Head Office: Università degli Studi di Padova

Brain, Mind and Computer Science

Ph.D. COURSE IN: Computer Science

CURRICOLO: Informatica

SERIES: XXX

Personalized Game Content Generation and Recommendation for Gamified Systems

Thesis written with the financial contribution of Fondazione Bruno Kessler (FBK),

Trento (Italy)

Coordinator: Prof. Luciano Gambarini

Supervisor: Prof. Annapaola Marconi

Co-Supervisor(s): Prof. Tullio Vardanega and Prof. Anna Spagnolli

Ph.D. Student: Reza Khoshkangini

University of Padua

Brain Mind and Computer Science (BMCS)

Doctorate Degree in Computer Science

Curriculum in Computer Science

Personalized Game Content Generation and Recommendation for Gamified Systems

Candidate:

Reza Khoshkangini

University of Padua, Padua, (Italy)

&

Fondazione Bruno Kessler (FBK), Trento (Italy)

Supervisors:

Dr. Annapaola Marconi

Fondazione Bruno Kessler (FBK), Trento (Italy)

Prof. Tullio Vardanega

University of Padua, Padua, (Italy)

Co-supervisor:

Prof. Anna Spagnolli

University of Padua, Padua, (Italy)

Abstract

Gamification, that is, the usage of game content in non-game contexts, has been successfully employed in several application domains to foster engagement, as well as to influence the behavior of end users. Although gamification is often effective in inducing behavioral changes in citizens, the difficulty in retaining players and sustaining the acquired behavior over time, shows some limitations of this technology. That is especially unfortunate, because changing players' demeanor (which have been shaped for a long time), cannot be immediately internalized; rather, the gamification incentive must be reinforced to lead to stabilization. This issue could be sourced from utilizing static game content and a one-size-fits-all strategy in generating the content during the game. This reveals the need for dynamic personalization over the course of the game.

Our research hypothesis is that we can overcome these limitations with Procedural Content Generation (PCG) of playable units that appeal to each individual player and make her user experience more varied and compelling.

In this thesis, we propose a deep, large and long solution, deployed in two main phases of *Design* and *Integration* to tackle these limitations. To support the former phase, we present a "PCG and Recommender system" to automate the generation and recommendation of playable units, named "Challenges", which are *Personalized* and *Contextualized* on the basis of players' preferences, skills, etc., and the game ulterior objectives. To this end, we develop a multi-layered framework to generate the personalized game content to be assigned and recommended to the players involved in the gamified system. To support the latter phase, we integrate two modules into the system including Machine Learning (ML) and Player Modeling, in order to optimize the challenge selection process and learning players' behavior to further improve the personalization, by deriving the style of the player, respectively.

We have carried out the implementation and evaluation of the proposed framework and its integration in two different contexts. First, we assess our Automatic Procedural Content Generation and Recommendation (APCGR)

system within a large-scale and long-running open field experiment promoting sustainable urban mobility that lasted twelve weeks and involved more than 400 active players. Then, we implement the "Player Modeling" module (in the integration phase) in an educational interactive game domain to assess the performance of the proposed play style extraction approach.

The contributions of this dissertation are a first step toward the application of machine learning in automating the procedural content generation and recommendation in gamification systems.

Acknowledgment

Firstly, I would like to express my sincere gratitude to my advisors Dr. Annapaola Marconi and Prof. Tullio Vardanega for their continued support throughout my Ph.D project, for their motivation, patience, and immense knowledge. Their guidance helped me from my first joining University of Padova and DAS unit until the writing of this dissertation. Your advice on both research as well as on my career has been invaluable.

Secondly, I would like to extend my gratitude to my family: my parents, sister, brother and wife for their comforting support of my study in particular my PhD and for all of the sacrifices that they made to keep me going. Your prayer for me was what sustained me thus far.

Beside my advisors, I would like to thank to Dr. Anna Spagnolli, for her encouragements as well as support during my PhD program.

My sincere thanks also to Dr. Marco Pistore and Paolo Traverso, who provided me an opportunity to join their units and smart community in Fondazione Bruno Kessler (FBK), and also to implement this research to their project in the real domain. Without their precious support my research would have not been possible.

I thank Dr. Giuseppe Valetto, who was my co-advisor in FBK for the days we were working together before deadlines, and all the fun we have had in the last three years. Also I thank my friends in Padova and Trento.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

The term Persuasive Technology (PT) refers to the design and the use of technologies that encourage people/citizens to make an effort to change their habits [1]. For example, in the marketing context, advertisers on social media networks (e.g., Facebook[1], Instagram[2]) use announcements and advertisements to attract users to purchase specific products. This technology can find application not only in marketing, but in many other circumstances of our daily lives, from our physical and social activity to eating [2–5], from teaching in school and driving style (e.g., to obey the speed limit on highways) to the use of utilities at home [6–8]. Although the developers of PT can reach to their goals in various manners, *Gamification* has recently gained the attention of researchers and industry, as a very attractive means.

Gamification, as a persuasive technology and interactive strategy has been successfully applied to different aspects of our everyday life [9, 10] and implemented in numerous application domains. As Deterding et al. defined in [11], gamification is:

> The use of game design elements in non-game contexts.

In other words, gamification is the concept of operating game mechanics (such as points, levels, quests, bonus, missions, etc. [12]), and designing game-

[1] www.facebook.com
[2] www.instagram.com

ful strategies in a non-entertainment context to incentivize and encourage people/players to overcome the real-world obstacles [11].

Significant efforts have been undertaken in last few years among researchers as well as practitioners to understand how interactive technologies can be leveraged to encourage people to change or take particular habits [13], share information in their networks [14], settle sustainability elements like energy consumption and safety [15], and have more sustainable life styles [16]; to name just a few [17–19].

While gamification applications that promote sustainable behaviors have shown a remarkable impact in a number of critical Smart City domains, including recycling [20], energy conservation [21, 22], and personal mobility [23, 24], a shortcoming that is often observed is that their effects tend to diminish and taper off in the medium to long term [25,26]. People usually have a positive inclination to improve and change their negative daily behaviors, but they often find it arduous to encourage themselves to engage in the beneficial behaviours permanently. That is especially unfortunate, because changing players' behavior cannot be immediately internalized; rather, it must be reinforced long enough to successfully stabilize new sustainable habits [27].

Although achievements from empirical studies highlighted that game elements play a key role in encouraging people to reach the gamified systems' goals [28–30], the lack of a system that effectively and dynamically interacts with people to keep them engaged in the gamified system has intensified this issue.

This issue can be sourced from two common errors: the dominance of predefined game element that are considered at design time, and a one-size-fits-all strategy in generating game content during the game.

The former points to the static elements of the game, which are designed before launch and use. Static ad-hoc tailoring might be doable for a short and small size of gamification [2, 31], but the scaling challenge dominates when providers or designers need to deal with large gamification campaigns. While feasible in principle, the process is costly (time-consuming, exposed to human errors), and cannot be sustained in the long term. This issue clearly reveals the need to move from static to dynamic content generation in the context of gamification. In this context, similarly to what happens for entertainment games, artificial intelligence (AI) techniques can be extremely helpful. The latter problem refers to the need for personalization, which is critically important since players are distinct, they have different abilities, personalities, skills, etc. so that tailoring the game to a group of players would cause boredom or frustration for individual players.

With attention to the research challenges that relate to the two issues mentioned above, we plan to defeat and address those issues, by developing a technology solution that enables the dynamic generation and personalization of game content in a large scale gamification domain.

In general, the work presented in this dissertation presents an effective solution to generate tailored game content dynamically. This work was motivated by the following research question:

- ***How can we construct an automatic framework that effectively, efficiently and dynamically provides personalized game content that can boost players' engagement, as good as manually content generation in a gamified system?***

1.2 Vision

In order to answer the above question, we envision a solution to automate the generation of highly personalized game content. More specifically, we demonstrate the idea of designing and integrating multiple strategies for automating and tailoring challenges to each individual player. Challenges are units of playable content including a demanding goal that a player should achieve – under temporal or other constraints – in exchange for an in–game prize or reward.

The work presented in this dissertation ultimately addresses the challenge of motivating and helping citizens to change their behavior to adopt a more sustainable and healthy lifestyle. In other words, we want to help to improve citizens' habits by developing an interactive framework in context of "Gamification" that helps to impel citizens to re-evaluate their traditional habits (which have been adopted for a long duration) and encourage them to improve it. More specifically, this dissertation presents the results we have achieved for automating the generation of game content (particularly challenges) by designing an automatic Procedural Content Generation (PCG) framework. PCG techniques are frequently used in contemporary electronic games to computationally generate a wide variety of game elements in order to enhance the user experience by increasing game diversity and keeping players engaged and interested by adapting the game to the personal preferences, abilities and style of each individual player [32].

Our system produces challenges that are personalized (based on the preferences, the past game history and performances of each player) and contextualized (based on the current state of the player in the game and her game objectives). Moreover, when recommending challenges, the system takes into account game objectives, which are administered by the designer (in our case, the administration of the Smart City). In this way, we can incentivize specific citizens' behavior that is in line with the game objectives such that these objectives can dynamically change during the game.

To arrive at the results presented in this dissertation, we followed the three-step approach illustrated in Figure 1.1:

Thorough review of the state of the art to determine the possible ingredient of the required solution "Phase 1"; Designing our own novel solution, implementation and experimental evaluation “Phase 2”; Integration, deployment and experimental evaluation “Phase 3”.

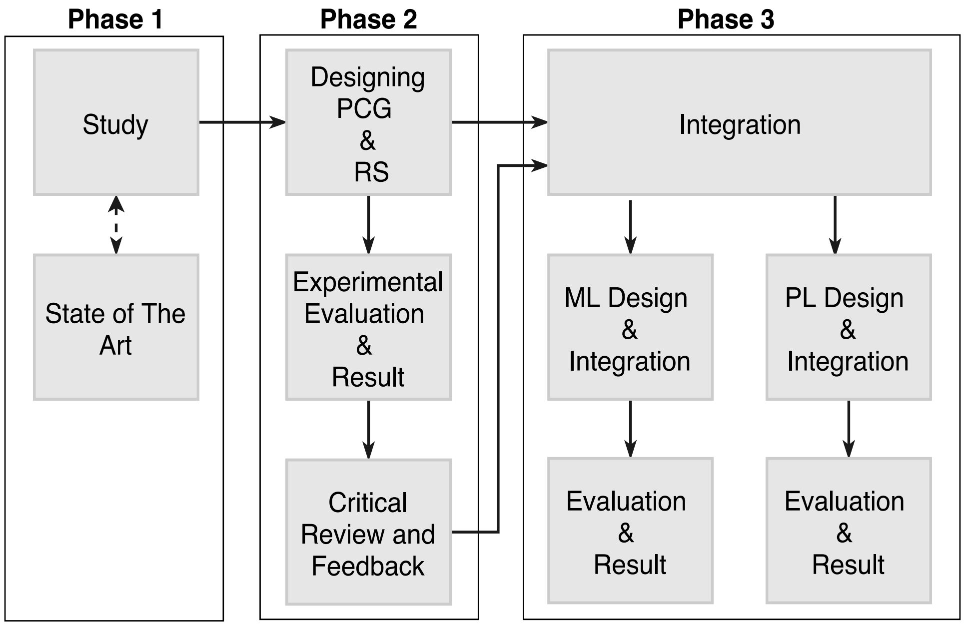


Figure 1.1: The three steps of our research project.

- **Phase 1:** The first phase involved the assessment of existing studies, which are close to our investigation to overcome such problems utilizing various techniques and strategies in different contexts. In addition, the user study that was empirically collected from the players (reported in [30]) enabled us to target the “challenges”, as one of the effective game content, to be personalized that can positively influence players’ motivation for healthy behavioral change.

- **Phase 2:** In this phase, we designed and implemented a novel solution that integrates the Procedural Content Generation (PCG) [33] strategy and Recommender system [34]. The vision behind our design was to introduce forms of PCG in gamification applications aimed at automating the generation and recommendation of game content –*Challenges*– to support and enhance the engagement of players in the gamified system. In this phase, we evaluated the proposed PCG framework in a large scale gamification campaign. Although the results demonstrated the validity of the approach, collected feedback led to design further modules with the aim of improving the peformance of the system.

- **Phase 3:** This phase involves integrating two new modules into the proposed PCG system to enhance the performance of the approach. Based on the proposed system's construct, two modules have formed and integrated into the system, with the aim of optimizing the performance of the approach. The former is "Machine Learning (ML)" that constructed to optimize game content assignment. The latter is "Play Style Recognition (PL)", which aims at extracting players' play styles to advance personalization in the gamified system. Phase 2 and 3 are described in details in Chapter 3.

Our solution, as the concatenation of multiple strategies, has been implemented and evaluated with two important concerns in mind: *Effectiveness* and *Efficiency*, which are reflected in seven objectives defined and detailed through the chapters.

- **Effectiveness:** In this dissertation, the aspect of effectiveness are interpreted in three different sub-aspects in evaluating our proposed solution. First, effectiveness refers to the performance of the game content, which are generated, personalized and recommended by our proposed framework (APCGR) vs. the game content that are administered manually, in motivating players to improve their behavior in the gamified system. Secondly, in the Integration phase (phase 3)–particularly in evaluating machine learning module– it points to the performance of the ML module to predict the success rate of the challenges. Finally, this aspect –particularly in evaluating player modeling module in phase 3– refers to the performance of the module in extracting players' play style in the game.

- **Efficiency:** Similar to the previous aspect, efficiency is interpreted differently in both the Design and Integration phases to evaluate the proposed approach. In the third phase, efficiency talks to whether our system assigns commensurate and balanced rewards for the challenges it automatically proposes to players vs. the manual generation by the expert judgments. This aspect in the integration phase –specially evaluating machine learning module– points to the time that ML module needs to complete the whole ML process.

The high level conceptual view of the proposed framework is illustrated in Figure 1.2, where the two points A and B represent the generation of game content and personalization process, respectively. Point C illustrates the recommendation of the personalized game content, and D refers to the learning process –learning players' behaviors in the game. The result of this process could be the input of the content generation for the next iteration to optimize the recommendation process, which are described in Chapter 3.3.

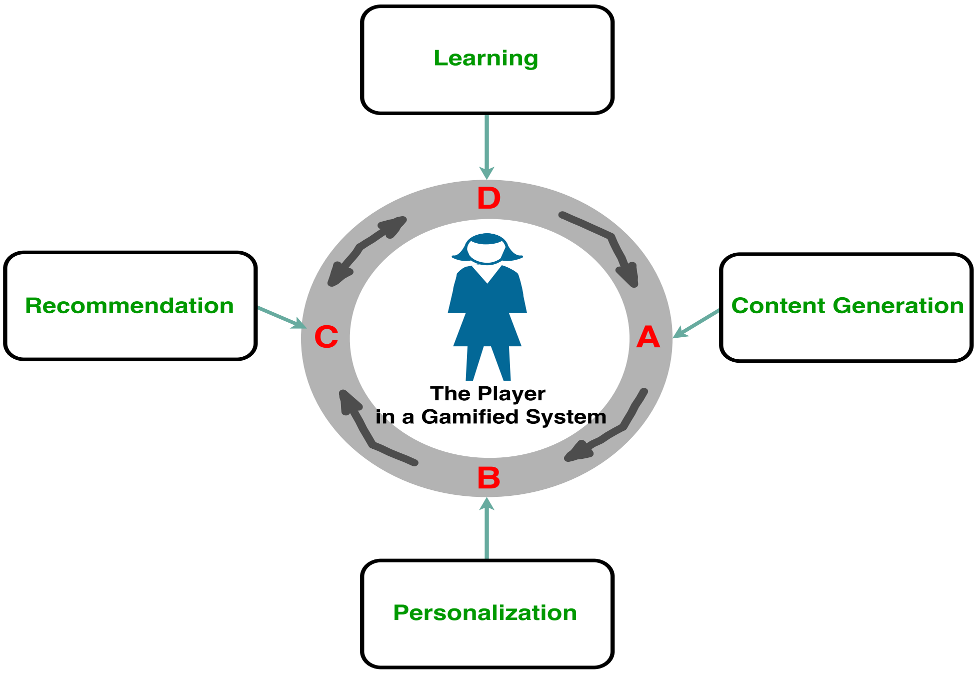


Figure 1.2: A General Overview of the Proposed Framework in Gamification.

