improved convergency method. *2011 6th IEEE Conference on Industrial Electronics and Applications*.

[172] Kardan, A. A. and O. R. Speily (2010). Smart lifelong learning system based on q-learning. *2010 Seventh International Conference on Information Technology: New Generations*.

[173] Kastanis, I. and M. Slater (2012). Reinforcement learning utilizes proxemics. *ACM Transactions on Applied Perception 9*(1), 1–15.

[174] Khalid, H., E. Shihab, M. Nagappan, and A. E. Hassan (2015). What do mobile app users complain about? *IEEE Software 32*(3), 70–77.

[175] Khribi, M. K., M. Jemni, and O. Nasraoui (2008). Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pp. 241–245. IEEE.

[176] Kim, H.-K. (2013). Architecture for adaptive mobile applications. *Int. J. Bio-Sci. Bio-Technol 5*(5), 197–210.

[177] Kim, K.-j. and H. Ahn (2004). Using a clustering genetic algorithm to support customer segmentation for personalized recommender systems. In *International Conference on AI, Simulation, and Planning in High Autonomy Systems*, pp. 409–415. Springer.

[178] Kim, Y. H., D. J. Kim, and K. Wachter (2013, dec). A study of mobile user engagement (MoEN): Engagement motivations, perceived value, satisfaction, and continued engagement intention. *Decision Support Systems 56*, 361–370.

[179] Kitchenham, B. and P. Brereton (2013). A systematic review of systematic review process research in software engineering. *Information and software technology 55*(12), 2049–2075.

[180] Kitchenham, B. A. and S. Charters (2007). *Guidelines for performing systematic literature reviews in software engineering.* Technical Report EBSE-2007-01, Keele Uni- versity and University of Durham.

[181] Kober, J. and J. Peters (2012). Reinforcement learning in robotics: A survey. In M. Wiering and M. Van Otterlo (Eds.), *Reinforcement learning*, Volume 12, pp. "596–597". Springer.

[182] Konda, V. R. and J. N. Tsitsiklis (2000). Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014.

[183] Koukoutsidis, I. (2003). A learning strategy for paging in mobile environments. *5th European Personal Mobile Communications Conference 2003*.

[184] Krakow, E. F., M. Hemmer, T. Wang, B. Logan, M. Arora, S. Spellman, D. Couriel, A. Alousi, J. Pidala, M. Last, and et al. (2017). Tools for the precision medicine era: How to develop highly personalized treatment recommendations from cohort and registry data using q-learning. *American Journal of Epidemiology 186*(2), 160–172.

[185] Krupitzer, C., F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker (2015). A survey on engineering approaches for self-adaptive systems. *Pervasive and Mobile Computing 17*, 184–206.

[186] Kuhn, T., K. Pittel, and T. Schulz (2003). Recycling for sustainability-a long run perspective? *International journal of global environmental issues 3*(3), 339–355.

[187] Lago, P. (2019). Architecture design decision maps for software sustainability. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pp. 61–64. IEEE.

[188] Lago, P., R. Verdecchia, N. C. Fernandez, E. Rahmadian, J. Sturm, T. van Nijnanten, R. Bosma, C. Debuysscher, and P. Ricardo (2020). Designing for sustainability: Lessons learned from four industrial projects. In *Environmental*

Informatics – Sustainability aware digital twins for urban smart environments (EnviroInfo). Springer.

[189] Lagoudakis, M. G. and R. Parr (2003). Least-squares policy iteration. *Journal of machine learning research 4*(Dec), 1107–1149.

[190] Lai, T. L. and H. Robbins (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics 6*(1), 4–22.

[191] Lan, A. S. and R. G. Baraniuk (2016). A contextual bandits framework for personalized learning action selection. In *EDM*.

[192] Laprie, J.-C. (2008). From dependability to resilience. In *38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks*, pp. G8–G9. Citeseer.

[193] Lee, G., S. Bauer, P. Faratin, and J. Wroclawski (2004). Learning user preferences for wireless services provisioning. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, 480–487.

[194] Lehmann, J., M. Lalmas, E. Yom-Tov, and G. Dupret (2012). Models of user engagement. In *Proceedings of the Conference on User Modeling, Adaptation, and Personalization*, UMAP, pp. 164–175. Springer.

[195] Lella, A. and A. Lipsman (2016). The 2016 U.S. Mobile App Report. comsCore white paper.

[196] Lewis, G. A. and P. Lago (2015). Architectural tactics for cyber-foraging: Results of a systematic literature review. *J. Syst. Softw. 107*, 158–186.

[197] Li, K. and M. Q.-H. Meng (2015). Personalizing a service robot by learning human habits from behavioral footprints. *Engineering 1*(1), 079–084.

[198] Li, L., W. Chu, J. Langford, and R. E. Schapire (2010a). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM.

[199] Li, L., W. Chu, J. Langford, and R. E. Schapire (2010b). A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th international conference on World wide web - WWW '10*.

[200] Li, Z., J. Kiseleva, M. de Rijke, and A. Grotov (2017). Towards learning reward functions from user interactions. *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval - ICTIR '17*.

[201] Liaw, A. and M. Wiener (2002). Classification and regression by random-forest. *R News 2*(3), 18–22.

[202] Liebman, E. and P. Stone (2015). Dj-mc: A reinforcement-learning agent for music playlist recommendation. In *AAMAS*.

[203] Lim, J., H. Son, D. Lee, and D. Lee (2017). An marl-based distributed learning scheme for capturing user preferences in a smart environment. *2017 IEEE International Conference on Services Computing (SCC)*.

[204] Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning 8*(3-4), 293–321.

[205] Linden, A. and S. Mantyniemi (2011). Using the negative binomial distribution to model overdispersion in ecological count data. *Ecology 92*(7), 1414–1421.

[206] Liu, Q., B. Cui, Z. Wei, B. Peng, H. Huang, H. Deng, J. Hao, X. Huang, and K.-F. Wong (2019). Building personalized simulator for interactive search. In *Proceedings of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pp. 5109–5115.

[207] Liu, Y., B. Logan, N. Liu, Z. Xu, J. Tang, and Y. Wang (2017). Deep reinforcement learning for dynamic treatment regimes on medical registry data. *2017 IEEE International Conference on Healthcare Informatics (ICHI)*.

[208] Liu, Y. and Y. Zhuang (2015). Research model of churn prediction based on customer segmentation and misclassification cost in the context of big data. *JCC 03*(06), 87–93.

[209] Llorente and S. E. Guerrero (2012). Increasing retrieval quality in conversational recommenders. *IEEE Transactions on Knowledge and Data Engineering 24*(10), 1876–1888.

[210] Lopez, F. S. and N. Condori-Fernández (2016). Design of an adaptive persuasive mobile application for stimulating the medication adherence. In *International Conference on Intelligent Technologies for Interactive Entertainment*, pp. 99–105. Springer.

[211] Lopez, F. S. and N. Condori-Fernández (2016). Design of an adaptive persuasive mobile application for stimulating the medication adherence. In *Intelligent Technologies for Interactive Entertainment - 8th International Conference, INTETAIN 2016, Revised Selected Papers*, Volume 178, pp. 99–105. Springer.

[212] Lotfy, H. M., S. M. Khamis, and M. M. Aboghazalah (2016). Multiagents and learning: Implications for webusage mining. *Journal of Advanced Research 7*(2), 285–295.

[213] Lowery, C. and A. A. Faisal (2013). Towards efficient, personalized anesthesia using continuous reinforcement learning for propofol infusion control. *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER).*

[214] Ma, Y., X. Liu, Y. Liu, Y. Liu, and G. Huang (2018). A tale of two fashions: An empirical study on the performance of native apps and web apps on android. *IEEE Transactions on Mobile Computing 17*(5), 990–1003.

[215] Macías-Escrivá, F. D., R. Haber, R. Del Toro, and V. Hernandez (2013). Self-adaptive systems: A survey of current approaches, research challenges and applications. *Expert Systems with Applications 40*(18), 7267–7279.

[216] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Volume 1, pp. 281–297. Oakland, CA, USA.

[217] Madani, O. and D. DeCoste (2005). Contextual recommender problems [extended abstract]. *Proceedings of the 1st international workshop on Utility-based data mining - UBDM '05*.

[218] Maes, P. and R. Kozierok (1993). Learning interface agents. In *AAAI*, Volume 93, pp. 459–465.

[219] Mahdavi-Hezavehi, S., V. H. Durelli, D. Weyns, and P. Avgeriou (2017). A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems. *Information and Software Technology 90*, 1–26.

[220] Mahmood, T., G. Mujtaba, and A. Venturini (2013). Dynamic personalization in conversational recommender systems. *Information Systems and e-Business Management 12*(2), 213–238.

[221] Mahmood, T. and F. Ricci (2007). Learning and adaptivity in interactive recommender systems. *Proceedings of the ninth international conference on Electronic commerce - ICEC '07*.

[222] Malavolta, I. (2016). Beyond native apps: web technologies to the rescue!(keynote). In *Proceedings of the 1st International Workshop on Mobile Development*, pp. 1–2. ACM.

[223] Malavolta, I., E. M. Grua, C.-Y. Lam, R. de Vries, F. Tan, E. Zielinski, M. Peters, and L. Kaandorp (2020). A framework for the automatic execution of measurement-based experiments on android devices. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering Workshops*, ASE '20, pp. 61–66.

[224] Malavolta, I., S. Ruberto, V. Terragni, and T. Soru (2015, June). End users' perception of hybrid mobile apps in the google play store. In *Mobile Services (MS), 2015 IEEE International Conference on*, pp. 25–32. Institute of Electrical and Electronics Engineers (IEEE).

[225] Malpani, A., B. Ravindran, and H. Murthy (2011). Personalized intelligent tutoring system using reinforcement learning. In *FLAIRS Conference*.

[226] Manickam, I., A. S. Lan, and R. G. Baraniuk (2017). Contextual multi-armed bandit algorithms for personalized learning action selection. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[227] Martin, K. N. and I. Arroyo (2004). Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*, pp. 564–572. Springer.