1.3 Research Challenges

In this dissertation, the most significant challenges that we have encountered to apply our solution in dealing with the highlighted problems can be divided into Dynamicity in selection process and Validation, which are described in the following:

- **Dynamicity:** Dynamicity imposes a significant challenge in selection process in our proposed approach. It refers to the concept of *Change* in e.g., players' preferences or game objectives over time. We faced this challenge in two phases of our solution including "designing and integration".

  First, it appeared in constructing the Filtering & Sorting module in the main framework, since the algorithm used at the core of this module had to deal with two different aspects: player's preference and objectives of the game. Dynamicity of these two aspects challenge game personalization in selecting the suitable "Challenges" in order for the players to be recommended. This is due to those cases, where the system based on the game objectives, tries to push a task that contradicts the players' preferences. To overcome this difficulty, we constructed a multi-criteria algorithm presented in Chapter 3.2 to find a sweet spot between the player's preference and the objectives of the gamified system.

Second, we encountered this issue in applying player modeling technique in gamification context. As players do not follow a particular play style during the game and they change their play style over time, such dynamicity turns the personalization task into a challenging issue. To handle this issue we built a module introduced in Chapter 3.4 capable of recognizing this dynamicity "on-the-fly" that can be utilized for game adaptation over time.

- **Validation:** Evaluating the proposed approach in a real-world scenario was a challenging task to prove the effectiveness of the designed system and its' integration. We implemented the proposed framework in a gamification campaign in the context of urban mobility system through the Trento Play&Go App[3] in Trento city (Italy) in 2016. It was specially arduous, because the framework was examined in a large scale and open-field gamified system.

Another challenge we faced is the shortage of individual player's data. Although, there was a considerable numbers of active players in the game, the system endured by having limited individual player's challenge history to model the player's feedback w.r.t the given challenges. Player's feedback refers to the ability of players to handle and complete the given challenges. To tackle this problem, we designed a ML module presented in Chapter 3.3 capable of modeling players' habit (with minimum numbers of samples and features).

1.4 Contributions

This dissertation shows that Automation and Personalization of game content are not only feasible, but it also indicates that the proposed approach is more effective and efficient in changing players behavior compared to the manual generation of challenges in a large scale gamification study. In particular, in this study we made three main contributions to the field of gamification and design of the interactive framework to boost behavior change, which are listed in the following:

- We have developed a framework that automatically generates personalized game content in context of gamification –in general– and applied to Urban Gamified Mobility system in a Smart City. We have validated the effectiveness and efficiency of the proposed framework through a large-scale and long-running gamification campaign by comparing the challenges that are automatically generated and personalized by our APCGR system with the challenges which are manually administered by the expert judgments (designers who are expertise in this field).

  This contribution is corroborated by the following papers:

[3] http://www.smartcommunitylab.it/apps/viaggia-trento-playgo/

  - ⋆ Reza Khoshkangini, Valetto Giuseppe, Annapaola Marconi and Marco Pistore, “Automatic Generation and Recommendation of Personalized Challenges for Gamification”. (Submitted to IEEE Transactions on Computational Intelligence and AI in Games (TCI-AIG)).
  - ⋆ Reza Khoshkangini, Giuseppe Valetto and Annapaola Marconni, “Generating Personalized Challenges to Enhance the Persuasive Power of Gamification”. Personalized Persuasive Technology, pp. 70-83. CEUR-WS.org, 2017.

- We have devised a machine learning module (ML module) that supports the prediction of probability of challenge success to optimize the selection and recommendation process. This potentially can increase the challenge success rate by the players that contributes players’ engagement and behavioral change in the gamified system. This contribution resulted in the following paper:

  - ⋆ Reza Khoshkangini, Annapaola Marconni and Giuseppe Valetto, “Machine Learning for Personalized Challenges in a Gamified Sustainable Mobility Scenario”. Annual Symposium on Computer-Human Interaction in Play (CHIPLAY), pp. 361-368, 2017.

- We have designed a player modeling module to automatically and dynamically extract and model the play style of the players in gamification. This play style extraction module can contribute to game adaptation in the gamified system. This contribution elaborates on the following paper:

  - ⋆ Reza Khoshkangini, Santiago Ontanon and Annapaola Marconi “Dynamically Extracting Play Style in Games”. (Accepted to GameOn 2018, UK).

  A possible extension this contribution is elaborated, on the following paper:

  - ⋆ Reza Khoshkangini, Zaffar Heidari, Enrica Loria and Annapaola Marconi “Semi-Supervised Learning to Extract Players’ Play Style in Gamification”, (Submitted to Annual Symposium on Computer-Human Interaction in Play (CHIPLAY) 2018).

  Other publications that supported the work reported in this dissertation are listed in the following:

  - ⋆ Reza Khoshkangini, Dinh Van Tran, Maria Silvia Pinni and Francesca Rossi “Constructing and Learning Users’ Conditional Behavior Via Bayesian Network”, (Accepted to SIR workshop 2018).

⋆ Reza Khoshkangini, Maria Silvia Pini and Francesca Rossi "A Self-Adaptive Context-Aware Group Recommender System", In Conference of the Italian Association for Artificial Intelligence (AI*IA), pp. 250-265. Springer, 2016.

⋆ Reza Khoshkangini, Maria Silvia Pini and Francesca Rossi "A Design of Context-Aware Framework for Conditional Preferences of Group of Users", In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 97-112. Springer, 2016.

The contributions of this dissertation have the potential for making the end-to-end generation of personalized game content process easier and flexible for each player, especially for a large-scale gamification campaign in the smart city.

1.5 Organization of this Dissertation

The remainder of this dissertation is organized as follows:

Chapter 2: Research Background: presents the state of the art, in which we introduce the notion of *Procedural Content Generation (PCG)* and its components, and the diversity of Recommender Systems (RSs). Since the main direction of our research is constructed based on these two research areas, we also provide a review of recommender systems, which are investigated in the domain of game and gamification. Furthermore, we report the use of *Machine Learning* in Game and Gamification, and conclude by reporting the existing results on *Player Modeling*. This Chapter presents a summary of existing works in the field that guided us to develop our APCGR system in the context of gamification. Finally, we introduce the components of the proposed APCGR framework.

Chapter 3: Design and Integration: this Chapter basically demonstrates the design of the proposed framework and integration of its' complements including Machine Learning and Play Style Extraction modules. In this Chapter we first present the key characteristics of our Gamification platform which built for smart cities. We overview the characteristics of our solution as the part of the broader view of the gamified platform that supports the automation of personalizing game content in gamification. Then, Section 3.2, presents the Challenge Model and structure of our Automatic Procedural Content Generation and Recommendation System. In this section, we detail the framework's modules including: how challenges are generated by the *Challenge Generation* module, valuated using the *Challenge Valuator* module, and finally assigned to each individual player by the *Filtering & Sorting* module.

Section 3.3, illustrates the construction of the *Machine Learning* module that has been integrated within the APCGR framework to optimize the performances of *Filtering* & *Sorting* module. Finally, in Section 3.4, we introduce *Player Modeling* concept and its' background in digital games. Then, we explain our proposed *Score-based* module to construct the play style of the player in a gamified system.

Chapter 4: Implementation, Evaluation and Discussion: This Chapter presents the implementation and evaluation of the proposed APCGR and its' integrated modules in two different gamification studies. Chapter 4 starts by describing Play&Go scenario, its' specification and how citizens could participate to the gamified system. Then, in Section 4.2, study setup and evaluation of APCGR are detailed. In the following, the objectives of the evaluation and results are presented, respectively. We evaluated the framework that whether the designed approach is effective and efficient for promoting behavioral change. In Section 4.5, the evaluation of Machine Learning is described and constructed aimed to enhance the challenge selection performance in Filtering & Sorting module. Then, e-learning gamified scenario (educational game-based learning domain) is introduced in Section 4.8 that followed by player modeling experimental setup and evaluation objectives. Finally, the results of our play style recognition module are reported. This Chapter concludes by discussing the achievements from the implementation of APCGR, Machine Learning and Play Style Extraction, respectively.

Chapter 5: Conclusion and Future Works: This Chapter finalizes the dissertation. First, it presents the summary of our research by highlighting the research problems in the context of gamification and our proposed solution to handle them. Then, we overview our findings and relate them back to the evaluation objectives which were defined based on effectiveness and efficiency. Then, we introduce our future investigation and specify the possible research directions that the proposed framework could be extended and applied.

Chapter 2

Research Background

This study builds upon prior work in the setting of 1) Recommender Systems, and 2) Procedural Content Generation (PCG) in the context of Gamification. Procedural Content Generation is being shaped to codify and systemize the generation of game content without recourse to manual generation. A Recommender system or recommendation system falls in the Information Filtering area, with the aim to predict and recommend services/goods contingent on users' preferences. Integrating these two ingredients enables the use of PCG in gamified contexts. Having in mind the above two important research areas, we designed an automatic PCG framework in the context of gamification. Before designing our solution, in this Chapter, we review the state of the art in this domain and position our vision within it.

In the following sections, we report the current studies in Procedural Content Generation (PCG) and Machine Learning approaches in the context of game and gamification. Then, we detail the concept of Recommender System (RS) and its' multiple derivatives in the domains of interest. Subsequently, we discuss how we integrated the modules (machine learning, for optimizing challenge selection process, and player modeling, for extracting players play styled over the course of the game) into our APCGR framework.

2.1 Procedural Content Generation & Machine Learning

Procedural Content Generation (PCG) techniques are developed to formulate and automate the construction of Game Design Elements (GDEs) to be delivered in electronic games. Elements may vary widely [33], from sounds and textures, to buildings, maps and whole game level layouts, to items, equipment and other virtual goods, all the way to *playable units of content* such as riddles to solve, obstacles to overcome, encounters with non-playing characters, challenges, missions and quests to complete, or even the entire storyline [35, 36].

Advancements in PCG have largely been driven by the need to contain the time and cost for developing diversified game content at scale, while improving the experience of players, who dislike excessive repetition. PCG has also been used to automatically adapt the game experience, using Machine Learning, to the characteristics of each player [32, 37]. For example, Zook et al. have used PCG to tailor the game play to a model that captures the player's current know–how and performance, and have applied to both missions in a role–playing fantasy game, and training scenarios in a military serious game [38, 39]. Often, PCG is used to automatically adjust the difficulty of playable units of content, and personalize the balance between player's satisfaction and challenge over time according to the concept of "flow" (see Figure 2.1). Flow is recognized as a major factor for fun and retention [40–43], if challenges are well-balanced to the player's skill during the game *over time*.

The concept of flow was introduced by Csikszentmihalyi [40] for the first time (mid-1970), that refers to players' experience with highest engagement and fulfillment. Recalling the requirements of the concept, Csikszentmihalyi highlighted eight main elements of flow [41]:

- Clear goals,
- Balance between skill/ability and difficulty of the challenge,
- A margin of action and awareness,
- High level of concentration on the specified task,
- Prompt feedback,
- A loss of self consciousness,
- Feeling of control for the given task,
- Transformation of time.

Having in mind the concept of flow, many attempts have been conducted either manually or automatically to adapt the difficulty tasks/levels to the player's skill, with tangible increase in the player's engagement in the game. For instance, Seth et al. [44] propose to utilize Player Rating Systems (e.g., Elo is a method to calculate players' relative skills in competitor games like *Chess* [45]) to select and sequence tasks as a framework in balancing the difficulty level in the context of Human Computation Games (HCGs). Togelius et al. [46] generated race tracks according to the acquired driving models, in order to improve the entertainment value of an auto racing game. Similarly, Lora et al. in [47] introduce an approach to dynamically change the difficulty level of the *Tetris* game. Such personalization of game content results a great potential at hindering players to be frustrated and of keeping them engaged in the game [48, 49].

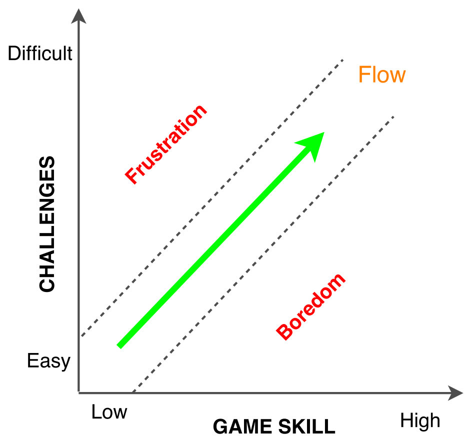


Figure 2.1: The Conceptual View of Flow in Game

The power of computational techniques adopted in PCG has been quickly grown with the demand for increasingly sophisticated generated content in contemporary electronic games. All effective PCG contains two facets: *exploration*, which generates a potentially very large and diverse number of options, and *selection*, which picks among the generated options those that are fittest for the purpose. For the former, some of the prevalent approaches are search–based techniques (e.g., genetic algorithms) [50]; as well as planning techniques, in particular when creating new gameplay scenarios or storylines [35, 36]. For the latter, some PCG approaches leverage insights from Recommender Systems.

State-of-the-art experiments have shown that the generation of game content has become increasingly prominent in the domain of *Electronic Games* in order to decline the production cost and time, and improve re–playability and tailor the game content to each individual player. While PCG and machine learning techniques have far less commonly been used in Gamification. For instance, Biptiste et. al in [51] introduce a *theoretical* adaptive and generic architecture to engage players in an existing web-learning environment. Their approach trace player's data and integrate multiple-game mechanics in order to adapt the game with the user's characteristics.

In the context of "eco-driving", a study that looks in the same direction (as this dissertation) is provided recently by Di Lena et al. [31]. They built a prototype that gathers the driver's braking style data to make a model to predict the saving energy pattern using a machine learning technique. Then, they exploit it at fostering driver's behavior in a gamified way. Another study has been recently done by Rodrigues et al. [52] in the education domain "Mentalmath"[1]. In this study, a template-based approach is introduced to generate a limited set of problems and scenarios in basis of players' math skills. In this experiment, at the end of each level of the game (MentalMath) participants (40 elementary students from a Brazilian school) were given various types of math problems (containing basic operations such as subtraction, divisions, additions and multiplications) to be solved.

The privilege of using machine learning techniques in gamification –in particular, PCG– relies on the capability of understanding players' real behavior and model in responding the recommended challenges and exploiting this model to further personalize the game content – especially the challenges in our domain– for a particular player.

The studies we have reported not only demonstrate the recent attentions in exploiting PCG techniques to foster players' experiences in digital games, but also help to discern the richness of these techniques to be expanded and advanced in another context of game (Gamification). Advancing this technique with a recommender system may have a positive influence on personalization that can enhance the players' engagement in gamification.

In the following section, we first discuss the notion of Recommender Systems and anatomize their classifications (we highlight the most important kind of recommender systems, which are widely used in the research community) such as *Content Based Filtering, Collaborating Filtering, Hybrid Systems* and *Context-Aware Recommender Systems.* Finally, we review the use of these types of systems in PCG, electronic game and gamification domains.

2.2 Recommender Systems

Generally, all Recommender Systems (RSs), starting from a set of given inputs make a pattern and then search for a specific service(s) in order to meet the user's demand(s). Recommender Systems are triangulated into three main categories of *Content-Based Filtering (CB), Collaborative Filtering (CF)*, and *Hybrid* (CF and CB combined) [34], which we discuss below, individually.

2.2.1 Content Based, Collaborating Filtering (CB) and Hybrid Systems

Content-based systems also known as "cognitive filtering" try to recommend items based on the similarity between user's past activities (from her profile), and the content of service (that could be an item or a product to recommend) [53]. User's profile contains her past activities in the application domain e.g., past selections, or the content that she has browsed before, which

[1] Mentalmath is a game based tool that was developed to foster kids' math skill. http://flexmath.ck12.org/mental-math

are characterized with similar features. Usually, products are represented and characterized by several features. For instance, price, size of the item, type of the item (e.g., computer accessories, etc.), to name a few. Generally, content based techniques are constructed based on users' information, while collaborating systems work based on other users' information who present some similarities. Content based systems have been widely used in various application domains from product guide selection [54] to movie, music recommendation, from health-care [55] and e-commerce [56] to education [57].

In contrast, collaborating systems act quite differently, since in these approaches, the recommender system suggests products/services that *other similar users* have liked, selected or used before. In other word, in these types of recommendations the similarity among users are computed in order for the particular user to propose new products, rather than the similarity between items [53]. Notably, *K-nearest neighbor (KNN)* algorithm is one of the most common algorithms used in data mining –in general– and recommender systems –in particular. These sort of recommender systems operate based on similarity functions including Cosine Similarity, Hamming Distance, Euclidean Distance and others [58, 59], as well as Correlation Pearrson measure that has been widely used in content based system to find the correlation between users' ratings [60].