[228] Martín-Guerrero, J. D., F. Gomez, E. Soria-Olivas, J. Schmidhuber, M. Climente-Martí, and N. V. Jiménez-Torres (2009). A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients. *Expert Systems with Applications 36*(6), 9737–9742.

[229] Martín-Guerrero, J. D., F. Gomez, E. Soria-Olivas, J. Schmidhuber, M. Climente-Martí, and N. V. Jiménez-Torres (2009). A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients. *Expert Systems with Applications 36*(6), 9737–9742.

[230] Martín-Guerrero, J. D., E. Soria-Olivas, M. Martínez-Sober, A. J. Serrrano-López, R. Magdalena-Benedito, and J. Gómez-Sanchis (2008). Use of reinforcement learning in two real applications. *Recent Advances in Reinforcement Learning*, 191–204.

[231] Massimo, D., M. Elahi, and F. Ricci (2017). Learning user preferences by observing user-items interactions in an iot augmented space. *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17.*

[232] Masumitsu, K. and T. Echigo (2000). Video summarization using reinforcement learning in eigenspace. *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101).*

[233] May, B. C., N. Korda, A. Lee, and D. S. Leslie (2012). Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research 13*(Jun), 2069–2106.

[234] McKnight, P. E. and J. Najab (2010). Mann-whitney u test. *The Corsini encyclopedia of psychology*, 1–1.

[235] Mengelkamp, E., J. Gärttner, and C. Weinhardt (2018). Intelligent agent strategies for residential customers in local electricity markets. *Proceedings of the Ninth International Conference on Future Energy Systems - e-Energy '18*.

[236] Mengelkamp, E. and C. Weinhardt (2018). Clustering household preferences in local electricity markets. *Proceedings of the Ninth International Conference on Future Energy Systems - e-Energy '18*.

[237] Menikpura, S., S. H. Gheewala, S. Bonnet, and C. Chiemchaisri (2013). Evaluation of the effect of recycling on sustainability of municipal solid waste management in thailand. *Waste and Biomass Valorization 4*(2), 237–257.

[238] Merkle, N. and S. Zander (2017). Agent-based assistance in ambient assisted living through reinforcement learning and semantic technologies. *Lecture Notes in Computer Science*, 180–188.

[239] Middelweerd, A., J. S. Mollee, C. N. van der Wal, J. Brug, and S. J. Te Velde (2014). Apps to promote physical activity among adults: a review and content analysis. *International journal of behavioral nutrition and physical activity 11*(1), 97.

[240] Mizouni, R., M. A. Matar, Z. Al Mahmoud, S. Alzahmi, and A. Salah (2014). A framework for context-aware self-adaptive mobile applications spl. *Expert Systems with applications 41*(16), 7549–7564.

[241] Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[242] Mo, K., Y. Zhang, S. Li, J. Li, and Q. Yang (2018). Personalizing a dialogue system with transfer reinforcement learning. In *AAAI*.

[243] Moghaddam, F. A., M. Simaremare, P. Lago, and P. Grosso (2017). A self-adaptive framework for enhancing energy efficiency in mobile applications. In *2017 Sustainable Internet and ICT for Sustainability (SustainIT)*, pp. 1–3. IEEE.

[244] Moghimi, M., J. Venkatesh, P. Zappi, and T. Rosing (2013). Context-aware mobile power management using fuzzy inference as a service. In D. Uhler, K. Mehta, and J. L. Wong (Eds.), *Mobile Computing, Applications, and Services*, Berlin, Heidelberg, pp. 314–327. Springer Berlin Heidelberg.

[245] Mohalik, S. K., N. C. Narendra, R. Badrinath, and D. Le (2017). Adaptive service-oriented architectures for cyber physical systems. In *IEEE Symposium on Service-Oriented System Engineering, SOSE*, pp. 57–62.

[246] Moher, D., A. Liberati, J. Tetzlaff, and D. G. Altman (2009). Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *Annals of internal medicine 151*(4), 264–269.

[247] Moling, O., L. Baltrunas, and F. Ricci (2012). Optimal radio channel recommendations with explicit and implicit feedback. *Proceedings of the sixth ACM conference on Recommender systems - RecSys '12*.

[248] Moon, A., T. Kang, H. Kim, and H. Kim (2007). A service recommendation using reinforcement learning for network-based robots in ubiquitous computing environments. *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*.

[249] Morandini, M., L. Penserini, and A. Perini (2008). Towards goal-oriented development of self-adaptive systems. In *2008 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, SEAMS*, pp. 9–16.

[250] Muccini, H., M. Sharaf, and D. Weyns (2016). Self-adaptation for cyber-physical systems: a systematic literature review. In *Proceedings of the 11th international symposium on software engineering for adaptive and self-managing systems*, pp. 75–81. ACM.

[251] Muccini, H., R. Spalazzese, M. T. Moghaddam, and M. Sharaf (2018). Self-adaptive iot architectures: an emergency handling case study. In *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, pp. 19. ACM.

[252] Müller, H., J. Gove, and J. Webb (2012). Understanding tablet use. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI*. ACM Press.

[253] Müller, H., A. Hanbury, N. Al Shorbaji, et al. (2012). Health information search to deal with the exploding amount of health information produced. *Methods of information in medicine 51*(6), 516.

[254] Mylopoulos, J., L. Chung, and B. A. Nixon (1992). Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Trans. Software Eng. 18*(6), 483–497.

[255] Naqvi, N. Z., J. Devlieghere, D. Preuveneers, and Y. Berbers (2016). Mascot: self-adaptive opportunistic offloading for cloud-enabled smart mobile applications with probabilistic graphical models at runtime. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pp. 5701–5710. IEEE.

[256] Narvekar, S., J. Sinapov, and P. Stone (2017). Autonomous task sequencing for customized curriculum design in reinforcement learning. In *IJCAI*.

[257] Nelissen, K., M. Snoeck, S. V. Broucke, and B. Baesens (2018, June). Swipe and tell: Using implicit feedback to predict user engagement on tablets. *ACM Trans. Inf. Syst. 36*(4), 35:1–35:36.

[258] Nemati, S., M. M. Ghassemi, and G. D. Clifford (2016). Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.

[259] Neumann, D., T. Mansi, L. Itu, B. Georgescu, E. Kayvanpour, F. Sedaghat-Hamedani, A. Amr, J. Haas, H. Katus, B. Meder, and et al. (2016). A self-taught artificial agent for multi-physics computational model personalization. *Medical Image Analysis 34*, 52–64.

[260] Neumann, D., T. Mansi, L. Itu, B. Georgescu, E. Kayvanpour, F. Sedaghat-Hamedani, J. Haas, H. Katus, B. Meder, S. Steidl, and et al. (2015). Vito – a generic agent for multi-physics model personalization: Application to heart modeling. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 442–449.

[261] Nocedal, J. and S. J. Wright (1999). *Numerical Optimization* (2nd ed.). Springer.

[262] O'Brien, H. and R. Bassett (2009). Exploring engagement in the qualitative research process. American Society for Information Science and Technology Annual Meeting, Vancouver, BC, October, 2009.

[263] O'Brien, H. L. and E. G. Toms (2008). What is user engagement? a conceptual framework for defining user engagement with technology. *J. Am. Soc. Inf. Sci. 59*(6), 938–955.

[264] Oh, D. and C. L. Tan (2004). Making better recommendations with online profiling agents. *AI Magazine 26*, 29–40.

[265] Ondruska, P. and I. Posner (2014). The route not taken: Driver-centric estimation of electric vehicle range. In *Twenty-Fourth International Conference on Automated Planning and Scheduling.*

[266] Palomba, F., M. L. Vásquez, G. Bavota, R. Oliveto, M. Di Penta, D. Poshyvanyk, and A. De Lucia (2018). Crowdsourcing user reviews to support the evolution of mobile apps. *Journal of Systems and Software 137*, 143–162.

[267] Pan, S. J. and Q. Yang (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering 22*(10), 1345–1359.

[268] Pant, V., S. Bhasin, and S. Jain (2017). Self-learning system for personalized e-learning. *2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT).*

[269] Paschou, M., E. Sakkopoulos, E. Sourla, and A. Tsakalidis (2013). Health internet of things: Metrics and methods for efficient data transfer. *Simulation Modelling Practice and Theory 34*, 186 – 199.

[270] Patompak, P., S. Jeong, I. Nilkhamhang, and N. Y. Chong (2017). Learning social relations for culture aware interaction. *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI).*

[271] Pecka, M. and T. Svoboda (2014). Safe exploration techniques for reinforcement learning–an overview. In *International Workshop on Modelling and Simulation for Autonomous Systems*, pp. 357–375. Springer.

[272] Peng, B., Q. Jiao, and T. Kurner (2016). Angle of arrival estimation in dynamic indoor thz channels with bayesian filter and reinforcement learning. *2016 24th European Signal Processing Conference (EUSIPCO).*

[273] Peng, C. and P. Vuorimaa (2004). Automatic navigation among mobile dtv services. In *ICEIS*.

[274] Perera, C., A. Zaslavsky, P. Christen, and D. Georgakopoulos (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials 16*(1), 414–454.

[275] Petersen, K., R. Feldt, S. Mujtaba, and M. Mattsson (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, Swinton, UK, UK, pp. 68–77. British Computer Society.

[276] Petersen, K., S. Vakkalanka, and L. Kuzniarz (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology 64*, 1–18.

[277] Picco, G. P., C. Julien, A. L. Murphy, M. Musolesi, and G.-C. Roman (2014). Software engineering for mobility: reflecting on the past, peering into the future. In *Proceedings of the on Future of Software Engineering*, pp. 13–28. ACM.

[278] Pine, B. J., B. Victor, and A. C. Boynton (1993). Making mass customization work. *Harvard business review 71*(5), 108–11.

[279] Pineau, J., M. G. Bellemare, A. J. Rush, A. Ghizaru, and S. A. Murphy (2007). Constructing evidence-based treatment strategies using methods from computer science. *Drug and Alcohol Dependence 88*, S52–S60.

[280] Pomprapa, A., S. Leonhardt, and B. J. Misgeld (2017). Optimal learning control of oxygen saturation using a policy iteration algorithm and a proof-of-concept in an interconnecting three-tank system. *Control Engineering Practice 59*, 194–203.

[281] Powell, A. C., J. Torous, S. Chan, G. S. Raynor, E. Shwarts, M. Shanahan, and A. B. Landman (2016). Interrater reliability of mhealth app rating measures: analysis of top depression and smoking cessation apps. *JMIR mHealth and uHealth 4*(1).

[282] Prasad, N., L.-F. Cheng, C. Chivers, M. Draugelis, and B. E. Engelhardt (2017). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *CoRR abs/1704.06300*.

[283] Preda, M. and D. Popescu (2005). Personalized web recommendations: Supporting epistemic information about end-users. *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*.

[284] Preuveneers, D., N. Z. Naqvi, A. Ramakrishnan, Y. Berbers, and W. Joosen (2016). Adaptive dissemination for mobile electronic health record applications with proactive situational awareness. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 3229–3238. IEEE.

[285] Priscoli, F. D., L. Fogliati, A. Palo, and A. Pietrabissa (2014). Dynamic class of service mapping for quality of experience control in future networks. In *WTC 2014; World Telecommunications Congress 2014*, pp. 1–6. VDE.

[286] Qian, W., X. Peng, H. Wang, J. Mylopoulos, J. Zheng, and W. Zhao (2018). MobiGoal: Flexible achievement of personal goals for mobile users. *IEEE Trans. Services Computing 11*(2), 384–398.

[287] Qin, Z., I. Rishabh, and J. Carnahan (2016). A scalable approach for periodical personalized recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*.

[288] Racherla, P., C. Furner, and J. Babb (2012). Conceptualizing the implications of mobile app usage and stickiness: A research agenda. *Available at SSRN 2187056*.

[289] Raghuveer, V. R., B. K. Tripathy, T. Singh, and S. Khanna (2014). Reinforcement learning approach towards effective content recommendation in mooc environments. *2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)*.

[290] Raheel, S. (2016). Improving the user experience using an intelligent adaptive user interface in mobile applications. In *Multidisciplinary Conference on Engineering Technology (IMCET), IEEE International*, pp. 64–68. IEEE.

[291] Ramesh, A., D. Goldwasser, B. Huang, H. D. III, and L. Getoor (2014). Learning latent engagement patterns of students in online courses. In *AAAI Conference on Artificial Intelligence*.

[292] Rennison, E. (1995). Personalized galaxies of infomation. In *Companion of the ACM Conference on Human Factors in Computing Systems (CHI'95)*.

[293] Resnick, P. and H. R. Varian (1997). Recommender systems. *Communications of the ACM 40*(3), 56–58.

[294] Riaño, D., F. Real, J. A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, R. Annicchiarico, and C. Caltagirone (2012). An ontology-based

personalization of health-care knowledge to support clinical decisions for chronically ill patients. *Journal of biomedical informatics 45*(3), 429–446.

[295] Ricci, F., L. Rokach, and B. Shapira (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pp. 14–17. Springer.

[296] Riecken, D. (2000). Personalized views of personalization. *Communications of the ACM 43*(8), 26–26.

[297] Riedmiller, M. (2005). Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pp. 317–328. Springer.

[298] Ritschel, H. and E. André (2017). Real-time robot personality adaptation based on reinforcement learning and social signals. *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*.

[299] Rivas-Blanco, I., C. Lopez-Casado, C. J. Perez-del Pulgar, F. Garcia-Vacas, J. C. Fraile, and V. F. Munoz (2018). Smart cable-driven camera robotic assistant. *IEEE Transactions on Human-Machine Systems 48*(2), 183–196.

[300] Rodrigues, P. P., J. Gama, and J. Pedroso. Hierarchical timeseries clustering for data streams. [Online;].

[301] Rodrigues, P. P., J. Gama, and J. Pedroso (2008). Hierarchical clustering of time-series data streams. *IEEE transactions on knowledge and data engineering 20*(5), 615–627.

[302] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics 20*, 53–65.

[303] Rudary, M., S. Singh, and M. E. Pollack (2004). Adaptive cognitive orthotics. *Twenty-first international conference on Machine learning - ICML '04*.

[304] Russell, S. and P. Norvig (2002). Artificial intelligence: a modern approach.

[305] Saha, S. and R. Quazi (2008). Emotion-driven learning agent for setting rich presence in mobile telephony. *2008 11th International Conference on Computer and Information Technology*.

[306] Sama, M., D. S. Rosenblum, Z. Wang, and S. Elbaum (2010). Multi-layer faults in the architectures of mobile, context-aware adaptive applications. *Journal of Systems and Software 83*(6), 906–914.

[307] Santos, M., C. Gralha, M. Goulão, and J. Araújo (2018). Increasing the semantic transparency of the KAOS goal model concrete syntax. In *Conceptual Modeling - 37th International Conference, ER*, pp. 424–439.

[308] Saraçli, S., N. Doğan, and İ. Doğan (2013). Comparison of hierarchical cluster analysis methods by cophenetic correlation. *J Inequal Appl 2013*(1), 203.

[309] Schafer, J. B., D. Frankowski, J. Herlocker, and S. Sen (2007). Collaborative filtering recommender systems. In *The adaptive web*, pp. 291–324. Springer.

[310] Sekhavat, Y. A. (2017). Mprl: Multiple-periodic reinforcement learning for difficulty adjustment in rehabilitation games. *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)*.

[311] Sener, I. N., R. B. Copperman, R. M. Pendyala, and C. R. Bhat (2008). An analysis of children's leisure activity engagement: examining the day of week, location, physical activity level, and fixity dimensions. *Transportation 35*(5), 673–696.

[312] Seo, Y.-W. and B.-T. Zhang (2000a). Learning user's preferences by analyzing web-browsing behaviors. In *Proceedings of the fourth international conference on Autonomous agents*, pp. 381–387. ACM.

[313] Seo, Y.-W. and B.-T. Zhang (2000b). Learning user's preferences by analyzing web-browsing behaviors. In *Proceedings of the fourth international conference on Autonomous agents*, pp. 381–387. ACM.

[314] Seo, Y.-W. and B.-T. Zhang (2000c). A reinforcement learning agent for personalized information filtering. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pp. 248–251. ACM.

[315] Shapiro, S. S. and M. B. Wilk (1965). An analysis of variance test for normality (complete samples). *Biometrika 52*(3/4), 591–611.

[316] Shawky, D. and A. Badawi (2018). A reinforcement learning-based adaptive learning system. *Advances in Intelligent Systems and Computing*, 221–231.

[317] Shen, S. and M. Chi (2016). Reinforcement learning. *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization - UMAP '16*.

[318] Shiffman, S., A. A. Stone, and M. R. Hufford (2008). Ecological momentary assessment. *Annu. Rev. Clin. Psychol. 4*, 1–32.

[319] Shirazi, A. S., N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt (2014). Large-scale assessment of mobile notifications. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM Press.

[320] Silber, D. et al. (2003). *The case for eHealth*. European Institute of Public Administration Maastricht.

[321] Silva, B. M., J. J. Rodrigues, I. de la Torre Díez, M. López-Coronado, and K. Saleem (2015). Mobile-health: A review of current state in 2015. *Journal of biomedical informatics 56*, 265–272.

[322] Snyder, S. (2009). *The New World of Wireless: How to Compete in the 4G Revolution*. FT Press.

[323] Snyder, S. (2016). Hyper-personalizing the user experience through data. Presentation at the Mobile World Congress - Contextual Commerce, Barcelona.

[324] Song, L., W. Hsu, J. Xu, and M. van der Schaar (2016). Using contextual learning to improve diagnostic accuracy: Application in breast cancer screening. *IEEE Journal of Biomedical and Health Informatics 20*(3), 902–914.

[325] Sprague, N. and D. Ballard (2003). Multiple-goal reinforcement learning with modular sarsa (0).

[326] Srinivasan, A. R. and S. Chakraborty (2016). Path planning with user route preference - a reward surface approximation approach using orthogonal legendre polynomials. *2016 IEEE International Conference on Automation Science and Engineering (CASE).*

[327] Srivihok, A. and P. Sukonmanee (2005). Intelligent agent for e-tourism: Personalization travel support agent using reinforcement learning. In *WWW 2005*.

[328] Statista (2018). Number of available applications in the google play store from december 2009 to june 2018.

[329] Su, P.-h., Y.-B. Wang, T.-h. Yu, and L.-s. Lee (2013). A dialogue game framework with personalized training using reinforcement learning for computer-assisted language learning. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*.

[330] Su, P.-H., C.-H. Wu, and L.-S. Lee (2014). A recursive dialogue game for personalized computer-aided pronunciation training. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 1–1.

[331] Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pp. 216–224. Elsevier.

[332] Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in neural information processing systems*, pp. 1038–1044.

[333] Sutton, R. S. and A. G. Barto (1998). *Reinforcement learning: An introduction*, Volume 1. MIT press Cambridge.

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

[335] Sutton, R. S., D. A. McAllester, S. P. Singh, and Y. Mansour (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063.

[336] Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning 4*(1), 1–103.

[337] Tabatabaei, S. A., M. Hoogendoorn, and A. van Halteren (2018). Narrowing reinforcement learning: Overcoming the cold start problem for personalized health interventions. In *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 312–327. Springer.

[338] Taghipour, N. and A. Kardan (2008). A hybrid web recommender system based on q-learning. *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*.

[339] Taghipour, N., A. Kardan, and S. S. Ghidary (2007). Usage-based web recommendations. *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07*.

[340] Tang, J., C. Abraham, E. Stamp, and C. Greaves (2015, aug). How can weight-loss app designers best engage and support users? a qualitative investigation. *British Journal of Health Psychology 20*(1), 151–171.