Although, the above RSs are massively used and implemented in several application domains, many approaches have been developed to integrate these kinds of recommendation systems (using various techniques [61–63]), which are known as *Hybrid Systems* [64, 65]. Hybrid Systems have been developed to gain a peak performance with fewer issues of any individual technique to maximize the users' satisfaction in the context of recommender systems. The conceptual view of the Hybrid Systems is illustrated in Figure 2.2.

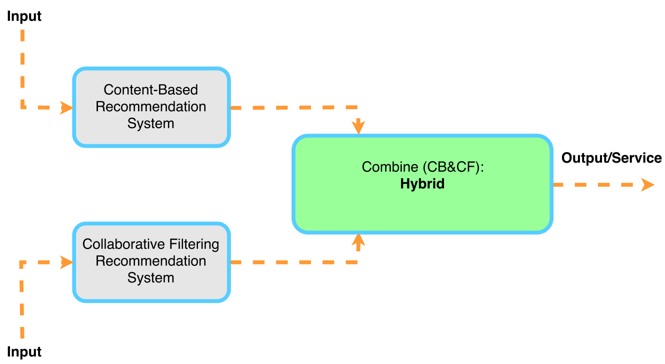


Figure 2.2: The Conceptual View of the Hybrid Recommender Systems.

2.2.2 Context-Aware Recommender System

The newest generation of RSs, called *Context-Aware Recommender Systems (CARSs)*, builds on *user, item* and the *context of the user.* Context is defined in [66] as:

> Any information that can be used to characterize the situation of an entity or a domain.

Such systems are extensively employed in dynamic domains (in these domains, context changes over time and may therefore influence users' preferences) to optimize the prediction of user's preference. From the workflow structure point of view, CARSs are categorized into two main methods:

- *Recommendation via context driven querying and search*; The implementation of these kinds of systems has been seen in the domain of mobile, tourist guide and movie recommendation [67, 68]. Such systems use context (e.g., interest, local time, location, etc.), to query or search the user's preferences in a specific repository of resources (e.g., movie or restaurant [67]). Thereafter, the systems try to provide the best service to users.

- *Recommendation via contextual preferences elicitation and estimation*; In contrast with the first method, in which the system employs the current users' context as a query or a preference to search for some services, the second method tries to model and learn users' preferences by observing users' interactions with the system.

Basically, both models try to recommend a set of services to a user or a group of users considering the context. For instance, Simen et al. in [69] introduce a prototype of group CARS in a *Concert* application. In this study, they consider only *Time* and *Location* as the *context* in order for customizing the recommendation to fulfill users' demand, in which different CF algorithms such as *K-nearest Neighbor*, *Matrix Factorization* and *Hybrid* methods are implemented.

In order to maximize users' satisfaction in recommendation context, more features have been used to model and characterize users' preferences in personalization process. Palmisano et al. [70] introduced a hierarchy of contextual information with multi-dimensions in their system, where each dimension could have a numbers of sub-dimensions including time, weather, location, etc. Similarly, in [71] every feature is defined as a dimension, such that a rating function R is used to specify how likely a user u prefers item i at time t.

Acknowledging the the diversity of recommender systems that we surveyed above, and borrowing from the taxonomy in [72], our proposed PCG framework can be categorized as a type of Context-aware Recommender System that is specifically constructed for any gamification scenario. Current game status, and contextual information such as time, day, etc. define the context in this recommendation domain. In addition, we use content-based approach in order to personalize a part of users experience.

2.2.3 Recommender Systems in Game and Gamification

In the domain of electronic games, recommender systems (RSs) have been most often used to recommend new games to fulfill players' inclinations and game play. For instance,

Sifa et al. [73] proposed two different techniques of Archetypal Analysis in order to recommend a new game to players in basis of their activities during the game. Likewise, Skocir and his colleagues in [74] introduce a Multi-Agent Recommendation System that uses a number of different parameters elicited during the game play, to propose new mobile games that best suit the player's style and skill.

Similarly to our research line, a recommender system is used in combination with PCG [75], aimed to propose the procedurally generated content (e.g., map) by taking into account the individual player's characteristics in a game platform. Subsequently, Harrison et al. in [76] proposed a Collaborative Filtering system, based on the player's past quests to propose the achievements that a given player may enjoy most taking on next, for the *World of Warcraft* MMORPG.

In the context of automating game content generation, Andersen et al. in [77] introduced a theoretical trace-based framework that was demonstrated on two domains (elementary and middle school mathematics and well-known puzzle game *Refraction*). They aimed to estimate the difficulty of the procedural problem by mining the features that were traced in the game. Later on, a large-scale experiment is evaluated on the same puzzle game *Refraction* by proposing a framework that automates the generation of "Level Progression" aimed to enhance players' engagement in the game [78].

In the gamification context, the closest work is *PHESS* [79], which provides recommendations on sustainability measures based on monitoring and accumulating user's history, in the domain of home energy conservation. Although, those measures are not automatically generated and taken from a limited repertoire of static alternatives.

As briefly reported above, game adaptation is implemented not only on game elements "either statically or dynamically" during the game, but it can be exploited to tailor different aspects of the game. *Player Modeling* has shown its great potential to maximize the engagement of players in the context of electronic game [80]. The final section of this Chapter focuses on player modeling in game and reports a number of studies, which have been done in the domain of gamifiaction.

2.3 Player Modeling in Game and Gamification

The concept of player modeling refers to the study of computational models that has been widely used in the domain of electronic games for customizing game content to players preferences, traits, abilities and personalities, in order to enhance players' engagement [80–82]. In addition, players could be distinct by play style, motivation and pleasure that then will be categorized as player typologies. The means by which these various player types are obtained is integrated by statistical methods to interpretative and psychological approaches. For instance, a well-known player taxonomy (Bartle) was introduced on the basis of player behavior and experience recorded in a long-term game [83]. He introduced four main player's types as follows;

- Achievers: achiever players enjoy to face and beat difficult challenges who can gain more rewards and try to finish the game as fast as possible. They are problem solvers who make few errors in the game.
- Killers: they try to impose themselves on others in the scope of the game provided by virtual world. The greater they can prove drama over the others, the more pleasure the players have. For example, cheaters, trolls and hackers could belong in this type.
- Socializers: as the name/type implies, they like more to have relation to the other players rather than playing the game. They are interested to share their knowledge to others or be involved in the game community.
- Explorers: they are curious to explore the game by visiting the whole game content and objects. They usually finer analyze the details of game content, short-cuts with a significant amount of attention.

Later, Nick Yee conducted a study of Massively Multiplayer Online Role-playing Games (MMORPG), and collected more data through questionnaires [84]. In this long term study, Yee develops Bartle's model and proposes that player's motivation can be divided into three main categories and ten sub-categories as follows (cited by [85] as well):

- Achievement: Mechanics, Competition and Advancement,
- Social: Relationship, Socializing and Teamwork,
- Immersion: Role playing, Escapism, Discovery and Customization.

Similarly, an empirical study on player modeling was investigated by Kallio et al. [86]. In this work, a gaming Mentality Heuristics model is introduced containing nine profiles of player's types in three different categories including Social Mentality, Casual Mentality and Committed Mentality in the context of digital games. These categories are constructed on the basis of regularity, length and social context of the game as follows;

- *Social Mentality* contains gamin with mates, gaming with kids and gaming for company;
- *Casual Mentality* accommodates filling gaps, killing time, and relaxing;
- *Committed Mentality* that covers entertainment, having fun and immersion profiles.

Over the last decades, potential research studies have loomed on personalization and player modeling in the context of the digital game.

Since player type represents the actual player's substrate, modeling and predicting player's preferences in the game may help to augment the player's engagement for a long term. Indeed, the play style delivers an understandable pattern that has a remarkable impact on adapting the game based upon players' behavior [87]. In addition, recommending a new personalized game content, product/game to players [88] could be the options that can act as a booster to keep the players immersed in the game. Hence, the essential task before game adaptation towards the players' play styles, is to model and extract the play style during the game.

Many approaches focus on analyzing players' behavior throughout the game while they play, by taking into account game metrics to obtain and characterize play style of the player [89–91]. Within these studies, some constructed the play style in the context of game exploiting feature selection and segmentation. For instance, one typified work has been introduced by Drachen et al. [92]. In this work a clustering approach is used to characterize players' play style taking into account game metrics such as number of death, time completion, etc.

Another example of a large-scale empirical experiment on modeling play styles that we can refer has been conducted by Thurau et al. in [93]. In this study, an unsupervised learning algorithm via Emergent Self-Organizing Maps (ESOMs) is applied to construct play style for the popular game entitle Tomb Rider: Under-world (TRU). Segmenting the whole game period into multiple time-intervals (windows) is another strategy, which was employed by Bifet et al. in [94] and also Martin et al. in [95], such that the player's behavior is captured in each time-interval to characterize the play style and used to predict the style for the next time-window.

Given the studies that focus on player type in digital game context, the importance of gamelay in digital game also expanded to other context like gamification. Unlike, player type in gamification is barely the same as the well-known player type in digital game (e.g., Massively multiplayer online role-playing games (MMORPGs)) introduced by Bartle [83, 96].

Turning the player type from digital game to gamification requires a complete re-design and potential research and empirical studies.

To this end, Marczewski [97] used Bartle original players' type and his advice to shape and design its two main category of Intrinsically motivated and Extrinsically motivated players. These two main models yield eight subcategories in gamification. Intrinsic covers players' types such as Philanthropists, Achiever, Socializers and Free Spirit. Conversely, the Extrinsic type ramifies in Self Seeker, Consumers, Networkers and Exploiters[2].

Although there are research studies on player modeling that introduced and validated the various player's types in the context of gamification, most of them are limited to a particular application domain. This is the case of Hexad framework [98], BrainHex model [99], among others [100].

Under the umbrella of the above models in game and gamification, Ferro et al. [101] discussed and theoretically analyzed the relationship between player's type, personality and traits in game and gamification domain. They proposed a typology of player's type that can contribute to better inform the design of gamified system toward engaging the people in the game. Similarly, Monterrat et al. [102] proposed a model to customize gamification features (such as point, rewards, score boards, etc.) on the basis of player type "BrainHex player type". In this study, they propose an association Matrix that is constructed based on a linear relation between player's type and game features provided in the learning environment. The proposed association matrix can be shaped by expert judgment or empirical data. They showed that expert-based matrix (researcher asked to fill the matrix of weights in order to map BrainHex models to the features of the game) performs better against empirical data (which needs more data to be trained and learned).

Apart from the studies in modeling players' behavior in gamification that are mostly application-based, we have not seen a potential attention in investigating and discussing how the play styles can be automatically and dynamically identified over time. In Chapter 3.4, we discuss on the player modeling –in particular– detail how to automatically construct the player's play style. We propose an Automatic Score-based System that is able to dynamically construct players' play styles in a supervised way and we applied it in a gamified system. The system is Rule Based and contains the characteristic of the players' play style inspired from [83, 96, 103, 104].

To our knowledge, the combination of computational generation of playable content and recommender systems in gamified applications, as a strategy to increase engagement and ensure retainment of players is novel. Thus, in this study we bridge the gap of constructing a systematic generation of game content (particularly Challenge), between *Generation* and *Assignment* game content to players by exploiting PCG and RS in Gamification context.

[2] http://yukaichou.com/gamification-study/user-types-gamified-systems/

2.4 Summary

Our review of the relevant state of the art shows that the strategies we surveyed including procedural content generation, recommender systems, machine learning and player modeling, have gone a long way in the field of digital game and Gamification. For example, we have seen the widespread use of procedural content generation (PCG) and machine learning in digital games to formulate the generation of game content and model the players' preferences, skills, etc. with the aim of boosting players' motivation in the game. In contrast, PCG has been rarely exploited in gamification.

We have also seen that researchers utilized the power of recommender systems in digital games and gamification context, e.g., to recommend new game products or content, in order for the players to be more immersed in playing the game.

Moreover, we have seen the investigations in player modeling in these two contexts, mostly to show that players are distinct in playing the game and the ways to analyze and extract them. The reported studies have given us a valuable insight into recent and on-going investigations that attempted to address the problem of enhancing players' engagement for a longer term (mostly in digital games) by exploiting various strategies and techniques.

By taking into account the investigations, we come up with a novel idea of integrating all these strategies to bridge the highlighted gap including Scaling and Personalization for a large number of participant in the context of gamification. The solution presented in this doctoral dissertation utilized the powerful strategies and techniques including *Procedural Content Generation*, *Recommender System*, *Machine Learning* and *Player Modeling*.

We proposed the system "Automatic Procedural Content Generation and Recommendation", with the aim of achieving the two main objectives: i) Automating the generation of game content, and ii) Personalizing the game content for each individual player, in order to overcome the problem of static content generation and one-size-fits-all strategy in a large scale gamified system, respectively. This helps to boost the players' healthy behavioral change in gamification.

To conclude this Chapter, Table 2.1 represents and summarizes the main existing approaches in the state of the art for procedural generation of content and recommender systems, which are close to our research direction. In particular, we have divided the literature into 4 sections as follows. First, we introduced the studies on Procedural Content Generation (PCG) and the use of Machine Learning in PCG. Secondly, we reviewed the well-known classification of Recommender systems and their derivatives such as Content-based systems, Collaborative Filtering system, Hybrid systems and Context-aware recommender systems. Thereafter, we reported the investigations that used recommender systems in PCG in the electronic games and recommender systems, which are applied in the domain of gamification. Finally, the concept of player modeling and its' categories are reported in game and gamification context. Then, we highlighted a number of valuable studies that exploited player modeling concept in game and gamification for the game adaptation.

In the next Chapter, we introduce the gamification platform, its' characteristics, and objectives. Furthermore, we indicate the integration of the proposed framework in the platform, to implement and validate the effectiveness of the designed approach in Trento Play&Go gamification project.

Table 2.1: The Summary of the Related Works

Approach	**Domain**	**Scholars**	**Application**
PCG & **Machine Learning**	Game and Gamification	Mark et al. [33], Ricardo et al. [32], Karpinskyj et al. [37] Zook et al. [38, 39], Charles et al. [48], Sauvik et al. [49] Jenova et al. [105], Ben et al. [42], Togelius et al. [46, 50], Lora et al. [47], David et al. [35] and Edirlei et al. [36], Biptiste et al. [51], Ferro et al. in [101], Di Lena et al. in [31].	Generating game content in electronic games, personalizing the game based on players experience and motivating them to keep engaged in the game. Using generic adaptation in web learning, analyzing relationship between player's type, customization in eco-driving.
Recommender Systems	Content-Based Systems	Balabanovic et al. [53], Duan et al. [55], Schafer et al. [56], Nguyen et al. [57].	In healthcare, e-commerce, education, movie, etc.
	Collaborative Systems	Balabanovic et al. [53], Adomavicius et al. [58], Sarwar et al. [59] , Burke et al. [60].	Using different CF techniques such as Hamming and Euclidean, Pearson , etc. to recommend various items.
	Hybrid Systems	Burke et al. [65], Claypo et al. [63], Smyth et al. [106], Basu et al. [61], Gemiss et al. [62] and Lampropoulos et al. [64].	Researchers used various combination of the two techniques to enhance the performance of their RS to meet users requests.
	Context-aware Systems	K. Dey [66], Chihiro et al. [67], Mar et al. [68], Simen et al. [69], Palmisano et al. [70], Gediminas [71], Bobadilla [72].	Analyzing the context, tourist and movie recommendation, CARS for concert application, introduced a hierarchy of contextual information with multi-dimensions, defining a rating function to consider *user, item* and *time*, taxonomy or RS.
Recommender System in Game and Gamification	Electronic Games	Sifa et al. in [73],Skocir et al. [74], Rafell et al. [75], Harrison et al. [76], Silvia et al. [79].	Archetypal Analysis for RSs, multi agent RS, integrating RS and PCG, using collaborating systems in Word of Warcraft, home energy conservation
Player Modeling	Game & Gamification	Bartle et al. [83], Nick Yee [84], Hendrik et al. [81], Riedl and Smith in [80, 82], Josep and Paras in [107, 108], Van et al. [87], Skocir [88].	Introducing different types of player types, applying in electronic games, motivating player for a long term in EG, player modeling in e-learning domain, investigating on player's play style pattern, using player model for recommending new game.