[341] Tang, L., Y. Jiang, L. Li, and T. Li (2014). Ensemble contextual bandits for personalized recommendation. *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14*.

[342] Tang, L., Y. Jiang, L. Li, C. Zeng, and T. Li (2015). Personalized recommendation via parameter-free contextual bandits. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15*.

[343] Tang, L., R. Rosales, A. Singh, and D. Agarwal (2013). Automatic ad format selection via contextual bandits. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*.

[344] Tavakol, M. and U. Brefeld (2017). A unified contextual bandit framework for long- and short-term recommendations. *Lecture Notes in Computer Science*, 269–284.

[345] Taylor, M. E. and P. Stone (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research 10*(Jul), 1633–1685.

[346] Tegelund, B., H. Son, and D. Lee (2016). A task-oriented service personalization scheme for smart environments using reinforcement learning. *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*.

[347] Theocharous, G., P. S. Thomas, and M. Ghavamzadeh (2015a). Ad recommendation systems for life-time value optimization. *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion*.

[348] Theocharous, G., P. S. Thomas, and M. Ghavamzadeh (2015b). Personalized ad recommendation systems for life-time value optimization with guarantees. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[349] Thomas, P. S., G. Theocharous, and M. Ghavamzadeh (2015). High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[350] Torlay, L., M. Perrone-Bertolotti, E. Thomas, and M. Baciu (2017). Machine learning–xgboost analysis of language networks to classify patients with epilepsy. *Brain informatics 4*(3), 159.

[351] Triki, S. and C. Hanachi (2017). A self-adaptive system for improving autonomy and public spaces accessibility for elderly. *Smart Innovation, Systems and Technologies*, 53–66.

[352] Tseng, H.-H., Y. Luo, S. Cui, J.-T. Chien, R. K. Ten Haken, and I. E. Naqa (2017). Deep reinforcement learning for automated radiation adaptation in lung cancer. *Medical Physics 44*(12), 6690–6705.

[353] Tsiakas, K., C. Abellanoza, and F. Makedon (2016). Interactive learning and adaptation for robot assisted therapy for people with dementia. *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '16*.

[354] Tsiakas, K., M. Huber, and F. Makedon (2015). A multimodal adaptive session manager for physical rehabilitation exercising. *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '15*.

[355] Tsiakas, K., M. Papakostas, B. Chebaa, D. Ebert, V. Karkaletsis, and F. Makedon (2016). An interactive learning and adaptation framework for adaptive robot assisted therapy. *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '16*.

[356] Tsiakas, K., M. Papakostas, M. Theofanidis, M. Bell, R. Mihalcea, S. Wang, M. Burzo, and F. Makedon (2017). An interactive multisensing framework for personalized human robot collaboration and assistive training using reinforcement learning. *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '17*.

[357] Urieli, D. and P. Stone (2014). Tactex'13: a champion adaptive power trading agent. In *AAMAS*.

[358] van den Hoven, J. (2016). Clustering with optimised weights for Gower's metric. Master's thesis, Vrij University, Amsterdam, the Netherlands.

[359] Van Hasselt, H., A. Guez, and D. Silver (2016). Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*.

[360] Vasan, G. and P. M. Pilarski (2017). Learning from demonstration: Teaching a myoelectric prosthesis with an intact limb via reinforcement learning. *2017 International Conference on Rehabilitation Robotics (ICORR)*.

[361] Vasilyeva, E., M. Pechenizkiy, and S. Puuronen (2005). Towards the framework of adaptive user interfaces for ehealth. In *Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on*, pp. 139–144. IEEE.

[362] Volpato, T., B. R. N. Oliveira, L. Garcés, R. Capilla, and E. Y. Nakagawa (2017). Two perspectives on reference architecture sustainability. In *Proceedings of the 11th European Conference on Software Architecture: Companion*, pp. 188–194. ACM.

[363] Wachter, K., Y. H. Kim, and M. Kim (2012). Mobile users: Choosing to engage. *International Journal of Sales, Retailing and Marketing 1*(1).

[364] Wang, L., Y. Gao, C. Cao, and L. Wang (2012). Towards a general supporting framework for self-adaptive software systems. *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*.

[365] Wang, P., J. Rowe, B. Mott, and J. Lester (2016). Decomposing drama management in educational interactive narrative: A modular reinforcement learning approach. *Lecture Notes in Computer Science*, 270–282.

[366] Wang, P., J. P. Rowe, W. Min, B. W. Mott, and J. C. Lester (2017). Interactive narrative personalization with deep reinforcement learning. In *IJCAI*.

[367] Wang, X., Y. Wang, D. Hsu, and Y. Wang (2014a). Exploration in interactive personalized music recommendation. *ACM Transactions on Multimedia Computing, Communications, and Applications 11*(1), 1–22.

[368] Wang, X., Y. Wang, D. Hsu, and Y. Wang (2014b). Exploration in interactive personalized music recommendation: a reinforcement learning approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 11*(1), 7.

[369] Wang, X., M. Zhang, F. Ren, and T. Ito (2015). Gongbroker: A broker model for power trading in smart grid markets. *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*.

[370] Watkins, C. J. and P. Dayan (1992). Q-learning. *Machine learning 8*(3-4), 279–292.

[371] Webb, M. S., V. N. Simmons, and T. H. Brandon (2005). Tailored interventions for motivating smoking cessation: using placebo tailoring to examine the influence of expectancies and personalization. *Health Psychology 24*(2), 179.

[372] Wesson, J. L., A. Singh, and B. Van Tonder (2010). Can adaptive interfaces improve the usability of mobile applications? In *Human-Computer Interaction*, pp. 187–198. Springer.

[373] Weyns, D. (2017). Software engineering of self-adaptive systems: an organised tour and future challenges. *Chapter in Handbook of Software Engineering*.

[374] Weyns, D., M. U. Iftikhar, D. Hughes, and N. Matthys (2018). Applying architecture-based adaptation to automate the management of internet-of-things. In *European Conference on Software Architecture*, pp. 49–67. Springer.

[375] Weyns, D., M. U. Iftikhar, S. Malek, and J. Andersson (2012). Claims and supporting evidence for self-adaptive systems: A literature study. In

Proceedings of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp. 89–98. IEEE Press.

[376] Wiering, M. and M. Van Otterlo (2012a). Reinforcement learning. *Adaptation, learning, and optimization 12*.

[377] Wiering, M. and M. Van Otterlo (2012b). Reinforcement learning. *Adaptation, learning, and optimization 12*.

[378] Williams, P. A. H. and V. McCauley (2013). A rapidly moving target: Conformance with e-health standards for mobile computing. In *2nd Australian eHealth Informatics and Security Conference*.

[379] Wohlin, C. (2014a). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pp. 38. ACM.

[380] Wohlin, C. (2014b). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, New York, NY, USA, pp. 38:1–38:10. ACM.

[381] Wohlin, C., P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén (2012). *Experimentation in software engineering*. Springer Science & Business Media.

[382] Wu, G., Y. Ding, Y. Li, J. Luo, F. Zhang, and J. Fu (2017). Data-driven inverse learning of passenger preferences in urban public transits. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*.

[383] Xu, J., T. Xing, and M. van der Schaar (2016). Personalized course sequence recommendations. *IEEE Transactions on Signal Processing 64*(20), 5340–5352.

[384] Yang, M., Q. Qu, K. Lei, J. Zhu, Z. Zhao, X. Chen, and J. Z. Huang (2018). Investigating deep reinforcement learning techniques in personalized dialogue

generation. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 630–638. SIAM.

[385] Yang, M., W. Tu, Q. Qu, Z. Zhao, X. Chen, and J. Zhu (2018). Personalized response generation by dual-learning based domain adaptation. *Neural Networks 103*, 72–82.

[386] Yang, M., Z. Zhao, W. Zhao, X. Chen, J. Zhu, L. Zhou, and Z. Cao (2017). Personalized response generation via domain adaptation. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17*.

[387] Yang, Z., Z. Li, Z. Jin, and Y. Chen (2014). A systematic literature review of requirements modeling and analysis for self-adaptive systems. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 55–71. Springer.

[388] Yuan, S.-T. (2003). A personalized and integrative comparison-shopping engine and its applications. *Decision Support Systems 34*(2), 139–156.

[389] Yue, Y., S. A. Hong, and C. Guestrin (2012). Hierarchical exploration for accelerating contextual bandits. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 979–986. Omnipress.

[390] Zaidenberg, S. and P. Reignier (2011). Reinforcement learning of user preferences for a ubiquitous personal assistant. In *Advances in Reinforcement Learning*. IntechOpen.

[391] Zaidenberg, S., P. Reignier, and J. L. Crowley (2008). Reinforcement learning of context models for a ubiquitous personal assistant. *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, 254–264.

[392] Zeng, C., Q. Wang, S. Mokhtari, and T. Li (2016). Online context-aware recommendation with time varying multi-armed bandit. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*.

[393] Zepeda-Mendoza, M. L. and O. Resendis-Antonio (2013). Hierarchical agglomerative clustering. In *Encyclopedia of Systems Biology*, pp. 886–887. Springer.

[394] Zhang, B.-T. and Y.-W. Seo (2001a). Personalized web-document filtering using reinforcement learning. *Applied Artificial Intelligence 15*(7), 665–685.

[395] Zhang, B.-T. and Y.-W. Seo (2001b). Personalized web-document filtering using reinforcement learning. *Applied Artificial Intelligence 15*(7), 665–685.

[396] Zhang, Y., R. Chen, J. Tang, W. F. Stewart, and J. Sun (2017). Leap. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*.

[397] Zhao, T. and I. King (2016). Locality-sensitive linear bandit model for online social recommendation. *Lecture Notes in Computer Science*, 80–90.

[398] Zhao, Y., M. R. Kosorok, and D. Zeng (2009). Reinforcement learning design for cancer clinical trials. *Statistics in medicine 28*(26), 3294–3315.