Chapter 3

Design and Integration

This Chapter is comprised of four main sections, in which we present the gamification platform, design and integration of the proposed multi-layer framework that automates the generation and personalization of the playable units called Challenges.

The first section of the Chapter provides a notion of a Gamified System in a smart society, in which Information and Communication Technologies (ICT) are used to improve the quality of life of their citizens. we provide an overview of the Gamification Platform by pointing its components and the objectives that we plan to achieve in any gamified system. Within the platform, we present the design of our proposed framework (Section 3.1.1) and its main modules that has been developed to generate the personalized game content within the *Trento Play&Go* gamification programme.

Second section presents the *Challenge Model* that is the concatenation of multiple elements such as goal/task, constraint, difficulty, reward. Then we describe the design of the proposed Automatic Procedural Content Generation and Recommendation (APCGR). Basically, the aim of the designed PCG framework is to dynamically tailor game content (particularly Challenges) with respect to the players' skills, game status, preferences, objective of the designer, etc. during the game. These elements could be parameterized through the Challenge Generator, Challenge Valuator and Filtering & Sorting modules which we describe in the following.

In Section 3.3, we illustrate the design and integration of the Machine Learning (ML) module. The proposed ML module concerns the optimization of the challenge selection process in order to increase the success rate of the recommended challenges by the players.

Finally, in Section 3.4, we describe the design and integration of an automatic player modeling module. The proposed play style recognition approach can support the framework to adapt the game to each individual player by dynamically extracting player's behavior throughout the game. We designed a Utility Function that enables the module to extract players behavior in a supervised fashion.

3.1 Gamification Platform in Smart City

This Section provides the key characteristics of the Gamification Platform for Smart Cities [109] that has been extended with APCGR technique developed within this research dissertation. The platform has been designed and developed having in mind the following principles:

- Open-ended integration of existing IT systems and services in a Smart City, whose interactions with citizens must become part of the game as player actions;
- Support for dynamic city policies and objectives, as well as a set of extensible game design elements;
- Sustain healthy behavioral change through long-running games that can keep players engaged in the system.

The architecture of the Gamification Platform supports the entire game life-cycle such as design, deployment, execution and analysis of games, which is organized in three layers as follows: Gamification Enablers, Gamification Services, and Gamification Front-end (see Figure 3.1 for an overview). In the following sub-sections, we describe each layer functionalities.

3.1.1 Gamification Enablers

The *Gamification Enablers* layer embeds for the basic functionality related to the design, deployment (*Game Definition*), execution of games (*Gamification Engine*), and its integration with Smart Cities IT systems (*Wrapping*).

This layer is also equipped with the *Challenge Generator* component, that is responsible for the automatic generation of challenges in the gamified system.

The design of the APCGR framework and its integration within the Gamification Platform is depicted in Figure 3.1, in which the proposed framework is designed between Game definition module that injects game model (specially challenge model) to framework, and Gamification engine that executes the whole game procedures and their functionalities.

Overview of The Designed APCGR Framework

The framework is constructed by three main modules: *Challenge Generator*, *Challenge Valuator* and *Filtering & Sorting*, and later we built up the propose framework by integrating a *Machine Learning* and a *Player Modeling* modules. We overview the actions and procedures implemented by these five modules and provide the details later on in the following Sections.

Challenge Generator: This module constructs a set of challenges, by enumerating all parameter combinations of each challenge model that the game designer intends to introduce in the game. That results in a large number of partially–filled instantiations of the challenge model detailed in Chapter 3.2. In the next Chapter, we define the *Challenge Model* and describe all its' parameters, then detail how and in which stage of the customization process they will be filled up.

Challenge Valuator: This module partially customizes the challenge's parameters that play a crucial role in following the concept of "Flow" in Gamification. This module is broken down into two sub-modules: *Difficulty Estimator* and *Reward Calculator*. The former is to estimate the difficulty level of each candidate challenge produced by the Challenge Generator for a given player using the distribution of all players' activities, and the latter sub-module is responsible to compute a commensurate reward for the specific challenge.

***Filtering & Sorting*:** This module implements a multi-criteria recommendation algorithm that aims to find a sweet spot between players' preferences (considering players' objectives, game status, etc.), and the objectives of the entity promoting the gamification campaign (in our case, a Smart City's interest could be incentivizing citizens to embrace and retain behaviors, which are in line with the objectives set by the administration). Finally, the module, considering the above criteria is able to pick up the k-top challenges in order for players to recommend. K is a parameter that can be defined by the game designer. For example, in our case study we set $k=2$ and assigned 2 challenges for each game week.

***Machine Learning*:** This module utilizes ML techniques using the player's performance to assess what challenges are most likely to be accepted and successfully completed by a player. The great advantages of machine learning have been proved in electronic game context to capture the hidden information, and continuously assess and predict the future outputs (e.g., generating new game content, adjusting new game level or recommending a new product) that fit to the players' preferences or game styles [32, 88, 110]. In Chapter 3.3 we discuss how we integrated and activated the ML module into the APCGR system. The proposed framework is implemented in Urban Mobility System, but it can be used in other domains such as Education, etc.

Player Modeling: In order to advance the personalization in gamification this module is constructed to extract the play style of the player in gamification. This module enables the system to automatically extract and capture the pattern that players play in the game (e.g., how precise , how fast/slow they play in the game). Consequently, the results can be used to tailor the game content toward their play styles aimed to maximize the players' engagement in the game.

Being the focus of this dissertation, a detailed description of these modules can be found in Chapter 3.2.

3.1.2 Gamification Services

The *Gamification Services* layer exposes the functionalities realized by the enablers as services, which can be exploited to build new gamification components and applications. For instance, services supporting the definition and deployment of games, services for accessing information about game and player state, services supporting the configuration of players' notifications, to name just a few.

3.1.3 Gamification Front-end

The *Gamification Front-end* layer contains end-user applications for the different stakeholders: it provides applications supporting the definition and deployment of games (for game experts), the presentation of game state (for game players), and the analysis of game results and impact vis-a-vis to the city objectives (for officials and decision makers).

The developed platform has been released in GitHub under the Apache License Version 2.0 [1] and is available as a stand-alone application as well as a software-as-a-service. This Gamification Platform has been deployed and experimented in a variety of Smart Cities games around Europe. The Gamification Platform is generic and we are exploring its application to various Smart City domains, including energy efficiency, participatory e-government, educational game and possibly health care. The main application domain that has been deployed so far hinges on the above platform is Urban Mobility system, in which we developed an App called Viaggia Trento Play&Go that players can install, register and interact with the game using their personal hand-held *Android* and *Ios* devices.

The detailed description of the implementation and evaluation of Trento Play&Go is provided in Chapter 4, which is followed by the assessment of the proposed framework. The two *Gamification Platform* and proposed APCGR (in sub-section 3.1.1) briefly described above demonstrate a generalized view of a gamified system that could be implemented to various aspects of a Smart City. Thus, in the following Sections, we fully describe the design of Automatic

[1] See `https : / / github . com / smartcommunitylab / smartcampus . gamification`

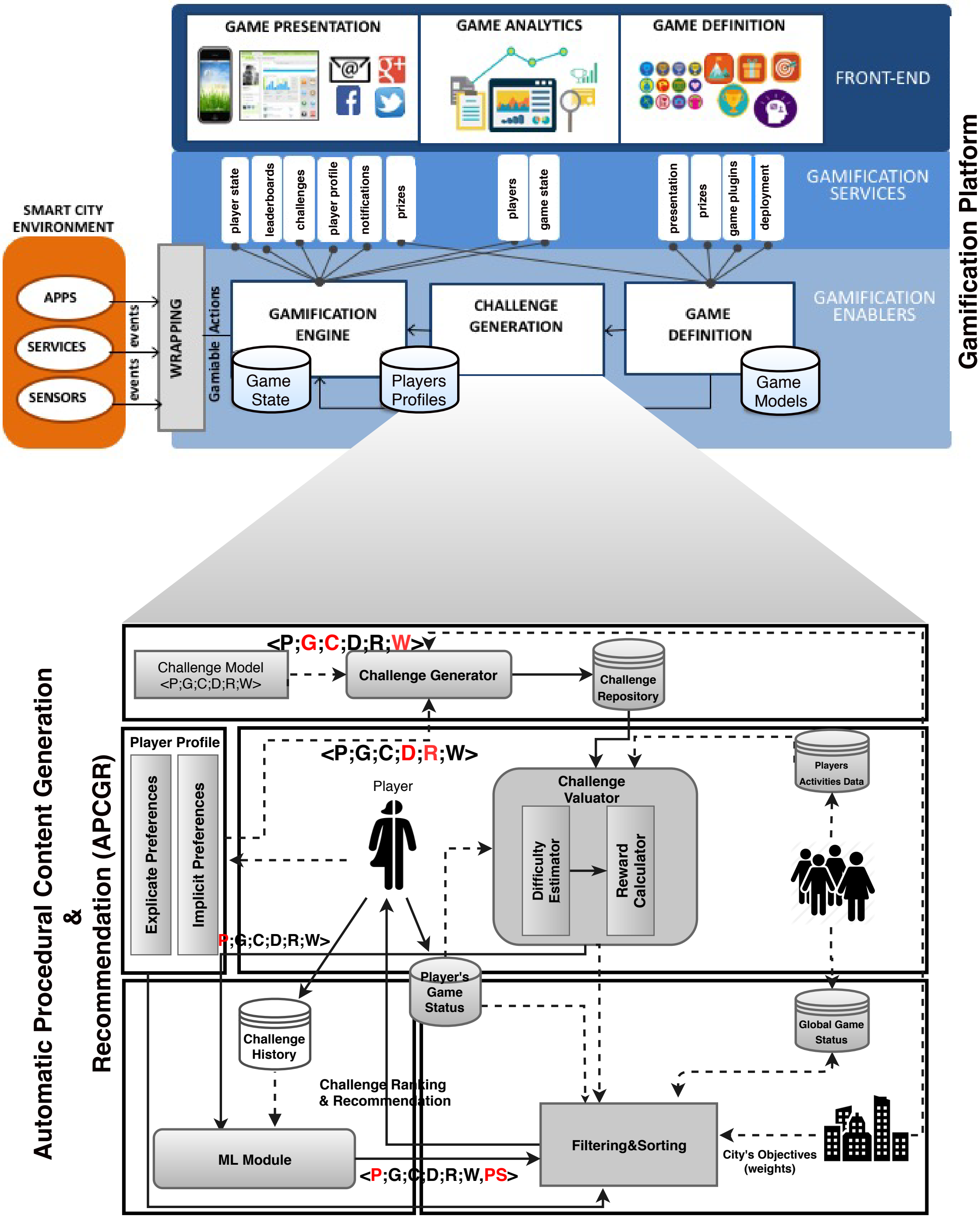


Figure 3.1: The Layout of the Smart City Gamification Platform and the Proposed Automatic Procedural Content Generation and Recommendation.

Procedural Content Generation and its extension by integrating the Machine Learning and Play Style recognition modules.

3.2 Proposed APCGR Framework

3.2.1 Challenge Model Definition

Challenges are units of a playable content consisting of a demanding goal or task that a player should achieve –under temporal or other constraints– in exchange to an in-game prize or reward. We define the playable unit as a tuple: $< P; G; C; D; R; W >$, where

- ***P*** refers to the individual player to whom the challenge is assigned. P has a profile, which may contain her preferences, either explicitly or implicitly derived. *Explicit* preference points at specific means/ or a set of means that the particular player expressed during the registration process at the beginning of the game, while *Implicit* preferences declare the actual player's behaviors which are captured and analyzed by the ML module. This type of preferences could be reflected to: use a specific means or set of means, kinds of challenges that she succeeded or failed during the game, etc.

- ***G*** defines the goal, that is a task or a performance target, which should be fulfilled to complete the challenge. For example, in our sustainable mobility game, *G* consists of two sub-elements that need to be filled up as follows:

  1. *MI* which indicates the mobility indicator,
  2. *T* that contains a value either for the amount of improvement (IMP), or the numbers of trips related to the type of the challenge.

  For instance, a goal can be expressed as *"take at least 6 public transport trips"*, or "*increase your walking activity by 20%*".

- ***C*** is a constraint for reaching the goal; a typical example is a *temporal deadline*. For example, player P must achieve goal G *within one week*. Here, *One Week* is a duration that the particular player has, in order to finish the given challenge.

- ***D*** represents the difficulty of the challenge for player P, considering goal G and constraint C. For the difficulty, we have been using a 4-level scale: *{Easy, Medium, Hard, Very Hard}*. Notice that the difficulty level for the same challenge, that is, with same goal and constraint, may be different for – and tailored to – different players. How to design these difficulty levels is detailed in the "*Difficulty Estimator*" sub-module, in the next Section.

- ***R*** is the reward (a.k.a. prize) awarded for completing the challenge. We define rewards in terms of GDEs that are part of the game. For example, a prize can be a bonus or booster for points accumulation, a badge in a collection of achievements. Rewards should be commensurate to the difficulty of the challenge that can also stimulate the players to accept and handle the challenge. To calculate and assign the right point to the particular challenge we use a type of linear function which is described in *Reward Calculator* sub-module.

- ***W*** is a numeric weight that captures how important the challenge goal G – and the behavior it means to foster – are, according to the entity promoting the gamification campaign. For instance, in a sustainable mobility game, a Smart City administration may assign during its "public transportation promotion week" highest weights to challenges that aim to increase usage of buses, and trains. For example, the numeric weight for the different transportation means can range from 1 to 5, in which the higher the value, the more important the transportation mean, in that specific game week.

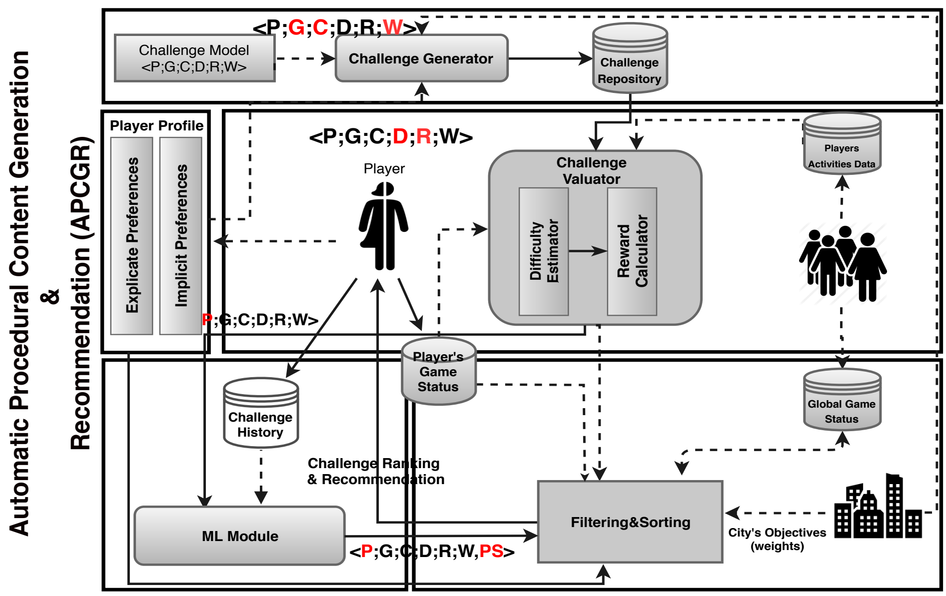


Figure 3.2: The Conceptual View of The Proposed PCG Framework.

In the following we list some examples of the challenges that our proposed framework, which is depicted in Figure 3.2, can recommend to users/players:

Example 1: "Increase ⟨Walking⟩ ⟨Kms⟩ by at least ⟨10%⟩ during⟨next week⟩ and receive ⟨100⟩ ⟨Green Points⟩".

Example 2: "Increase ⟨Public Transport⟩ ⟨legs⟩ by at least ⟨30%⟩ during ⟨next month⟩ and receive a ⟨177⟩ Green Points per ⟨leg⟩".

Example 3: "Do at least ⟨1⟩ ⟨bike sharing⟩ ⟨trip⟩ ⟨next week⟩ and receive ⟨150⟩ ⟨Green Points⟩".

Example 4: "Improve ⟨Bike⟩ ⟨Kms⟩ by at least ⟨20%⟩ during ⟨next week⟩ and receive a ⟨139⟩ Green Points".