[399] Zhao, Y., S. Wang, Y. Zou, J. Ng, and T. Ng (2017). Automatically learning user preferences for personalized service composition. *2017 IEEE International Conference on Web Services (ICWS)*.

[400] Zhao, Y., D. Zeng, M. A. Socinski, and M. R. Kosorok (2011). Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. *Biometrics 67*(4), 1422–1433.

[401] Zhao, Y., Q. Zhao, L. Xia, Z. Cheng, F. Wang, and F. Song (2013). A unified control framework of hvac system for thermal and acoustic comforts in office building. *2013 IEEE International Conference on Automation Science and Engineering (CASE)*.

[402] Zheng, G., F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li (2018). Drn. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*.

[403] Zheng, H. and J. Jumadinova (2016). Owls: Observational wireless life-enhancing system (extended abstract). In *AAMAS*.

[404] Zhou, L. and E. Brunskill (2016). Latent contextual bandits and their application to personalized recommendations for new users. In *IJCAI*.

[405] Zhou, M., Y. D. Mintz, Y. Fukuoka, K. Y. Goldberg, E. Flowers, P. Kaminsky, A. Castillejo, and A. Aswani (2018). Personalizing mobile fitness apps using reinforcement learning. In *IUI Workshops*.

[406] Zhu, F., J. Guo, Z. Xu, P. Liao, and J. Huang (2017). Group-driven reinforcement learning for personalized mhealth intervention. *arXiv preprint arXiv:1708.04001*.

[407] Zhu, R., Y.-Q. Zhao, G. Chen, S. Ma, and H. Zhao (2016). Greedy outcome weighted tree learning of optimal personalized treatment rules. *Biometrics 73*(2), 391–400.

SIKS Dissertatiereeks

2011

2011-01 Botond Cseke (RUN)

Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02 Nick Tinnemeier (UU)

Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

2011-03 Jan Martijn van der Werf (TUE)

Compositional Design and Verification of Component-Based Information Systems

2011-04 Hado van Hasselt (UU)

Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms

2011-05 Base van der Raadt (VU)

Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

2011-06 Yiwen Wang (TUE)

Semantically-Enhanced Recommendations in Cultural Heritage

2011-07 Yujia Cao (UT)

Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU)

BDI-based Generation of Robust Task-Oriented Dialogues

2011-09 Tim de Jong (OU)

Contextualised Mobile Media for Learning

2011-10 Bart Bogaert (UvT)

Cloud Content Contention

2011-11 Dhaval Vyas (UT)

Designing for Awareness: An Experience-focused HCI Perspective

2011-12 Carmen Bratosin (TUE)

Grid Architecture for Distributed Process Mining

2011-13 Xiaoyu Mao (UvT)

Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR)

Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA)

The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM)

Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA)

Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM)

Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU)

The Mind ' s Eye on Personal Profiles

2011-20 Qing Gu (VU)

Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD)

Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA)

System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA)

Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT)

Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)

Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU)

Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU)

Dynamic website optimization through autonomous management of design patterns

2011-28 Rianne Kaptein (UVA)

Effective Focused Retrieval by Exploiting Query Context and Document Structure

2011-29 Faisal Kamiran (TUE)

Discrimination-aware Classification

2011-30 Egon van den Broek (UT)

Affective Signal Processing (ASP): Unraveling the mystery of emotions

2011-31 Ludo Waltman (EUR)

Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

2011-32 Nees-Jan van Eck (EUR)

Methodological Advances in Bibliometric Mapping of Science

2011-33 Tom van der Weide (UU)

Arguing to Motivate Decisions

2011-34 Paolo Turrini (UU)

Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

2011-35 Maaike Harbers (UU)

Explaining Agent Behavior in Virtual Training

2011-36 Erik van der Spek (UU)

Experiments in serious game design: a cognitive approach

2011-37 Adriana Burlutiu (RUN)

Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

2011-38 Nyree Lemmens (UM)

Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU)

Organizing Adaptation using Agents in Serious Games

2011-40 Viktor Clerc (VU)

Architectural Knowledge Management in Global Software Development

2011-41 Luan Ibraimi (UT)

Cryptographically Enforced Distributed Data Access Control

2011-42 Michal Sindlar (UU)

Explaining Behavior through Mental State Attribution

2011-43 Henk van der Schuur (UU)

Process Improvement through Software Operation Knowledge

2011-44 Boris Reuderink (UT)

Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT)

Statistical Language Models for Alternative Sequence Selection

2011-46 Beibei Hu (TUD)

Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

2011-47 Azizi Bin Ab Aziz (VU)

Exploring Computational Models for Intelligent Support of Persons with Depression

2011-48 Mark Ter Maat (UT)

Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

2011-49 Andreea Niculescu (UT)

Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

2012

2012-01 Terry Kakeeto (UvT)

Relationship Marketing for SMEs in Uganda

2012-02 Muhammad Umair (VU)

Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

2012-03 Adam Vanya (VU)

Supporting Architecture Evolution by Mining Software Repositories

2012-04 Jurriaan Souer (UU)

Development of Content Management System-based Web Applications

2012-05 Marijn Plomp (UU)

Maturing Interorganisational Information Systems

2012-06 Wolfgang Reinhardt (OU)

Awareness Support for Knowledge Workers in Research Networks

2012-07 Rianne van Lambalgen (VU)

When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

2012-08 Gerben de Vries (UVA)

Kernel Methods for Vessel Trajectories

2012-09 Ricardo Neisse (UT)

Trust and Privacy Management Support for Context-Aware Service Platforms

2012-10 David Smits (TUE)

Towards a Generic Distributed Adaptive Hypermedia Environment

2012-11 J.C.B. Rantham Prabhakara (TUE)

Process Mining in the Large: Preprocessing, Discovery, and Diagnostics

2012-12 Kees van der Sluijs (TUE)

Model Driven Design and Data Integration in Semantic Web Information Systems

2012-13 Suleman Shahid (UvT)

Fun and Face: Exploring non-verbal expressions of emotion during playful interactions

2012-14 Evgeny Knutov (TUE)

Generic Adaptation Framework for Unifying Adaptive Web-based Systems

2012-15 Natalie van der Wal (VU)

Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes

2012-16 Fiemke Both (VU)

Helping people by understanding them - Ambient Agents supporting task execution and depression treatment

2012-17 Amal Elgammal (UvT)

Towards a Comprehensive Framework for Business Process Compliance

2012-18 Eltjo Poort (VU)

Improving Solution Architecting Practices

2012-19 Helen Schonenberg (TUE)

What's Next? Operational Support for Business Process Execution

2012-20 Ali Bahramisharif (RUN)

Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing

2012-21 Roberto Cornacchia (TUD)

Querying Sparse Matrices for Information Retrieval

2012-22 Thijs Vis (UvT)

Intelligence, politie en veiligheidsdienst: verenigbare grootheden?

2012-23 Christian Muehl (UT)

Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction

2012-24 Laurens van der Werff (UT)

Evaluation of Noisy Transcripts for Spoken Document Retrieval

2012-25 Silja Eckartz (UT)

Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application

2012-26 Emile de Maat (UVA)

Making Sense of Legal Text

2012-27 Hayrettin Gurkok (UT)

Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games

2012-28 Nancy Pascall (UvT)

Engendering Technology Empowering Women

2012-29 Almer Tigelaar (UT)

Peer-to-Peer Information Retrieval

2012-30 Alina Pommeranz (TUD)

Designing Human-Centered Systems for Reflective Decision Making

2012-31 Emily Bagarukayo (RUN)

A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure

2012-32 Wietske Visser (TUD)

Qualitative multi-criteria preference representation and reasoning

2012-33 Rory Sie (OUN)

Coalitions in Cooperation Networks (COCOON)

2012-34 Pavol Jancura (RUN)

Evolutionary analysis in PPI networks and applications

2012-35 Evert Haasdijk (VU)

Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics

2012-36 Denis Ssebugwawo (RUN)

Analysis and Evaluation of Collaborative Modeling Processes

2012-37 Agnes Nakakawa (RUN)

A Collaboration Process for Enterprise Architecture Creation

2012-38 Selmar Smit (VU)

Parameter Tuning and Scientific Testing in Evolutionary Algorithms

2012-39 Hassan Fatemi (UT)

Risk-aware design of value and coordination networks

2012-40 Agus Gunawan (UvT)

Information Access for SMEs in Indonesia

2012-41 Sebastian Kelle (OU)

Game Design Patterns for Learning

2012-42 Dominique Verpoorten (OU)

Reflection Amplifiers in self-regulated Learning

2012-43

Withdrawn

2012-44 Anna Tordai (VU)

On Combining Alignment Techniques

2012-45 Benedikt Kratz (UvT)

A Model and Language for Business-aware Transactions

2012-46 Simon Carter (UVA)

Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

2012-47 Manos Tsagkias (UVA)

Mining Social Media: Tracking Content and Predicting Behavior

2012-48 Jorn Bakker (TUE)

Handling Abrupt Changes in Evolving Time-series Data

2012-49 Michael Kaisers (UM)

Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

2012-50 Steven van Kervel (TUD)

Ontologogy driven Enterprise Information Systems Engineering

2012-51 Jeroen de Jong (TUD)

Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

2013

2013-01 Viorel Milea (EUR)

News Analytics for Financial Decision Support

2013-02 Erietta Liarou (CWI)

MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing

2013-03 Szymon Klarman (VU)

Reasoning with Contexts in Description Logics

2013-04 Chetan Yadati (TUD)

Coordinating autonomous planning and scheduling

2013-05 Dulce Pumareja (UT)

Groupware Requirements Evolutions Patterns

2013-06 Romulo Goncalves (CWI)

The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience

2013-07 Giel van Lankveld (UvT)

Quantifying Individual Player Differences

2013-08 Robbert-Jan Merk (VU)

Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

2013-09 Fabio Gori (RUN)

Metagenomic Data Analysis: Computational Methods and Applications

2013-10 Jeewanie Jayasinghe Arachchige (UvT)

A Unified Modeling Framework for Service Design.