The challenge model is filled up at different stages by passing through the modules, as depicted in red font in Figure 3.2. Borrowing from the taxonomy in [50], our procedural generation and recommendation approach described in the remainder of this Chapter can be characterized as an *on-line* – as it takes into account the current player state – *constructive* – as the generated instances are built all at once for every round of challenge administration and are all valid by design – and *parameterized* – as the generative process works by choosing appropriate values in a parameter space – case of PCG applied to gamification, which results in the injection of personalized and contextualized units of playable content.

Let us now describe this process, and the three modules of "Challenge Generator", "Challenge Valuator" and "Filtering & Sorting", which are responsible for it.

3.2.2 Challenge Generator

This module is responsible for generating a set of candidate challenge instances, that are then stored in the *Challenge repository*, by instantiating the player (P), goal (G), constraint (C) and Weight (W) parameters of the challenge model.

In our sustainable mobility game, G consists of two sub-elements: the mobility indicator MI and the target T, that specifies the target value to be reached to win the challenge. As mobility indicators we considered both Km and trips to be done by different transport modes (i.e., walk, bike, bus, train, bike sharing, park and ride) or combination of modes (i.e., public transport, impact zero, no car trips). In the game we also set the constraint C to "one week" for all generated challenge instances.

For example, a challenge instance with values, P=Alice, MI=Bus_Trips, T=8, and C=1 week, asks player Alice to take at least 8 trips by bus in the next week.

In order to instantiate challenges the module takes into account players' history (particularly, previous week) and follows the below approach for parameterizing G:

- For each player P and each mobility indicator MI, if P has done any activity for MI during previous week, the algorithm fills T considering a set of target percentage improvements (i.e., 10%, 20%, 30%, .., 100% as defined in Table 3.1). For instance, if Alice has done 10 Walk_Km in the second week of the game, the module takes this number of KM traveled as the baseline to instantiate the target values for candidate Walk_Km challenges in the third week. In this example, it will generate ten challenge instances: "Do at least T$\in \{11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$ MI=Walk_Km during C=next week".

- If the player was not active for a specific mobility indicator MI=Bike_Trips during the previous game week, the algorithm sets T to 1, requiring the player to do at least one trip. E.g., "Do at least T=1 MI=Bike_Trips during C=next week".

More candidate challenges are similarly derived for all players and mobility indicators, and they are all stored in our Challenge Repository.

This challenge generation process is time-bound and is repeated weekly. Although for our mobility game we set constraint C to one week, the proposed approach is generic and can support for different constraints on challenge duration. For example, this element can be set daily, or even the module can be programmed to generate new personalized challenges, when the particular player can overcome the given challenges earlier than the appointed constraint. This fulfill the requirements of the players who are interested to play quickly and gain more points to win the game as fast as possible. Practically, these players are distinguished as *Achiever/Goal Seeker* players type under the umbrella of *Player Modeling* concept in digital game community, which we discuss in Section 3.4.

3.2.3 Challenge Valuator

The purpose of this module is twofold: estimate the difficulty of each candidate challenge produced by the Challenge Generator for a given player, and then compute a commensurate reward. Therefore, the module is responsible to fill the D and R parameters of our challenge model which are highlighted with red font in Figure 3.2. Accordingly, it consists of two main sub-modules: *Difficulty Estimator* and *Reward Calculator*.

Many studies in the context of game and gamification discussed the importance of relationship between players' engagement and the challenge specification (e.g., level of difficulty, reward) [111–114]. Indeed, accurately estimating the difficulty of a challenge with respect to the player's game status, game skill and progress is significant for two reasons. Firstly, it is a pre-requisite for the system to assign a fair reward for the effort required to complete that challenge; Secondly, it contributes to keep the player interested in the game by striking a balance between the satisfaction for accomplishing goals and the stimulation of being challenged –concept of flow [42, 115]. Hence, to find out the difficulty balance of a challenge for each personal player we have followed the below steps.

Estimation of Difficulty

Since we have a dynamic game competition where participants have different personalities, skills and preferences such that these features (skill, personalities, etc.) may be affected either positively or negatively, by the internal or external context (e.g., motivated positively by other players' activities, or external reasons for negatively like weather, or some physical issues on player), we need to have a different level of difficulty that can be matched with player's skill, progress, etc., and changed in different stage of the game.

Hence, we have utilized an *Unsupervised Clustering* algorithm to differentiate the amount of players' activities in each particular mode. We considered the output of the clustering as the number of difficulty levels. Thus, we implemented the *Expectation-maximization (EM)* algorithm [116] considering only one feature to find out the maximum likelihood among the players' activities in each specific transportation mode. For example, for mode walk we took into account the amount of kilometers that each player has recorded during a game week; for mode bus, we considered the number of bus trips that she (as a player) has done in the previous week; and similarly for mode zeroImpact, we heeded the number of trips (in the last week) which were recorded in her profile. Thus, by conducting the *EM* algorithm on the recorded data for each transportation mode, we have identified *Four* different clusters that express the different level of players' activities. However, for some modes we have extracted a different number of clusters (e.g., 2, 6), this might have happened due to the noisy data (e.g., extreme values).

In our work we have normalized the levels of difficulty into *Four* categories and labeled them *Easy, Medium, Hard* and *Very Hard*.

Once we obtain the number of difficulty levels, by given a challenge goal and a candidate player, our Difficulty Estimator module assigns a difficulty label among [Easy, Medium, Hard, Very Hard] by taking into account the distribution of all players' past performances related to that goal.

For example, if the goal of a challenge requires player *P=Alice* to walk X Km. during the next week of the game, the relevant distribution is that of weekly walked Km among all players[2]. We divide that distribution in 10 equal intervals, that is, *deciles*, and 4 "Zones", as follows (see Figure 3.3):

[2]This may configure, at the very beginning of the game, a sort of *cold start problem*; to bypass that, one can have an initial phase of the game without injection of challenges, in which sufficient game data is collected; alternatively, one can leverage data from previous instantiations of the same game, or any other suitable statistics, to establish an initial baseline distribution.

- Zone 1: the first zone contains the first four deciles or 40% of the data. This zone has more players inside w.r.t the other decile, and the players in this zone have done less activities compared to other deciles.

- Zone 2: the second zone contains the following three deciles, or 30% of the data.

- Zone 3: the third zone covers two deciles, or 20% of the data.

- Zone 4: the forth zone, which is the last zone, includes the last 10% of the data.

We then evaluate the position in the distribution of the past performance of the candidate player vs. the performance required by the challenge goal, according to the following rules:

- Easy: difficulty is set to *Easy*, if completing the challenge does not move the player's performance from the current zone to the next one. Hence, to define the difficulty of the goal the movement of the performance from the current zone (that could be the first zone, second zone, etc.) is considered,

- Medium: difficulty is set to *Medium*, if completing the challenge moves the player's performance from the current zone to the next one. For example, in Figure 3.3, a movement from *zone 2* to *zone 3* is considered as a *Medium* Challenge,

- Hard: difficulty is set to *Hard*, if completing the challenge moves the player's performance two zones higher. An example is shown in Figure 3.3 such that a movement form *zone 1* to *zone 3* is labeled as a Hard challenge,

- Very Hard: difficulty is set to *Very Hard*, if completing the challenge moves the player's performance three zones higher; A Very Hard movement is illustrated form *zone 1* to *zone 4* in Figure 3.3,

- If the challenge goal would move the player's performance beyond the tenth decile, we consider the range *(V)* of distribution values in the tenth decile to build a dynamic threshold to set the challenge difficulty. The next zone is set at the maximum of the distribution plus *1*V* (representing, in a sense the eleventh decile). The previous rules apply then to this extended range. For example, if the current player performance is already in the highest zone, and the goal would move at about *2*V* beyond the maximum value in the observed distribution, the challenge is considered *Hard*.

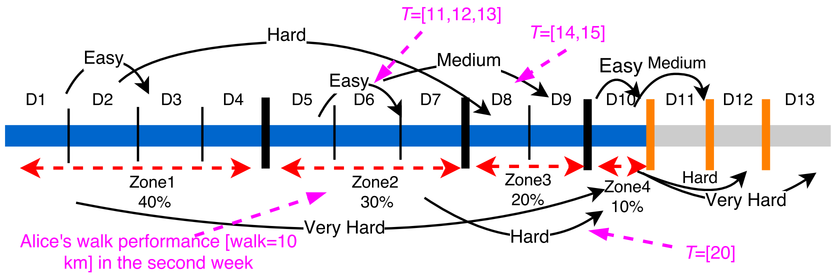


Figure 3.3: Difficulty Assignment Method.

For instance, looking at Alice's history in the second week of the game (Walk_Km=10) and considering the distribution of all players activities in that mode, Alice's performance is located in Zone 2 (see Figure 3.3). Hence, the module assigns difficulty $D{=}Easy$ for the challenge instances with T$\in \{11, 12, 13\}$, since these improvements do not move Alice's current position to the next Zones. While the other challenges, T$\in \{14, 15\}$ and T$\in \{20\}$, move Alice current position of one and two next Zones respectively, thus $D{=}Medium$ and $D{=}Hard$ will be assigned for these challenge instances.

Computation of Rewards

Challenge rewards are construed as "in–game" rewards: winning a challenge gets rewarded with some GDE, such as a badge for an achievement, a point bonus, etc., which has the potential to improve the player's status in the game. In this current implementation, the Reward Calculator sub-module works with point bonuses as the chosen form of challenge prizes.

The value of the assigned reward/prize "R" depends on the estimated difficulty level (dif), and the amount of behavioral improvement (imp) required by a given challenge, according to the function below:

$$R = f(dif, imp) \tag{3.1}$$

This is a type of linear function called "Plane Flat"[3] that is used to construct the table of rewards (see Table 3.1). The plane is a flat surface that exists in three dimensions and can move in each direction shown in Figure 3.4. For example, as is described in Section 3.2.3, we defined 4 level of difficulties that reflect in 4 points in one direction of the plane (Easy, Medium, Hard and Very Hard –highlighted by red in the Plane). Similarly, for the Improvement, 10 points are located in the other direction of the plane, since we defined 10 different percentage of improvements. It is noticeable, that these points can be selected and parameterized, by any numbers in any direction of the plane so that we can reach up to infinity planes. Hence, this is just a case that we implemented in our project.

[3] https://en.wikipedia.org/wiki/Plane_(geometry)

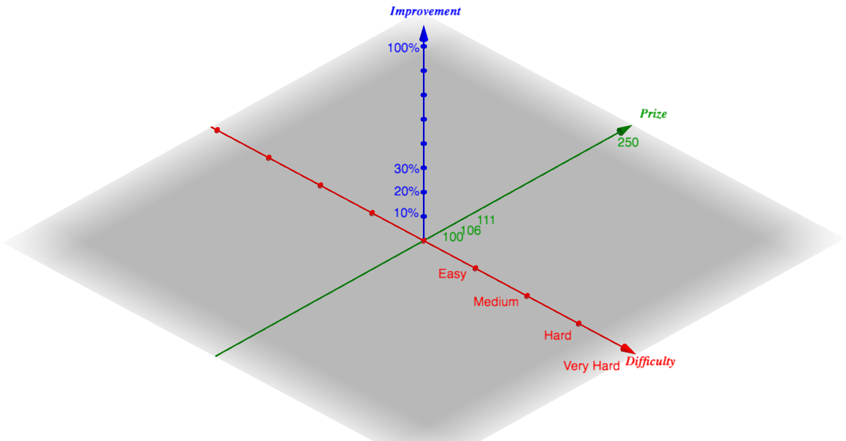


Figure 3.4: The Visualization of 2 Dimensional Plane Function.

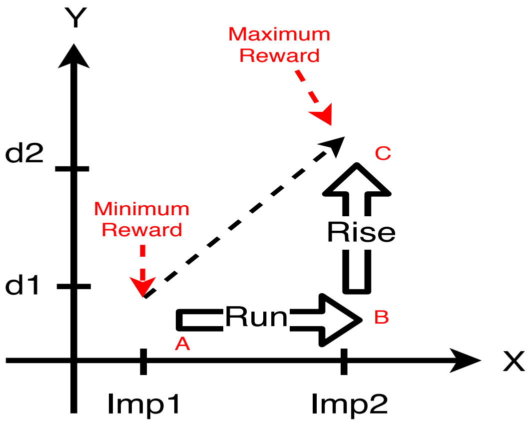


Figure 3.5: The Visualization of Run and Rise.

Thus, to calculate the rewards in all the rows in the table, three points need to be specified (they are explained, with an example in the next paragraph). Then, we need to calculate two other elements called Run and Rise[4]. Run refers to the particular moving "Slope" within a horizontal line (x-axis) segment from a point A to a point B (in our case is the improvement and indicated with m in Equation 3.2). While, Rise points to the changes in a vertical direction (y-axis), e.g., from point B to a point C as indicated in Figure 3.5. In our case, it is the level of difficulty that specified by n in Equation 3.2.

[4] https://www.math10.com/en/geometry/line-slope/line-slope.html

Hence, taking into account Function 3.1, f has constant slopes in both the $x = imp$ direction and the $y = dif$ direction (see Table 3.1), so that we fill the table by the following equation:

$$\begin{aligned} R &= m(imp) + n(dif) \\ m &= (imp_{max} - imp_{min})/(N-1) \\ n &= (dif_{max} - dif_{min})/(M-1) \end{aligned} \tag{3.2}$$

where m is the slope along the *(imp)* dimension, when *dif* is constant "Run", and n indicates the slope along the *(dif)* dimension, when *imp* is constant "Rise"; N and M are the numbers of different improvement brackets (e.g., 10%, 20% , etc.) and the number of difficulty levels, respectively. Hence, $M-1$ and $N-1$ specify the maximum number of movements in improvement (10-1=9) and in level of difficulty (4-1=3) in the plane, respectively.

The formula above can produce a table of prizes for all improvement / difficulty combinations for a given challenge (see Table 3.1 for an example); it simply requires the game designer to provide a range with minimum (e.g., in our case 100 as it shown in Table 3.1) and maximum (e.g., 250 in our case) prize, plus one additional point value (e.g., the value for the top / right corner in the table) in order to set the m and n slopes as desired (the three points mentioned above). By associating those values to each mobility indicator, the designer can modulate the relative importance assigned to behavioral improvement (a general characteristic) vs. difficulty (a personalized characteristic).

Table 3.1: Prize Table For a Challenge With a Prize Range of 100-250 Points.

Dif \ Imp	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Easy	**100**	106	111	119	125	130	135	140	145	**150**
Medium	133	139	144	150	156	161	165	170	175	183
Hard	166	172	177	180	186	192	197	203	210	217
VeryHard	197	205	211	219	225	230	235	240	245	**250**

IMP: Improvement.
Dif: Difficulty.

As an example, we recall the challenge instances generated for Alice related to *MI=Walk_Km*. The module calculates the rewards R by taking into account the difficulty and the percentage improvement and assigns the following rewards: $R = \{100, 106, 111\}$ for 10%, 20%, 30% improvements (labeled as *Easy*), $R = \{150, 156\}$ for 40% and 50% improvements (*Medium* challenges) and $R = 217$ for 100% improvement (*Hard* challenge).

3.2.4 Filtering and Sorting

Filtering & Sorting is the final stage, which takes generated challenges that have been valuated for each individual player in terms of difficulty and reward, and recommends a subset of them.

This module tries to find a *sweet spot* between the player's own interest, and the interest of the entity promoting the gamification campaign (in our case, a Smart City). The player's interest is to take game actions that maximize her chances to elevate her status in the game; the city's interest is to incentivize as much as possible its citizens to embrace and retain behaviors that are in line with the policies (objectives) set by the administration.

We construe the player's interest by considering objectives for the player, which are typically tied to some GDE included in the game. For example, in a point-based game a game objective can be to climb at least one position in the leaderboard for a certain type of point; another objective can be earning enough points to reach a new level; in a game based on achievements, an objective could be instead earning a new badge, to come closer to complete a certain badge collection, or unlock some other new game element. In general, at any point during a game, a player may be interested to reach one or more such objectives. If we consider the distance between those objectives and the current state of the player, as depicted in Figure 3.6, a good challenge for the player is one which offers a reward that enables that player to reach (or at least come as close as possible) one of her current game objectives. Among any such challenges, the ones that are less difficult to win are preferable from the player's point of view, however some players do not always prefer less difficulty. For instance, some players would like to deal with the difficult obstacles that sourced from their intrinsic motivation [117, 118], rather than gaining rewards (with less effort) to compete with the other players in the game.