2013-11 Evangelos Pournaras (TUD)

Multi-level Reconfigurable Self-organization in Overlay Services

2013-12 Marian Razavian (VU)

Knowledge-driven Migration to Services

2013-13 Mohammad Safiri (UT)

Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly

2013-14 Jafar Tanha (UVA)

Ensemble Approaches to Semi-Supervised Learning Learning

2013-15 Daniel Hennes (UM)

Multiagent Learning - Dynamic Games and Applications

2013-16 Eric Kok (UU)

Exploring the practical benefits of argumentation in multi-agent deliberation

2013-17 Koen Kok (VU)

The PowerMatcher: Smart Coordination for the Smart Electricity Grid

2013-18 Jeroen Janssens (UvT)

Outlier Selection and One-Class Classification

2013-19 Renze Steenhuizen (TUD)

Coordinated Multi-Agent Planning and Scheduling

2013-20 Katja Hofmann (UvA)

Fast and Reliable Online Learning to Rank for Information Retrieval

2013-21 Sander Wubben (UvT)

Text-to-text generation by monolingual machine translation

2013-22 Tom Claassen (RUN)

Causal Discovery and Logic

2013-23 Patricio de Alencar Silva (UvT)

Value Activity Monitoring

2013-24 Haitham Bou Ammar (UM)

Automated Transfer in Reinforcement Learning

2013-25 Agnieszka Anna Latoszek-Berendsen (UM)

Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

2013-26 Alireza Zarghami (UT)

Architectural Support for Dynamic Homecare Service Provisioning

2013-27 Mohammad Huq (UT)

Inference-based Framework Managing Data Provenance

2013-28 Frans van der Sluis (UT)

When Complexity becomes Interesting: An Inquiry into the Information eXperience

2013-29 Iwan de Kok (UT)

Listening Heads

2013-30 Joyce Nakatumba (TUE)

Resource-Aware Business Process Management: Analysis and Support

2013-31 Dinh Khoa Nguyen (UvT)

Blueprint Model and Language for Engineering Cloud Applications

2013-32 Kamakshi Rajagopal (OUN)

Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development

2013-33 Qi Gao (TUD)

User Modeling and Personalization in the Microblogging Sphere

2013-34 Kien Tjin-Kam-Jet (UT)

Distributed Deep Web Search

2013-35 Abdallah El Ali (UvA)

Minimal Mobile Human Computer Interaction

2013-36 Than Lam Hoang (TUe)

Pattern Mining in Data Streams

2013-37 Dirk Börner (OUN)

Ambient Learning Displays

2013-38 Eelco den Heijer (VU)

Autonomous Evolutionary Art

2013-39 Joop de Jong (TUD)

A Method for Enterprise Ontology based Design of Enterprise Information Systems

2013-40 Pim Nijssen (UM)

Monte-Carlo Tree Search for Multi-Player Games

2013-41 Jochem Liem (UVA)

Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning

2013-42 Léon Planken (TUD)

Algorithms for Simple Temporal Reasoning

2013-43 Marc Bron (UVA)

Exploration and Contextualization through Interaction and Concepts

2014

2014-01 Nicola Barile (UU)

Studies in Learning Monotone Models from Data

2014-02 Fiona Tuliyano (RUN)

Combining System Dynamics with a Domain Modeling Method

2014-03 Sergio Raul Duarte Torres (UT)

Information Retrieval for Children: Search Behavior and Solutions

2014-04 Hanna Jochmann-Mannak (UT)

Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation

2014-05 Jurriaan van Reijsen (UU)

Knowledge Perspectives on Advancing Dynamic Capability

2014-06 Damian Tamburri (VU)

Supporting Networked Software Development

2014-07 Arya Adriansyah (TUE)

Aligning Observed and Modeled Behavior

2014-08 Samur Araujo (TUD)

Data Integration over Distributed and Heterogeneous Data Endpoints

2014-09 Philip Jackson (UvT)

Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language

2014-10 Ivan Salvador Razo Zapata (VU)

Service Value Networks

2014-11 Janneke van der Zwaan (TUD)

An Empathic Virtual Buddy for Social Support

2014-12 Willem van Willigen (VU)

Look Ma, No Hands: Aspects of Autonomous Vehicle Control

2014-13 Arlette van Wissen (VU)

Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains

2014-14 Yangyang Shi (TUD)

Language Models With Meta-information

2014-15 Natalya Mogles (VU)

Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare

2014-16 Krystyna Milian (VU)

Supporting trial recruitment and design by automatically interpreting eligibility criteria

2014-17 Kathrin Dentler (VU)

Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability

2014-18 Mattijs Ghijsen (VU)

Methods and Models for the Design and Study of Dynamic Agent Organizations

2014-19 Vinicius Ramos (TUE)

Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support

2014-20 Mena Habib (UT)

Named Entity Extraction and Disambiguation for Informal Text: The Missing Link

2014-21 Kassidy Clark (TUD)

Negotiation and Monitoring in Open Environments

2014-22 Marieke Peeters (UU)

Personalized Educational Games - Developing agent-supported scenario-based training

2014-23 Eleftherios Sidirourgos (UvA/CWI)

Space Efficient Indexes for the Big Data Era

2014-24 Davide Ceolin (VU)

Trusting Semi-structured Web Data

2014-25 Martijn Lappenschaar (RUN)

New network models for the analysis of disease interaction

2014-26 Tim Baarslag (TUD)

What to Bid and When to Stop

2014-27 Rui Jorge Almeida (EUR)

Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty

2014-28 Anna Chmielowiec (VU)

Decentralized k-Clique Matching

2014-29 Jaap Kabbedijk (UU)

Variability in Multi-Tenant Enterprise Software

2014-30 Peter de Cock (UvT)

Anticipating Criminal Behaviour

2014-31 Leo van Moergestel (UU)

Agent Technology in Agile Multiparallel Manufacturing and Product Support

2014-32 Naser Ayat (UvA)

On Entity Resolution in Probabilistic Data

2014-33 Tesfa Tegegne (RUN)

Service Discovery in eHealth

2014-34 Christina Manteli (VU)

The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.

2014-35 Joost van Ooijen (UU)

Cognitive Agents in Virtual Worlds: A Middleware Design Approach

2014-36 Joos Buijs (TUE)

Flexible Evolutionary Algorithms for Mining Structured Process Models

2014-37 Maral Dadvar (UT)

Experts and Machines United Against Cyberbullying

2014-38 Danny Plass-Oude Bos (UT)

Making brain-computer interfaces better: improving usability through post-processing.

2014-39 Jasmina Maric (UvT)

Web Communities, Immigration, and Social Capital

2014-40 Walter Omona (RUN)

A Framework for Knowledge Management Using ICT in Higher Education

2014-41 Frederic Hogenboom (EUR)

Automated Detection of Financial Events in News Text

2014-42 Carsten Eijckhof (CWI/TUD)

Contextual Multidimensional Relevance Models

2014-43 Kevin Vlaanderen (UU)

Supporting Process Improvement using Method Increments

2014-44 Paulien Meesters (UvT)

Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.

2014-45 Birgit Schmitz (OUN)

Mobile Games for Learning: A Pattern-Based Approach

2014-46 Ke Tao (TUD)

Social Web Data Analytics: Relevance, Redundancy, Diversity

2014-47 Shangsong Liang (UVA)

Fusion and Diversification in Information Retrieval

2015

2015-01 Niels Netten (UvA)

Machine Learning for Relevance of Information in Crisis Response

2015-02 Faiza Bukhsh (UvT)

Smart auditing: Innovative Compliance Checking in Customs Controls

2015-03 Twan van Laarhoven (RUN)

Machine learning for network data

2015-04 Howard Spoelstra (OUN)

Collaborations in Open Learning Environments

2015-05 Christoph Bösch (UT)

Cryptographically Enforced Search Pattern Hiding

2015-06 Farideh Heidari (TUD)

Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes

2015-07 Maria-Hendrike Peetz (UvA)

Time-Aware Online Reputation Analysis

2015-08 Jie Jiang (TUD)

Organizational Compliance: An agent-based model for designing and evaluating organizational interactions

2015-09 Randy Klaassen (UT)

HCI Perspectives on Behavior Change Support Systems

2015-10 Henry Hermans (OUN)

OpenU: design of an integrated system to support lifelong learning

2015-11 Yongming Luo (TUE)

Designing algorithms for big graph datasets: A study of computing bisimulation and joins

2015-12 Julie M. Birkholz (VU)

Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks

2015-13 Giuseppe Procaccianti (VU)

Energy-Efficient Software

2015-14 Bart van Straalen (UT)

A cognitive approach to modeling bad news conversations

2015-15 Klaas Andries de Graaf (VU)

Ontology-based Software Architecture Documentation

2015-16 Changyun Wei (UT)

Cognitive Coordination for Cooperative Multi-Robot Teamwork

2015-17 André van Cleeff (UT)

Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs

2015-18 Holger Pirk (CWI)

Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories

2015-19 Bernardo Tabuenca (OUN)

Ubiquitous Technology for Lifelong Learners

2015-20 Loïs Vanhée (UU)

Using Culture and Values to Support Flexible Coordination

2015-21 Sibren Fetter (OUN)

Using Peer-Support to Expand and Stabilize Online Learning

2015-22 Zhemin Zhu (UT)

Co-occurrence Rate Networks

2015-23 Luit Gazendam (VU)

Cataloguer Support in Cultural Heritage

2015-24 Richard Berendsen (UVA)

Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation

2015-25 Steven Woudenberg (UU)

Bayesian Tools for Early Disease Detection

2015-26 Alexander Hogenboom (EUR)

Sentiment Analysis of Text Guided by Semantics and Structure

2015-27 Sándor Héman (CWI)

Updating compressed column-stores

2015-28 Janet Bagorogoza (TiU)

Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO

2015-29 Hendrik Baier (UM)

Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains

2015-30 Kiavash Bahreini (OUN)

Real-time Multimodal Emotion Recognition in E-Learning

2015-31 Yakup Koç (TUD)

On Robustness of Power Grids

2015-32 Jerome Gard (UL)