In Chapter 3.3, we have detailed how the framework can obtain the actual players' preferences w.r.t the challenge difficulty, and tune these settings for the personal player.

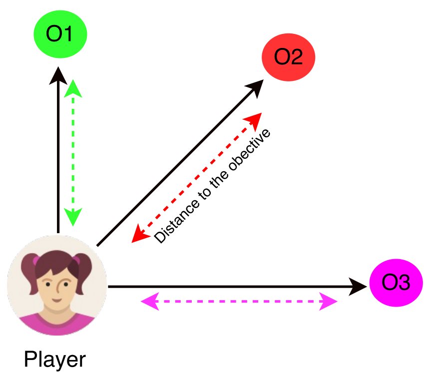


Figure 3.6: Game Objectives vs. Player State.

In this stage of the Chapter, we discuss the two important perspectives of the *Player* and the *Smart City* that the module takes into account to filter and sort the valuated challenges in her challenge pool:

- From the player's perspective: our recommender system tends to adopt an opportunistic stance, and aims at providing immediate value and a measure of satisfaction, in return for the player's effort to complete the proposed challenges. Moreover, the recommendations produced in this way are situational: a challenge that is preferable at a given juncture in the game, may become less valuable for the same player at some other time.

- From the Smart City point of view: the preferable challenges are those most in line with the goals currently being promoted, and among those, the ones that incentivize the most significant behavior improvement for the player to whom they are assigned.

The above two perspectives reveal a potential contradiction between two preferences. On the one hand, most players will wish to gain more points with less effort. On the other hand, the Smart city will try to motivate (engage) the players towards higher behavioral improvement for the least possible prize. To tackle this problem and identify the best candidate challenges that can find the sweet spot between these two perspectives: the recommender system uses the concept of *Weighted Improvement* (WI), that is, the product between the weight property W of the challenge and the amount of improvement mandated by its Goal. *Weight* W is a static value between [1-5] and is assigned to each mobility indicator on the basis of the theme of the week by the game designer. For instance a challenge which requires a 20% improvement, and has a weight $W=5$ has a $WI=100$, and is preferable to another challenge which requires a 30% improvement but has $W=3$ ($WI=90$). It may happen in a public transportation theme week, e.g., the importance of mode Bus has set more than mode Bike (for example W=5 vs. W=3), however completing a challenge with Bike mode seems difficult than a challenge with Bus mode. For example, improve your bike trips/KM. by 20% vs. increase your Bus trips by 20%.

At the end of the process described above, the recommender system proposes to each player the K top personalized and ranked challenges from her challenge pool, where K is a parameter that can be adjusted by the game designer. In Chapter 4, we describe the application of the proposed framework in a gamified urban mobility experiment (Trento Play &Go) aimed to evaluate the effectiveness and efficiency of the personalized game content (particularly challenges) in changing the behavior of players towards more sustainable transport means.

The further component of our system is the *Machine Learning* module. We designed and integrated the module into the proposed APCGR aim to enhance the performance of the challenge recommendation process in Filtering

& Sorting module. Hence, we have described this module in Section 3.3, where the module leverages machine learning techniques over the personal player history and performance on past challenge to further personalize and tune the selection, filtering and sorting of challenges to be administered in the future.

3.3 Machine Learning Integration for Personalized Challenges

Section 3.2 described a system and framework that was designed to automate the generation of game content in gamification. The proposed APCGR framework generates personalized game content –particularly *challenges* in this study– and acts as a recommender system to assign the personalized challenges to each individual player. Since the *Challenge Model* inspired from concatenation of several components (introduced in Section 3.2.1), this personalization was not only applied on one single game content. Indeed, it insufficiently covers multiple game content in the gamified system. For example, game difficulty balancing or dynamic difficulty adjustment (DDA) [119, 120] was implemented by introducing diverse level of difficulty for each individual player w.r.t her ability and activity in the game. It is also reflected in the prize by tuning the reward that commensurates with the goal/task (that is generated and assigned, towards the player's ability) and the difficulty level.

Having said that, this personalization was deployed only based on the previous week of players' activity and participation in the game. In addition, the framework did not consider *players' feedback* (with respect to the given challenges) to generate and tailor the challenges. Player's feedback refers to *the action that she behaved in the corresponding week of the game to overcome the recommended challenges*. This brings forward the need to develop a module that is able to fill this gap to further personalize the unit of playable content in gamification. In fact, advancement in tailoring game content w.r.t the players' characteristics and personalities [83, 121] (e.g., players' play style including explorer, goal seeker), also help to boost player's engagement and motivation in improving her behavior in the game.

Machine Learning (ML) with its ability to discern patterns from players' data (activity) in order to predict future output, is a subset of AI that exploits statistical techniques to reach the above intent (advancing personalization in gamification). Indeed, ML became an essential and inevitable pillar in a wide range of research domains including e.g., pattern recognition [122], data mining [123], expert systems [124], entertainment and game industry [37]. These potential have been also proved in the domain of electronic games to extract hidden information, continuously assess and predict the future output (e.g., recommending a new product like a new action game similar to the ones that the player liked before, generating new game content, adjusting new game level) with respect to the players' personalities or play styles [32, 88, 110].

In the context of gamification, Machine Learning (ML) can be applied in several aspects of the game to predict the forthcoming action from players who are involved in the gamified system. For instance, learning techniques could be used to identify a pattern from player's behavior (e.g., how fast and slow they played the game, how much they are precise to proceed the game) in dealing with the game to be used in game adaptation, where game content could be tailored w.r.t her pattern. It can also be used to model the players' abilities and skills in order to handle various obstacles, challenges, etc.

To this end, as the preliminary work, we have extended our APCGR framework by equipping it with a Machine Learning (ML) module aimed at optimizing the challenge selection process in Filtering & Sorting module. In general, the module takes and uses player's history (in the game) as a model to properly assign the generated challenges that can suit her preferences under the direction of the gamified system. In other words, the system takes into account the challenge components to learn her feedbacks w.r.t the given challenges in order to enhance Filtering & Sorting module's performance to obtain a set of values (for each challenge components) that can help to boost players' motivation.

Therefore, given the history of a player (P) that is a list of challenges $P_i = \{ch_1, ..., ch_n\}$, where each challenge contains multiple features $Ch_j = \{f_1, f_2, ..., f_m\}$, indicating the players' behavior/ability and habits w.r.t the given challenge, the challenge recommendation system aims at personalizing and suggesting the challenge(s) with highest probability of success among the wide variety of challenges that could be instanced by the module.

3.3.1 Feature Selection and Modeling

Feature selection is the set of operations that extract a subset of attributes from the data-set and exploit it in modeling the specific action, behavior, etc. for prediction purpose. In the context of machine learning, feature selection aims to reduce the size of effective attributes that lead the training and the classification more efficient. In addition, it enables the system in order for improving the classification performance by eliminating unnecessary and noisy attributes. In fact, feature selection boils down to capture hidden business insights and then make the suitable decision in electing attributes by removing irrelevant features that do not contribute to the accuracy at predicting the model. In general, feature selection algorithms are broken down into three distinct categories of: *Filter* methods (e.g., information gain, Chi squared test and correlation coefficient scores.), *Wrapper* methods (e.g., best-first search and random hill-climbing algorithms) and *Embedded methods* (e.g., regression algorithm) [125, 126], which are utilized hinge on different purposes in various contexts.

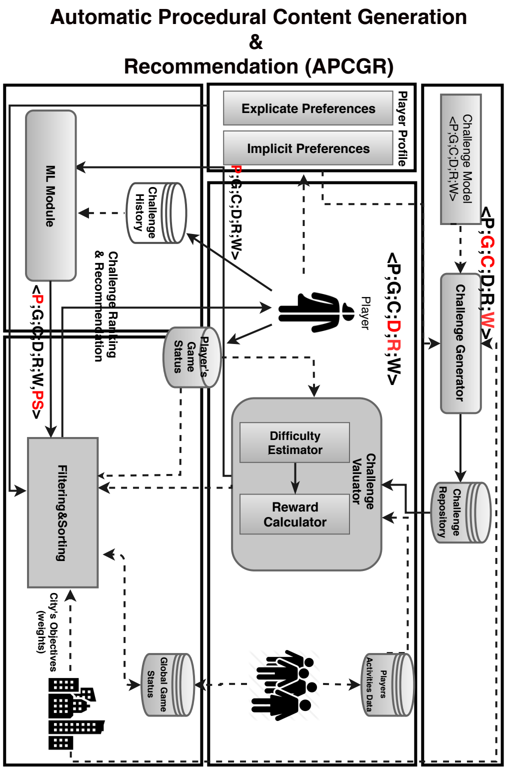
According to the aim of the module "*optimizing the challenge selection process in Filtering and Sorting module*", we concentrate on the features that are represented and shaped in/as the Challenge model. Since the challenge model inspired from a bounded number of features in this study, it is not needed to use such feature selection algorithms to decrease and remove the redundant attributes. Thus, we have exploited our domain knowledge and selected four features out of the attributes we singled out in the Challenge model (Section 3.2.1), to model only players' feedback with respect to the given challenges as follows;

- **Mode**: The *Mode* of transportation such as *Bus, Bike* and *Bike_Sharing*, *Walk, Train* or the combination of these transportation means that could appear by *ZeroImpact* trips, *Public Transportation* means, to name a few;
- **Type of Challenge**: This element could be filed up by two kinds of challenges such as *percentageIncrement*, that asks players to improve by some given percentage one's performance on a mobility indicator with respect to the previous week, or *absoluteIncrement*, which requests a fixed numbers of trips for a mobility mode;
- **Reward**: Is the prize assigned to the player who completed the given challenge;
- **Improvement**: This element inquires players to improve the specified mode's usage w.r.t their last week.

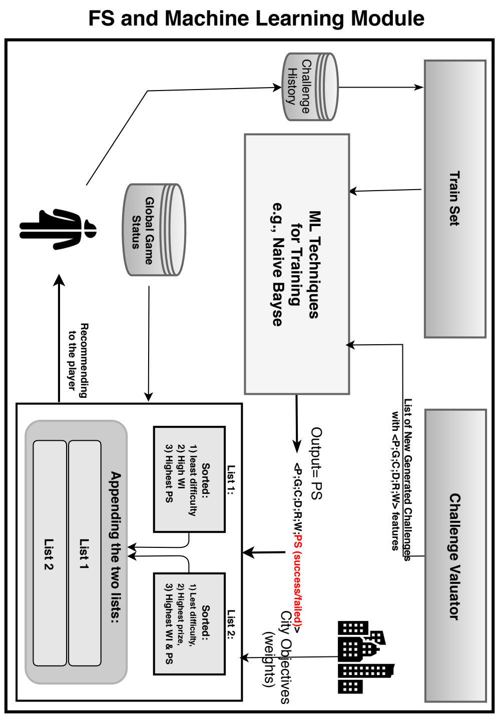
We take into account the above features as the main elements that can manipulate the prediction of the output in our gamified mobility case study, in which the outputs/results could be "succeeded" or "failed". "*Succeeded*" refers to the generated challenge(s) that the player can complete and finish it, while "*Failed*" points to challenge(s) that the player most likely is going to fail. The action flow is detailed in Figure 3.7, where the list of player's past challenges (from challenge history) is sent to the ML module as the "train set" to build a Model using a classification method. The ML module is also injected with the list of new generated and evaluated challenges as a "test set/input" from the *Challenge Valuator* Module. Hence, using a particular classification model (e.g., Naive Bayes, Decision Tree, etc. which are detailed later on), the output of the ML module is a value that indicates the probability that the challenge will be in a specific class ("succeeded/failed").

Once the prediction is done, the module appends the predicted value (probability of success*PS*) to that particular challenge as one extra parameter that is exploited by the *Filtering & Sorting* module. Giving the list of challenges from the ML component, the *Filtering & Sorting* module operates for each individual player according to the following algorithm, which is shown in Figure 3.7b:

1. Divide the challenges in the repository in two sub-lists: those whose prize is sufficient for the player to reach some of the player's current



(a) The Conceptual View of the APCGR



(b) The Conceptual View of the FS and Learning Modules

Figure 3.7: The Design of the Framework and the ML Module

game objectives "List 1", and the others "List 2" which are not well enough to pass from any game objectives. As we mentioned in Chapter 3.2, Section 3.2.4, player game objectives could be gaining enough points to catch a new level of the game or climbing at least one position in the leaderboard for a certain kind of points.

2. Sort the first subset of challenges by least difficulty;
3. within each difficulty level, calculate the weighted improvement of each challenge and sort them by highest *WI*. We have described in Section 3.2.4, what the weighted improvement is, and how to calculate it.
4. Thereafter, sort the above sorted list by the highest probability of success *PS*;
5. sort the second subset of challenges by least difficulty and highest prize. Then the list will be sorted again by weighted improvement, and finally *PS*;
6. Append the sorted list from the second subset at the end of the first sorted subset to obtain a single ranking.

At the end of the process described above, the system proposes to each player the K-top challenges in her personalized ranking above (where K is a parameter that can be adjusted).

We implemented the proposed ML module on the data collected from a gamified urban mobility experiment "Trento Play&GO". The aim of the experiment is to assess the effectiveness of the ML module in predicting the players' feedback with respect to the recommended challenges. This prediction can contribute to maintaining players' participation in the game. Thus, in the next Chapter we detail the design of ML module implementation, which is concentrated on two important aspects of Effectiveness and Efficiency to validate the potency of this module.

3.4 Players Modeling Integration in Gamification

In the previous section (3.3), we have presented a possible design of the ML module that integrates into the APCGR framework in improving the success rate of the recommended challenges. However, the success rate of the challenges does not ensure that the player has improved her behavior under the direction of the gamified system, while it has positive influence to make them engaged in the game; therefore they can gradually improve their demeanor in a sustainable fashion. We constructed the ML to partially personalize the game content –particularly challenges– w.r.t the various features including player's skills, game status, player's objectives, player's feedback, as well as the objective of the game designer. In Section 3.2.3 we discussed that players are distinct and have different abilities and skills to handle different challenges, which we introduced challenges with various level of difficulties. The differences in players' abilities and skills may reflect in different characteristics and traits that may appear in the way they play the game. For example, some players play fast and some slow, some of them play precisely and some concentrate on the game content and some do differently. Thus, taking into account the players' play styles in tailoring the game content may have a value to further engage them towards the direction of the gamified system.

This section discusses the essential steps to further improvement of personalization based on the players' play styles in gamification. In particular, we propose an approach to automatically extract player's play style in gamification that could be exploited to advance personalization in gamification domain. In the following sections, we first provide a notion of player modeling and follow up by some related investigations in the context of electronic game. Then, we detail the proposed approach including five modules: Data Collection, Data Preparation, Play Style Recognition and Play Style Prediction. Data Collection concerns the collection of the data from different domains. As the name implies Data Preparation discusses how it excludes the data from noise and outliers in order to send to the Play Style Recognition module. This module is in charge of extracting play style given the player's data using the introduced Utility Function. Finally, the design of Player Prediction module is presented; the module can predict the play style given the previous styles. This module enables the framework to adapt the game taking into account the chain of changes that the player has done in her behavior during the game.

3.4.1 Player Modeling Notion

The concept of *player modeling* refers to the study of computational models that have been widely used in the domain of video games for customizing game content to player's preferences, traits, abilities and personalities [80–82]. Advancements in player modeling have been driven by the need to increase players engagement and minimize frustration, for example, in educational games [107, 108], electronic games [127, 128], and digital entertainment [129, 130].

Play style represents the actual player's substrate and provides an understandable pattern to the system to be used for adapting the game based upon the player's behavior [87], may therefore hold value to augment the players' engagement for a long term, or recommending a new personalized product/game to the player [88].

Having in mind the existing work in play style taxonomies [83, 96, 103, 104] reported in Chapter 2, we aim to extract 4 main play styles. To recall, the play styles are listed as follows:

- *Achievers* are problem solvers and willing to play as fast as possible with a minimum number of errors. They usually focus on game content that is necessary to complete and win the game as soon as possible.
- *Explorers* analyze and investigate all the game elements. For example, they might visit all the game items with a significant amount of attention, and also make few mistakes in the game.
- *Careless* play quickly and make many mistakes/errors.
- *Lost players* play slowly and make many errors/mistakes.

These last two play styles usually do not pay any attention to the game and make little effort to play well.