Corporate Venture Management in SMEs

2015-33 Frederik Schadd (UM)

Ontology Mapping with Auxiliary Resources

2015-34 Victor de Graaff (UT)

Geosocial Recommender Systems

2015-35 Junchao Xu (TUD)

Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

2016

2016-01 Syed Saiden Abbas (RUN)

Recognition of Shapes by Humans and Machines

2016-02 Michiel Christiaan Meulendijk (UU)

Optimizing medication reviews through decision support: prescribing a better pill to swallow

2016-03 Maya Sappelli (RUN)

Knowledge Work in Context: User Centered Knowledge Worker Support

2016-04 Laurens Rietveld (VU)

Publishing and Consuming Linked Data

2016-05 Evgeny Sherkhonov (UVA)

Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

2016-06 Michel Wilson (TUD)

Robust scheduling in an uncertain environment

2016-07 Jeroen de Man (VU)

Measuring and modeling negative emotions for virtual training

2016-08 Matje van de Camp (TiU)

A Link to the Past: Constructing Historical Social Networks from Unstructured Data

2016-09 Archana Nottamkandath (VU)

Trusting Crowdsourced Information on Cultural Artefacts

2016-10 George Karafotias (VUA)

Parameter Control for Evolutionary Algorithms

2016-11 Anne Schuth (UVA)

Search Engines that Learn from Their Users

2016-12 Max Knobbout (UU)

Logics for Modelling and Verifying Normative Multi-Agent Systems

2016-13 Nana Baah Gyan (VU)

The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

2016-14 Ravi Khadka (UU)

Revisiting Legacy Software System Modernization

2016-15 Steffen Michels (RUN)

Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

2016-16 Guangliang Li (UVA)

Socially Intelligent Autonomous Agents that Learn from Human Reward

2016-17 Berend Weel (VU)

Towards Embodied Evolution of Robot Organisms

2016-18 Albert Meroño Peñuela (VU)

Refining Statistical Data on the Web

2016-19 Julia Efremova (Tu/e)

Mining Social Structures from Genealogical Data

2016-20 Daan Odijk (UVA)

Context & Semantics in News & Web Search

2016-21 Alejandro Moreno Célleri (UT)

From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

2016-22 Grace Lewis (VU)

Software Architecture Strategies for Cyber-Foraging Systems

2016-23 Fei Cai (UVA)

Query Auto Completion in Information Retrieval

2016-24 Brend Wanders (UT)

Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach

2016-25 Julia Kiseleva (TU/e)

Using Contextual Information to Understand Searching and Browsing Behavior

2016-26 Dilhan Thilakarathne (VU)

In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains

2016-27 Wen Li (TUD)

Understanding Geo-spatial Information on Social Media

2016-28 Mingxin Zhang (TUD)

Large-scale Agent-based Social Simulation - A study on epidemic prediction and control

2016-29 Nicolas Höning (TUD)

Peak reduction in decentralised electricity systems - Markets and prices for flexible planning

2016-30 Ruud Mattheij (UvT)

The Eyes Have It

2016-31 Mohammad Khelghati (UT)

Deep web content monitoring

2016-32 Eelco Vriezekolk (UT)

Assessing Telecommunication Service Availability Risks for Crisis Organisations

2016-33 Peter Bloem (UVA)

Single Sample Statistics, exercises in learning from just one example

2016-34 Dennis Schunselaar (TUE)

Configurable Process Trees: Elicitation, Analysis, and Enactment

2016-35 Zhaochun Ren (UVA)

Monitoring Social Media: Summarization, Classification and Recommendation

2016-36 Daphne Karreman (UT)

Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies

2016-37 Giovanni Sileno (UvA)

Aligning Law and Action - a conceptual and computational inquiry

2016-38 Andrea Minuto (UT)

Materials that Matter - Smart Materials meet Art & Interaction Design

2016-39 Merijn Bruijnes (UT)

Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect

2016-40 Christian Detweiler (TUD)

Accounting for Values in Design

2016-41 Thomas King (TUD)

Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

2016-42 Spyros Martzoukos (UVA)

Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

2016-43 Saskia Koldijk (RUN)

Context-Aware Support for Stress Self-Management: From Theory to Practice

2017-44 Thibault Sellam (UVA)

Automatic Assistants for Database Exploration

2016-45 Bram van de Laar (UT)

Experiencing Brain-Computer Interface Control

2016-46 Jorge Gallego Perez (UT)

Robots to Make you Happy

2016-47 Christina Weber (UL)

Real-time foresight - Preparedness for dynamic innovation networks

2016-48 Tanja Buttler (TUD)

Collecting Lessons Learned

2016-49 Gleb Polevoy (TUD)

Participation and Interaction in Projects. A Game-Theoretic Analysis

2016-50 Yan Wang (UVT)

The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

2017

2017-01 Jan-Jaap Oerlemans (UL)

Investigating Cybercrime

2017-02 Sjoerd Timmer (UU)

Designing and Understanding Forensic Bayesian Networks using Argumentation

2017-03 Daniël Harold Telgen (UU)

Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines

2017-04 Mrunal Gawade (CWI)

Multi-core Parallelism in a Column-store

2017-05 Mahdieh Shadi (UVA)

Collaboration Behavior

2017-06 Damir Vandic (EUR)

Intelligent Information Systems for Web Product Search

2017-07 Roel Bertens (UU)

Insight in Information: from Abstract to Anomaly

2017-08 Rob Konijn (VU)

Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

2017-09 Dong Nguyen (UT)

Text as Social and Cultural Data: A Computational Perspective on Variation in Text

2017-10 Robby van Delden (UT)

(Steering) Interactive Play Behavior

2017-11 Florian Kunneman (RUN)

Modelling patterns of time and emotion in Twitter #anticipointment

2017-12 Sander Leemans (TUE)

Robust Process Mining with Guarantees

2017-13 Gijs Huisman (UT)

Social Touch Technology - Extending the reach of social touch through haptic technology

2017-14 Shoshannah Tekofsky (UvT)

You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior

2017-15 Peter Berck (RUN)

Memory-Based Text Correction

2017-16 Aleksandr Chuklin (UVA)

Understanding and Modeling Users of Modern Search Engines

2017-17 Daniel Dimov (UL)

Crowdsourced Online Dispute Resolution

2017-18 Ridho Reinanda (UVA)

Entity Associations for Search

2017-19 Jeroen Vuurens (UT)

Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

2017-20 Mohammadbashir Sedighi (TUD)

Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

2017-21 Jeroen Linssen (UT)

Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

2017-22 Sara Magliacane (VU)

Logics for causal inference under uncertainty

2017-23 David Graus (UVA)

Entities of Interest — Discovery in Digital Traces

2017-24 Chang Wang (TUD)

Use of Affordances for Efficient Robot Learning

2017-25 Veruska Zamborlini (VU)

Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

2017-26 Merel Jung (UT)

Socially intelligent robots that understand and respond to human touch

2017-27 Michiel Joosse (UT)

Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors

2017-28 John Klein (VU)

Architecture Practices for Complex Contexts

2017-29 Adel Alhuraibi (UvT)

From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"

2017-30 Wilma Latuny (UvT)

The Power of Facial Expressions

2017-31 Ben Ruijl (UL)

Advances in computational methods for QFT calculations

2017-32 Thaer Samar (RUN)

Access to and Retrievability of Content in Web Archives

2017-33 Brigit van Loggem (OU)

Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity

2017-34 Maren Scheffel (OU)

The Evaluation Framework for Learning Analytics

2017-35 Martine de Vos (VU)

Interpreting natural science spreadsheets

2017-36 Yuanhao Guo (UL)

Shape Analysis for Phenotype Characterisation from High-throughput Imaging

2017-37 Alejandro Montes Garcia (TUE)

WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy

2017-38 Alex Kayal (TUD)

Normative Social Applications

2017-39 Sara Ahmadi (RUN)

Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR

2017-40 Altaf Hussain Abro (VUA)

Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems

2017-41 Adnan Manzoor (VUA)

Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

2017-42 Elena Sokolova (RUN)

Causal discovery from mixed and missing data with applications on ADHD datasets

2017-43 Maaike de Boer (RUN)

Semantic Mapping in Video Retrieval

2017-44 Garm Lucassen (UU)

Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

2017-45 Bas Testerink (UU)

Decentralized Runtime Norm Enforcement

2017-46 Jan Schneider (OU)

Sensor-based Learning Support

2017-47 Jie Yang (TUD)

Crowd Knowledge Creation Acceleration

2017-48 Angel Suarez (OU)

Collaborative inquiry-based learning

2018

2018-01 Han van der Aa (VUA)

Comparing and Aligning Process Representations

2018-02 Felix Mannhardt (TUE)

Multi-perspective Process Mining

2018-03 Steven Bosems (UT)

Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

2018-04 Jordan Janeiro (TUD)

Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

2018-05 Hugo Huurdeman (UVA)

Supporting the Complex Dynamics of the Information Seeking Process

2018-06 Dan Ionita (UT)

Model-Driven Information Security Risk Assessment of Socio-Technical Systems

2018-07 Jieting Luo (UU)

A formal account of opportunism in multi-agent systems

2018-08 Rick Smetsers (RUN)

Advances in Model Learning for Software Systems

2018-09 Xu Xie (TUD)

Data Assimilation in Discrete Event Simulations

2018-10 Julienka Mollee (VUA)

Moving forward: supporting physical activity behavior change through intelligent technology

2018-11 Mahdi Sargolzaei (UVA)

Enabling Framework for Service-oriented Collaborative Networks

2018-12 Xixi Lu (TUE)

Using behavioral context in process mining

2018-13 Seyed Amin Tabatabaei (VUA)

Computing a Sustainable Future

2018-14 Bart Joosten (UVT)

Detecting Social Signals with Spatiotemporal Gabor Filters

2018-15 Naser Davarzani (UM)

Biomarker discovery in heart failure

2018-16 Jaebok Kim (UT)

Automatic recognition of engagement and emotion in a group of children

2018-17 Jianpeng Zhang (TUE)