Apart from the attempts employed in game adaptation based on play style, most of them assume that the play style of players is a fixed property which does not change over time as they play. However, as it is shown in [107], this is far from true, and players, in fact, often switch play styles within a single session of play.

In this Section, we build upon previous work [107] on identifying the dynamic play style exhibited by players in educational games (the 4 main play style introduced above). By following the strategies provided by Bifel [94] and Martin [95] (described in Chapter 2.3) to capture the play style, in this study we propose a module that automates the construction of player modeling in educational game-based learning and apply it in both on time scale and segmentation. The approach is equipped by five main modules, in which a scoring system is designed using a utility function to construct the play style of the player. The proposed scoring system is generic in the sense that it enables the framework to implement and score infinity play styles (as long as their traits are defined) in various game and gamification domains.

The key idea of the proposed approach is to automate the process of extracting play style of the player in each stage of the game that enables the framework to dynamically adapt the game to each personal player over time. The framework enhances the performance of the extraction by exploiting a type of *Scoring* function that is able to uncover the whole play styles that the particular player behaved during the game.

The proposed approach can be divided into four main modules: *Data Collection*, *Data Preparation*, *Play Style Recognition*, *Play Style Prediction* and *Game Adaptation*, however the main focus of this work is the play style *Recognition* module. This module has access to a collection of variables recorded for each player automatically sorted by importance in predicting play style, and considers these features and their weight in order to calculate the score of the play style over the time of the player at hand.

We claim that accurately recognizing and predicting the play style of players can significantly impact the way adaptive games are built, by choosing or even constructing new game content to fit the current player's play style. Game adaptation could be implemented for various aspects of game elements such as visual appearance, levels, challenges, difficulty and even story-lines [131, 132] by using systematic techniques like procedural content generation (PCG) [133].

The proposed approach has been evaluated on data captured from an educational interactive game-based learning called *Solving the Incognitum* [107], which aimed to teach the relationship between fossils and geological time record taken in the historic Charles W. Peale's Museum of Art and Science.

Borrowing from taxonomies in [83, 96, 103, 104], where various play styles were introduced from different studies (in various domains) such as Achiever, Socializer, Killer, Explorer, Lost or Confused, Philanthropists, Free Spirit, Careless, Uncertain etc. (detailed in Section 2.3), the proposed algorithm acts in a supervised way to extract certain types of play styles in this study. Hence, the framework seeks to extract the current play style of players as *Achievers*, *Explorers*, *Careless* and *Lost*, which suit to the application domain (in education game based context) among the other introduced play styles.

3.4.2 Play Style Extraction Approach

In this section, we describe the proposed play style identification approach that is constructed based on a *Utility Function*. At the core of the framework there is an agent that iterates through the features to identify the play style of players by constructing a vector of scores (for each player in each game section). This vector is achieved by applying the utility function on the sorted collection of features gathered from gameplay sessions. Sessions will be manually or automatically determined by the game designer or based on the structure of the game respectively (for example completing a level in the game, or achieving a certain amount of points in the game, etc.). This is due to the fact that game content could be generated and personalized to propose to each individual player in each section of the game, however the game content could be updated even at the middle of each section.

Hereby, we describe the module and its five main components responsible for play style recognition: *Data Collection*, responsible for collecting data while players are playing the game; *Data Preparation*, where data is polished from noise and outliers; *Play Style Recognition*, which groups players based on their play style; *Play Style Prediction*, which predicts the play style of the player at hand given the groups identified in the previous step; and finally *Game Adaptation*, which is responsible for personalizing the game content to an individual player's characteristics. These components are shown in Figure 3.8, however, the last component is out of the scope of this study and we focus on just the first four.

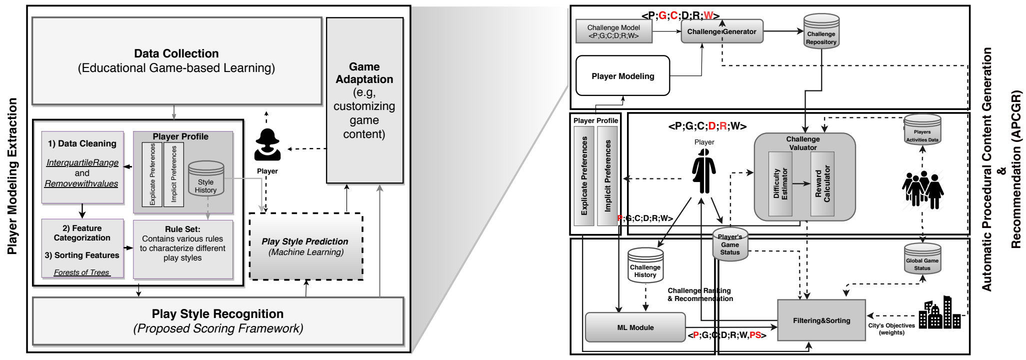


Figure 3.8: The Conceptual View of the PL Module Integrated to the Main Framework.

3.4.3 Data Collection For Player Modeling

This component is in charge of collecting data from players who participate in the game. In general, it logs players' raw data in their profiles that indicate the actual player's activity and participation during the game. From domain to domain, various kinds of techniques (e.g., monitoring, questioners), equipments (such as smart devices, mobile, tablet), interfaces (including web and mobile applications, etc.) could be used to capture the data.

In our urban mobility game described in Chapter 4, this component acts whenever the player starts playing the game, therefore the system is able to record the data into her profile based upon her activity. Hence, the execution of the main proposed framework and the proposed play style extraction can be mandated and determined in basis of game objectives or/and the game designer. For instance, the proposed APCGR framework introduced in Section 3.2 was executed once per week, by taking into account the data that was collected during the previous week to generate the personalized challenges to be recommended to personal player.

Having said that, in this phase of PhD project we intend to implement the proposed play style extraction in different game context such that the data collection environment is quite different. The data that we have used in this study to validate the proposed play style recognition module was collected from *Solving the Incognitum* [107], which is a game-based interactive learning environment. A screenshot of the game and its environment is shown in Figure 3.9.

3.4.4 Data Preparation

The purpose of this component is two-fold: cleaning the data from noise and outliers, and discretizing continuous features into categorical values. Accordingly, the component contains two main sub-components of *Data*

Figure 3.9: Screenshot of a game scene. .

Cleaning and *Data Categorization*.

Data Cleaning: Cleaning data from noise and outliers is critically important for two reasons. First, noise can have negative influence on data categorization, when we proceed to discretize the continuous values into discrete labels. Secondly, noise can also be the source of errors in extracting and predicting play styles, and consequently game adaptation.

We use domain knowledge in order to select the suitable set of features among the available features that were recorded to characterize the meaningful play style. Thereafter, WEKA[5] libraries are used to clean the data from outliers and extreme values. To this end, we execute *InterquartileRange* function (coded in Python) to recognize the outliers. Then, we use *Removewithvalues* (coded in Python) to remove those outliers from the data set. These result a data set free from noises that skew feature categorization. In Section 4.10.1 we detailed how many samples are excluded from the data set due to containing extremely noisy data.

Feature Categorization: Converting continuous data to categorical data has been widely studied in the literature of data analysis and machine learning [135], and can bring advantages and disadvantages in various cases [136]. For the specific case of play style prediction, our experiments show the admissible improvement with respect to directly using the raw continuous data (as done in previous work [107]). In our experiments, we have discretized each continuous feature into three categories (*Low*, *Medium*, and *High*). To perform this discretization, three intervals are defined by finding four points, where the first and last correspond to the minimum and maximum value of the feature, and the two middle points are calculated as follows:

[5]For this, we took the advantage of *InterquartileRange* and *Removewithvalues* functions made available in WEKA [134].

$Second\ Point = (mean(data) + minimum)/2$
$Third\ Point = (mean(data) + maximum)/2$

where *data* refers to the list of data points that players have done in a particular feature (e.g., a feature in our case study is "*item visited*" that refers to the number of items that a player visited in a specific section or time-window of the game.). Then, the three intervals are defined as:

- *Low:* indicates the region between first point and the second point.
- *Medium:* it contains an interval between the second point that is the starting point of the *Medium* category and it will end at the third point.
- *High:* the third interval includes the range of players whose activities are between the third and the forth point.

Finally, the list of categorized features, which are sorted in importance using Scikit-learn[6] package, will be send to the next component to be considered in the recognition process.

Since, every feature has a different weight in characterizing a play style, we used *Forests of trees* function to recognize this importance. Hence, we exploit ground-truth data set which are labelized manually by the expert researchers (detailed in Section 4.8) to specify the weight of each feature w.r.t the target values (play styles).

In spite of the fact that the framework is applied on each individual player's data to extract the play style, this categorization and extraction is related to the other players' activity. The categorization process conducted off-line that allows having the data of all players at hand. In addition, this may result to provide a ground-truth that enables the system to implement the module on each individual player's data on-line without descritizing the data.

3.4.5 Play Style Recognition

This component concerns the extraction of the play style according to the players' behaviors that are yielded by monitoring their activities throughout the game, and recorded in the form of a vector of features (described in Section 4.8).

As it is shown in [107], play style of the players do not fall into a particular play style and they act differently through the game sessions. For example, a player who is flagged as an Achiever in one session could be changed to an Explorer in the next game session.

Thus, in this study we automate the procedure of recognizing the dynamicity of play style introduced in [107] by designing a Scoring framework that is

[6]To sort the list, we exploit the *forests of trees* function which is available in Scikit-learn [137] in Python.

able to automatically recognize this dynamicity with its confidence value, and uncover all play styles in accordance to which the particular player exhibited during the same game session.

To this end, a *Utility Function* is designed and used to score the play style that she has behaved in each session of the game in a supervised way. Since different features have different impact on characterization of a player's play style, the function utilizes weight and order of computation for each individual feature to calculate the score of each style for a given player. The utility function is defined in the below equation:

$$\varsigma_j^s = \sum_{i=1}^{n} x * wf_i^{s^j} \ni x = \begin{cases} 0 & \text{if } (f_i^{s^j} = \text{``medium''}) \\ 1 & \text{if } (f_i^{s^j} = rs_i^{s^j}) \\ -1 & \text{otherwise} \end{cases} \tag{3.3}$$

where n is the number of features and wf indicates the weight of the feature for the specific style s^j in the given vector. x refers to the statement that whether the features' values are identical in comparison between player's behavior, which is defined by the vector of feature f and the characteristics of a play style defined in the Rule Set rs.

We use "Forests of Trees" [138] function to order the features (in each vector) in importance on an artificial classification task. This function enables the estimation of feature importance on a specific model (in our case study the play style e.g., Achiever, Explorer). Consequently, this importance value is used as weight in the utility function to calculate the score of each play style.

In addition, the framework is equipped with a Confidence Calculator Component (CCC) to measure the confidence value of the retrieved styles for each player. The component uses a type of *Harmonic Series* equation (illustrated in Equation 3.4) to obtain the confidence value of the play styles, where the calculated score "ς^s" is divided by the sum of all weights of the sorted features (in the list) that can be possible to fulfill the requirements of a pure play style.

$$Conf^{s_j} = \varsigma^{s_j} * \frac{1}{\sum_{i=1}^{n} w_i} \tag{3.4}$$

Thus, given the vector of features that reflects the player's behavior in a game session, the algorithm uses an agent to iterate trough the features and compare the values (these values are described later on in the Evaluation section) against the rule set that defines characteristics of all play styles in a supervised manner. If any feature from the vector is identical to any play style's characteristic defined in the rule, the algorithm adds a score for the particular style. The score will be penalized with weight and order of the feature, where the player has behaved exactly the opposite compared to the specified style in the Rule Set. Since there may be a feature with a *medium* value in a vector, the algorithm considers the feature as the neutral feature toward the specific style in the scoring procedure.

Due to the similarity among players' play style characteristics e.g., Careless and Achiever players usually play quickly, or Explorer and Lost players play slowly, it is not possible that a player acts only in a specific style in a particular session. Thus, this scoring system can bring up the styles with their confidence, which enables the framework to recognize which player plays in margin, and which one trends toward a specific style with high confidence. Consequently, to assign a style to a player, the algorithm can pick up a style that has high a score and confidence value.

3.4.6 Play Style Prediction

This component concerns the prediction of the players' play styles for the next session of the game to generate and personalize game content. The action flow is detailed in Figure 3.8, where players' past behaviors are fed to the prediction component to train the model exploiting machine learning classification algorithms. Thus, given the model, the module by feeding the new data of the player is able to predict the play style for the next section or the time-windows of the game. This prediction allows the system to dynamically adapt the game content to the individual player, by taking into account players' past behavior in different sections of the game. A full description of the exploited classification methods can be found in Section 4.10.1.

3.5 Summary

Persuasive systems *particularly Gamification* can be an effective strategy to incentivize healthy demeanor change. The performance of the gamified system might depend on appropriately tailoring the gamification elements to individual players. Hence, the first step in order to personalize game element is to determine the right and important element that can effectively influence players in changing their behaviors under the gamified system. The empirical results obtained and reported in [30,109] show that Challenges have a positive effect on changing players' behavior in the gamification system. Thus, in this Chapter, we first described the notion of gamification platform in smart cities, then introduced the design and integration of the multi-layer framework that automates the generation and recommendation of personalized challenges. We showed the design of our APCGR consisting of three modules: Challenge Generator, Challenge Valuator, and Filtering & Sorting. The proposed APCGR is constructed with the aim of solving two problems; The former is the scaling problem in manually generating game content that the proposed system aimed to overcome such problem in a large-scale gamification system. The latter is a one-size-fits-all strategy for generating game content in a gamified system. Since players are distinct, personalization can help to foster their engagement in gamification.

Then, we presented how we can improve personalization by integrating a Machine Learning module into APCGR, in order to increase the success rate of the recommended challenges during the game. Basically, the module generates a value as the probability of the challenge success (PS), by learning and modeling the players' past behavior (players' feedback) with respect to the given challenges.

In the last section of this Chapter, we have provided a scoring framework in order to extract the players' play styles during the game. This enables the proposed APCGR system to further personalize gamification based on players' traits and personalities. However, this study does not provide how to tailor the game content "Challenges" according to the recognized play style. This work is an essential step towards investigating and providing methods for tailoring game content based on play style to individual players in gamification.

In the next Chapter the implementation, evaluation and results of the designed Automatic Procedural Content Generation and Recommendation, the integration of the proposed Machine Learning module, as well as the Play Style Recognition module are described.

Chapter 4

Evaluation and Results

The previous Chapter has presented the design of the proposed Automatic Procedural Content Generation and Recommendation (APCGR) framework, as well as the integration of the two supplement modules: Machine Learning and Player Recognition. This Chapter describes the implementation, evaluation and results from APCGR and the two integrated modules. The proposed solution was implemented and evaluated in two different gamification scenarios; Urban Mobility system and Educational game-based e-learning. APCGR and the Machine Learning module are assessed, in an Urban Mobility gamification scenario called Trento Play&Go. This evaluation helps determine whether the automatic generation and personalization of the playable content "Challenges" are effective and efficient compared to the traditional manual content generation guided by the expert judgment. Mutually, the evaluation of the Machine Learning (ML) is shaped to assess how ML module can help to enhance the success rate of the recommended challenges that may support players' motivation in game. Player recognition module is implemented, in an educational interactive learning domain, with the aim of assessing the performance of the automatic play style recognition during the game. In addition, the players involved in this experiment presented play styles that varied during the different game phases; thus confirming that player types cannot be considered a static aspect, but a dynamic characteristic of the player that needs to be updated during game-play.

In Section 4.1, we first describe the Play&Go scenario, its' specifications, methods, and tools. Then, the way that players' data is logged and collected in Play&Go is detailed. In the following Section (4.2), we detail the experiment setup of the proposed APCGR which is followed by the evaluation objectives and assessment of the approach. Lastly, we overview the lessons that we learned from this experiment.

Section 4.5 presents the experiment setup, objective of the evaluation and results of the Machine Learning module. Finally, this section concludes with the finding from the evaluation of this experiment.

Section 4.7 shows the evaluation of the Play Style Recognition module. This Section begins with the description of the educational interactive game-based learning scenario called *Solving the Incognitum* [107] and the strategy in collecting players' data. This is followed by the experiment setup, evaluation objectives and results. We close this Section by describing our finding from the results of this experiment.