On Graph Sample Clustering

2018-18 Henriette Nakad (UL)

De Notaris en Private Rechtspraak

2018-19 Minh Duc Pham (VUA)

Emergent relational schemas for RDF

2018-20 Manxia Liu (RUN)

Time and Bayesian Networks

2018-21 Aad Slootmaker (OUN)

EMERGO: a generic platform for authoring and playing scenario-based serious games

2018-22 Eric Fernandes de Mello Araujo (VUA)

Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

2018-23 Kim Schouten (EUR)

Semantics-driven Aspect-Based Sentiment Analysis

2018-24 Jered Vroon (UT)

Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots

2018-25 Riste Gligorov (VUA)

Serious Games in Audio-Visual Collections

2018-26 Roelof Anne Jelle de Vries (UT)

Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology

2018-27 Maikel Leemans (TUE)

Hierarchical Process Mining for Scalable Software Analysis

2018-28 Christian Willemse (UT)

Social Touch Technologies: How they feel and how they make you feel

2018-29 Yu Gu (UVT)

Emotion Recognition from Mandarin Speech

2018-30 Wouter Beek (VU)

The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

2019

2019-01 Rob van Eijk (UL)

Comparing and Aligning Process Representations

2019-02 Emmanuelle Beauxis Aussalet (CWI, UU)

Statistics and Visualizations for Assessing Class Size Uncertainty

2019-03 Eduardo Gonzalez Lopez de Murillas (TUE)

Process Mining on Databases: Extracting Event Data from Real Life Data Sources

2019-04 Ridho Rahmadi (RUN)

Finding stable causal structures from clinical data

2019-05 Sebastiaan van Zelst (TUE)

Process Mining with Streaming Data

2019-06 Chris Dijkshoorn (VU)

Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

2019-07 Soude Fazeli (TUD)

2019-08 Frits de Nijs (TUD)

Resource-constrained Multi-agent Markov Decision Processes

2019-09 Fahimeh Alizadeh Moghaddam (UVA)

Self-adaptation for energy efficiency in software systems

2019-10 Qing Chuan Ye (EUR)

Multi-objective Optimization Methods for Allocation and Prediction

2019-11 Yue Zhao (TUD)

Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs

2019-12 Jacqueline Heinerman (VU)

Better Together

2019-13 Guanliang Chen (TUD)

MOOC Analytics: Learner Modeling and Content Generation

2019-14 Daniel Davis (TUD)

Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

2019-15 Erwin Walraven (TUD)

Planning under Uncertainty in Constrained and Partially Observable Environments

2019-16 Guangming Li (TUE)

Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models

2019-17 Ali Hurriyetoglu (RUN)

Extracting actionable information from microtexts

2019-18 Gerard Wagenaar (UU)

Artefacts in Agile Team Communication

2019-19 Vincent Koeman (TUD)

Tools for Developing Cognitive Agents

2019-20 Chide Groenouwe (UU)
Fostering technically augmented human collective intelligence

2019-21 Cong Liu (TUE)
Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

2019-22 Martin van den Berg (VU)
Improving IT Decisions with Enterprise Architecture

2019-23 Qin Liu (TUD)
Intelligent Control Systems: Learning, Interpreting, Verification

2019-24 Anca Dumitrache (VU)
Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing

2019-25 Emiel van Miltenburg (VU)
Pragmatic factors in (automatic) image description

2019-26 Prince Singh (UT)
An Integration Platform for Synchromodal Transport

2019-27 Alessandra Antonaci (OUN)
The Gamification Design Process applied to (Massive) Open Online Courses

2019-28 Esther Kuinderman (UL)
Cleared for take-off: Game-based learning to prepare airline pilots for critical situations

2019-29 Daniel Formolo (VU)
Using virtual agents for simulation and training of social skills in safety-critical circumstances

2019-30 Vahid Yazdanpanah (UT)
Multiagent Industrial Symbiosis Systems

2019-31 Milan Jelisavcic (VU)
Alive and Kicking: Baby Steps in Robotics

2019-32 Chiara Sironi (UM)
Monte-Carlo Tree Search for Artificial General Intelligence in Games

2019-33 Anil Yaman (TUE)
Evolution of Biologically Inspired Learning in Artificial Neural Networks

2019-34 Negar Ahmadi (TUE)
EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES

2019-35 Lisa Facey-Shaw (OUN)
Gamification with digital badges in learning programming

2019-36 Kevin Ackermans (OUN)
Designing Video-Enhanced Rubrics to Master Complex Skills

2019-37 Jian Fang (TUD)
Database Acceleration on FPGAs

2019-38 Akos Kadar (OUN)
Learning visually grounded and multilingual representations

2020

2020-01 Armon Toubman (UL)

Calculated Moves: Generating Air Combat Behaviour

2020-02 Marcos de Paula Bueno (UL)

Unraveling Temporal Processes using Probabilistic Graphical Models

2020-03 Mostafa Deghani (UvA)

Learning with Imperfect Supervision for Language Understanding

2020-04 Maarten van Gompel (RUN)

Context as Linguistic Bridges

2020-05 Yulong Pei (TUE)

On local and global structure mining

2020-06 Preethu Rose Anish (UT)

Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support

2020-07 Wim van der Vegt (OUN)

Towards a software architecture for reusable game components

2020-08 Ali Mirsoleimani (UL)

Structured Parallel Programming for Monte Carlo Tree Search

2020-09 Myriam Traub (UU)

Measuring Tool Bias and Improving Data Quality for Digital Humanities Research

2020-10 Alifah Syamsiyah (TUE)

In-database Preprocessing for Process Mining

2020-11 Sepideh Mesbah (TUD)

Semantic-Enhanced Training Data Augmentation Methods for Long-Tail Entity Recognition Models

2020-12 Ward van Breda (VU)

Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment

2020-13 Marco Virgolin (CWI)

Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

2020-14 Mark Raasveldt (CWI/UL)

Integrating Analytics with Relational Databases

2020-15 Konstantinos Georgiadis (OUN)

Smart CAT: Machine Learning for Configurable Assessments in Serious Games

2020-16 Ilona Wilmont (RUN)

Cognitive Aspects of Conceptual Modelling

2020-17 Daniele Di Mitri (OUN)

The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences

2020-18 Georgios Methenitis (TUD)

Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems

2020-19 Guido van Capelleveen (UT)

Industrial Symbiosis Recommender Systems

2020-20 Albert Hankel (VU)

Embedding Green ICT Maturity in Organisations

2020-21 Karine da Silva Miras de Araujo (VU)

Where is the robot?: Life as it could be

2020-22 Maryam Masoud Khamis (RUN)

Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar

2020-23 Rianne Conijn (UT)

The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging

2020-24 Lenin da Nobrega Medeiros (VUA/RUN)

How are you feeling, human? Towards emotionally supportive chatbots

2020-25 Xin Du (TUE)

The Uncertainty in Exceptional Model Mining

2020-26 Krzysztof Leszek Sadowski (UU)

GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization

2020-27 Ekaterina Muravyeva (TUD)

Personal data and informed consent in an educational context

2020-28 Bibeg Limbu (TUD)

Multimodal interaction for deliberate practice: Training complex skills with augmented reality

2020-29 Ioan Gabriel Bucur (RUN)

Being Bayesian about Causal Inference

2020-30 Bob Zadok Blok (UL)

Creatief, Creatieve, Creatiefst

2020-31 Gongjin Lan (VU)

Learning better – From Baby to Better

2020-32 Jason Rhuggenaath (TUE)

Revenue management in online markets: pricing and online advertising

2020-33 Rick Gilsing (TUE)

Supporting service-dominant business model evaluation in the context of business model innovation

2020-34 Anna Bon (MU)

Intervention or Collaboration? Redesigning Information and Communication Technologies for Development

2020-35 Siamak Farshidi (UU)

Multi-Criteria Decision-Making in Software Production

2021

2021-01 Francisco Xavier Dos Santos Fonseca (TUD)

Location-based Games for Social Interaction in Public Space

2021-02 Rijk Mercuur (TUD)

Simulating Human Routines:Integrating Social Practice Theory in Agent-Based Models

2021-03 Seyyed Hadi Hashemi (UVA)

Modeling Users Interacting with Smart Devices

2021-04 Ioana Jivet (OU)

The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning

2021-05 Davide Dell'Anna (UU)

Data-Driven Supervision of Autonomous Systems

2021-06 Daniel Davison (UT)

"Hey robot, what do you think?" How children learn with a social robot

2021-07 Armel Lefebvre (UU)

Research data management for open science

2021-08 Nardie Fanchamps (OU)

The Influence of Sense-Reason-Act Programming on Computational Thinking

2021-09 Cristina Zaga (UT)

The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play

2021-10 Quinten Meertens (UvA)

Misclassification Bias in Statistical Learning

2021-11 Anne van Rossum (UL)

Nonparametric Bayesian Methods in Robotic Vision

2021-12 Lei Pi (UL)

External Knowledge Absorption in Chinese SMEs

2021-13 Bob R. Schadenberg (UT)

Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

2021-14 Negin Samaeemofrad (UL)

Business Incubators: The Impact of Their Support

2021-15 Onat Ege Adali (TU/e)

Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm

2021-16 Esam A. H. Ghaleb (MU)

BIMODAL EMOTION RECOGNITION FROM AUDIO-VISUAL CUES

2021-17 Dario Dotti (UM)

Human Behavior Understanding from motion and bodily cues using deep neural networks

2021-18 Remi Wieten (UU)

Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

2021-19 Roberto Verdecchia (VU)

Architectural Technical Debt: Identification and Management

2021-20 Masoud Mansoury (TU/e)

Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

2021-21 Pedro Thiago Timbó Holanda (CWI)

Progressive Indexes

2021-22 Sihang Qiu (TUD)

Conversational Crowdsourcing

2021-23 Hugo Manuel Proença (LIACS)

Robust rules for prediction and description

2021-24 Kaijie Zhu (TUE)

On Efficient Temporal Subgraph Query Processing