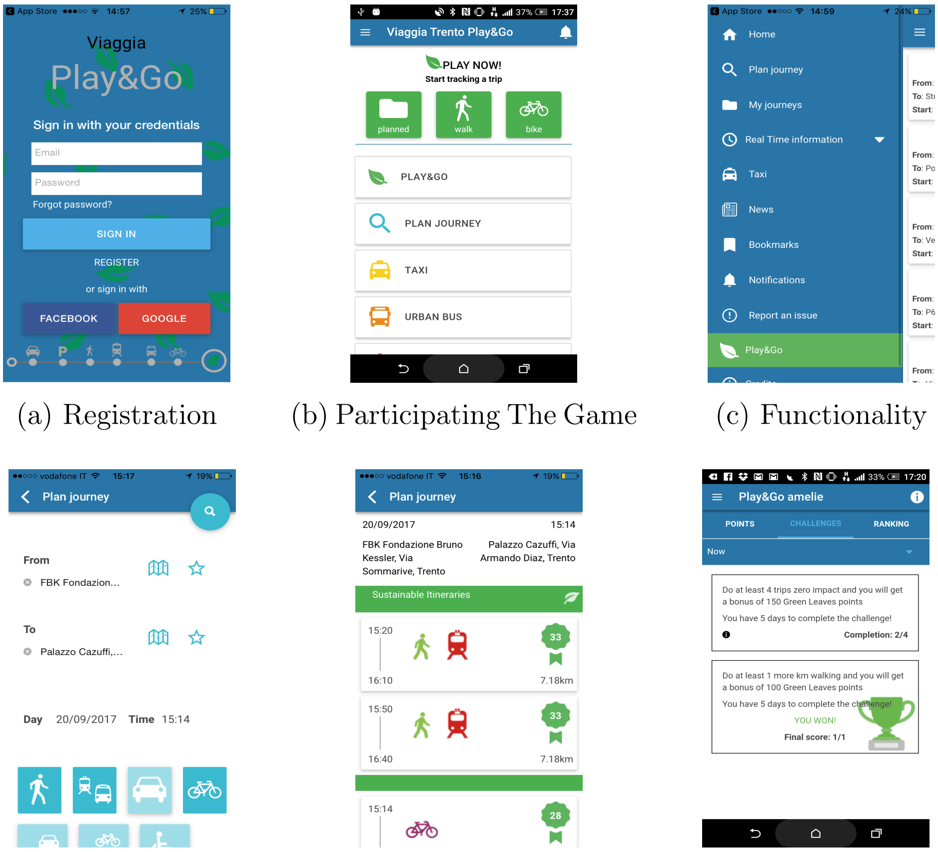
4.1 Play&Go Scenario

Trento Play&Go was a large-scale and long-running open field experiment in the context of gamification that lasted twelve weeks (from September 10 to December 2, 2016). Trento Play&Go pilot has the main goal to evaluate the impact of gamification on citizens' engagement and voluntary travel behavioural changes over a long time frame. Being an open field experiment, Trento Play&Go required a considerable effort for publicity and engagement purposes, which involved –among others– the local administration of Trento, the local associations of enterprises, schools, etc. Advertisements of the experiment have been held on local newspapers, radio, television, and social networks, as well as in physical social locations as bars, local shops, streets and shopping centers. A strong and effective promotion has been done through the participation at Smart City initiatives, such as the Trento Smart City Week (September 10-15, 2016) and the Researcher Night in Trento (September 30, 2016).

4.1.1 Specifications: Methods and Tools

Trento Play&Go game consists in collecting Green Leaves points, which are assigned according to the Km traveled with sustainable transportation means, and with bonuses associated to Zero-impact trips (no CO^2 emissions). In addition, the game supports weekly and global leader-boards for Green Leaves, as well as a variety of badges and badge collections assigned when reaching certain amounts of Green Leaves, or taking specific kind of trips (e.g., the bike aficionado collection assigns badges every 5-10 trips by bike), or exploring some mobility alternatives (e.g., when using a designated Park&Ride facilities for the first time, trying bike-sharing service for the first time, or exploring different bike-sharing stations).

Moreover, the game introduced innovative game mechanics, such as thematic weeks and personalized challenges, implemented into the game with the



(a) Registration (b) Participating The Game (c) Functionality

(d) Making a Plan (e) Possible Options (f) Challenges

Figure 4.1: The Screenshot of the Trento Play&Go Game

goal of maintaining existing users and engaging new ones through a dynamic game. Every week is characterized by a different theme (e.g., bike week, public transportation week, zeroimpact week, to name a few), and personalized challenges (in this dissertation we have focused on this game element which are detailed in the next chapters) are proposed to players on the basis of the theme, their game status and mobility history. Challenges award green leaves bonuses upon completion. For instance, during the bike week, we have different types of challenges promoting the usage of bicycle transport, targeted to different kind of players and personalized to their profile: from challenges pushing players to try the bike / bike-sharing mode (for those who never tried it before) to challenges requiring a small (or significant) improvement in terms of Km and trips travelled by bike or bike-sharing (for players who are already accustomed to use bicycles for their transportation needs).

These combination of transportation modes and the challenges are designed to make the game attractive and playable by newcomers (who are incentivized to compete in the short-term challenges and ranks), as well as to sustain participation of committed players in the long run. Final and weekly prizes which were offered by local sponsors for players who were in top positions of the leader-boards.

Prizes associated to weekly leader-boards give also latecomers a chance to win, as long as they committed for an entire week. Some examples of prizes distributed during Trento Play&Go are yearly subscription to bike-sharing and car-sharing, tickets for music shows, sport events and museums.

4.1.2 Player Experience in Trento Play&Go

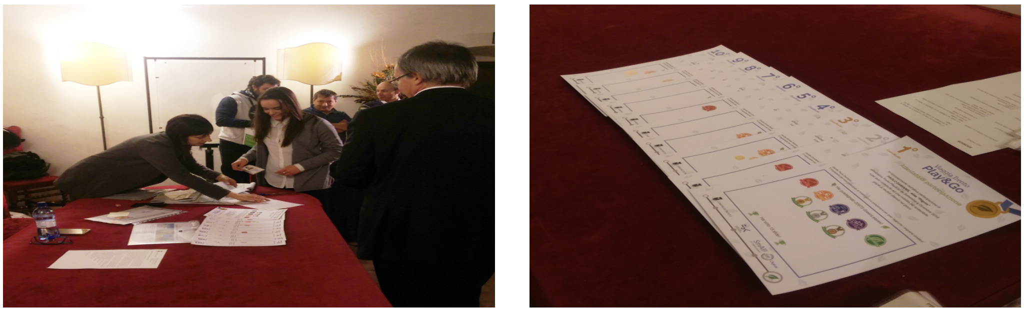
To take part in the game, a player needs to: install the ViaggiaTrento Play&Go App, which is available on Android Play Store and Apple Store[1], register to the game, and use the App for journey planning (Figure 4.1d) and tracking sustainable itinerary choices. The player can use the App also to check her status in the game (e.g., points and badges earned, active challenges, weekly and global leader boards rank), share her results on social networks (i.e. Facebook or Twitter), and inspect the rules of the game and the weekly prizes. Figure 4.1 presents some selected screenshots of the functionality offered by Trento Play&Go App. Trento Play&Go is a travel assistance App that supports the end-user throughout the travel experience such as multi-modal journey planning, real-time notifications, integrated information about all available transport means, and support for recurrent trips.

The App is augmented with a game experience promoting mobility habits that are in-line with city-specific mobility policies and objectives. The main provided game-related functionalities concerns the following:

- Registration of players to the game (see Figure 4.1a),
- Preview of the earned Green Leaves points when planning a multi-modal trip (see Figure 4.1e);
- Inspection of player's results such as Green Leaves points earned, badges and badge collections, active challenges with completion status, weekly and global leader boards rank;
- Possibility of sharing player's own results on Facebook and Twitter;
- Compilation of initial and final surveys and access to game rules and regulation.

At the end of the game, a public event has been organized to present the game results and reward the winners of the weekly and final prizes (see Figure 4.2a). The attendees were players, organizers, city managers and sponsors. Every player received a participation certificate attesting her game results and achievements (see Figure 4.2b).

[1] https://play.google.com/store/apps/details?id=it.smartcommunitylab.viaggiatrento.playgo&hl=en



(a) The Final Event of The Game (b) Winners' Certificates

Figure 4.2: The Final Event of The Trento Play&Go Game and The Certificates

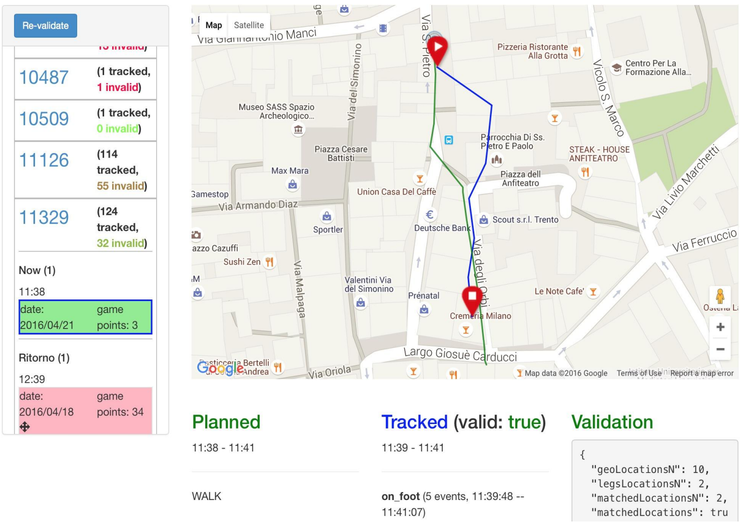


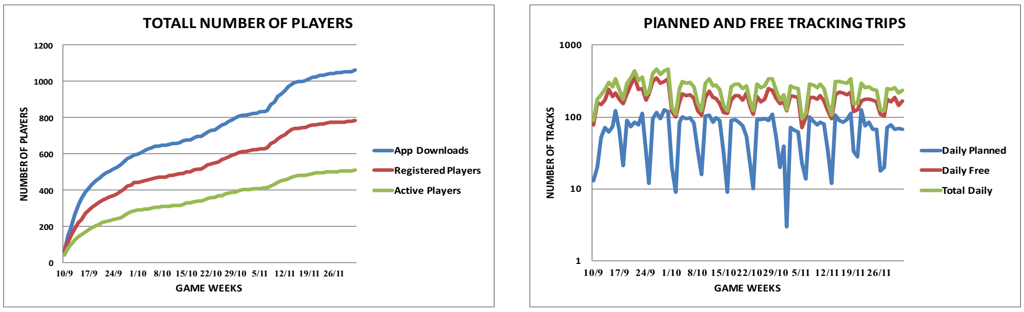
Figure 4.3: Itinerary Validation Console - A Screenshot

Although discussing and detailing the strategies that we used to implement and collect the data in Trento Play&Go is out scope of this dissertation, in this stage of this Chapter we shortly mention how we collected the data, and the general achievement of this gamification scenario.

4.1.3 Data Collection in Play&Go

The data sources that we used to collect information about this pilot and for its evaluation are listed as follows:

- Log data from Viaggia Trento Play&Go App, in particular planned and saved itineraries and tracked itineraries which are depicted in Figure 4.3 (a screenshot of the itinerary validation console),
- Log data from the gamification engine about players' gamified actions and results;



(a) Dynamics of The Game Participation (b) Tracking Trips During The Game

Figure 4.4: The Trento Play&Go Participations and Itineraries

- Baseline questionnaires submitted to *end users* about their mobility behaviour;

- Ex-Post questionnaires submitted to End Users about their mobility behaviour and the game content.

4.1.4 Play&Go Finding

The long-running game Trento Play&Go saw an influx of prospective players throughout its duration, due to the continuous engagement activities put in place throughout the operation of this experiment. Figure 4.4a shows the daily dynamics of the number downloads, registered players, and active players. All in all, we had 785 citizens who registered to the game, and more than 400 active players who recorded their itineraries and actually competed in the game.

If we take a closer look at the distribution of the game actions over the 12 weeks (as reported in Figure 4.4b), we can observe a persistent and continuous participation throughout the game duration. This provides a first positive answer to the challenge of sustaining the participation in a long-running game. It is interesting to observe that, as expected, relative minimum points in the graph (shown in Figure 4.4b) are reached on Sundays and National holidays (absolute minimum on November 1st).

Having in mind the above evaluation and finding that we shortly mentioned in this Section, this dissertation mainly focuses on *Challenge Generator* Component in the Gamification Platform (introduced in Chapter 3.1) by providing the design of an end-to-end Automatic Procedural Content Generation and Recommendation framework. Thus, in the next Section, we detail the description of the study setup, implementation, and assessment of the proposed APCGR.

4.2 APCGR Experiment Setup and Evaluation

During the twelve game weeks, 1061 citizens downloaded the App, 785 registered to the game, and 410 actively participated in it (*active players* from now on). Within the twelve weeks, we collected more than 20K trip traces, which were tracked by the players' App and validated (in terms of mode and path) through a semi-automated itinerary validation system. Those traces enabled us to collect detailed statistics about the itineraries of App users when playing the game, such as, trips, trip legs and Km traveled in each transport mode.

During Trento Play&Go, we carried out an A/B test in the last three weeks of the game, with the aim of comparing our system that automatically generates, and proposes personalized challenges to the players (*RS challenges*) with a semi-automatic approach, used throughout the twelve weeks of the game, were challenges were first decided and administrated by the game designers using their expert judgments and then injected in the game rule set (*non–RS challenges*). For simplicity, we use *RS challenges* and *non-RS challenges* labels throughout this dissertation, for the challenges which are generated by the automatic framework and the semi-automatic approach, respectively. In that semi-automatic approach, we have developed a frontend tool for the challenge generator, which is shown in Figure 4.5. In the tool, the designer can pick what models she wants to instantiate, specify values for their parameter sets, and define which assignment criteria must be applied. In the example re-

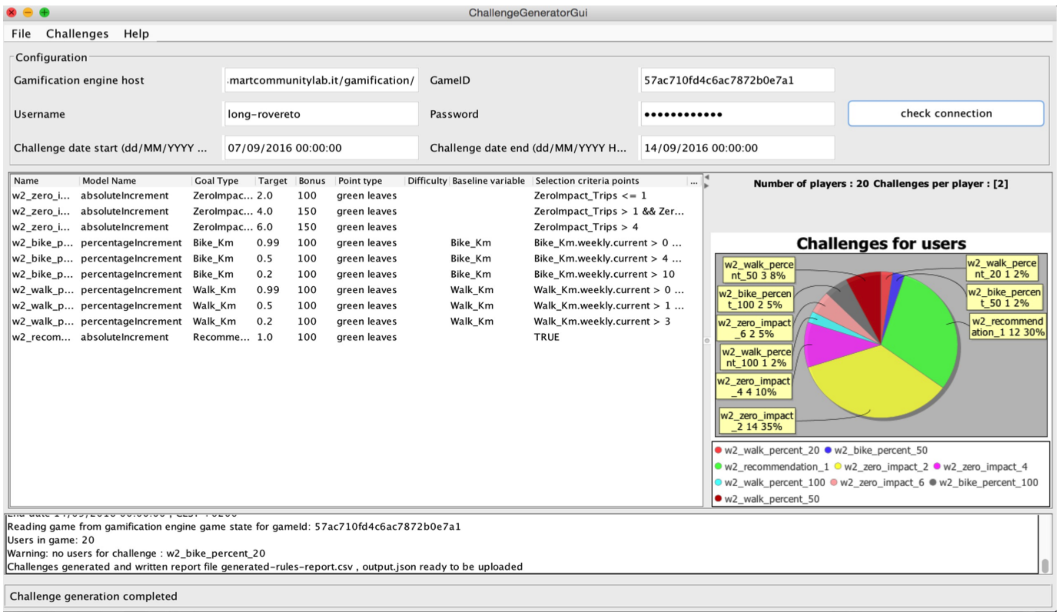


Figure 4.5: The Challenge Generator Frontend.

ported in the Figure, only two models are shown: *percentageIncrement*, which sets an improvement goal relative to one's past performance, and *absoluteIn-*

crement, which asks to reach a set performance goal. In this approach also, the game designer can achieve a lot of variation by applying different value combinations for the parameter set of each model. For example, the *absoluteIncrement* model can be used to provide various goals on the number of zero–impact weekly trips; but it can be used as well to ask to every participant to refer the game to at least one potential new player. Further variation and personalization come from specifying what assignment criteria – logical clauses that predicate on the game state and profiling variables collected for each player – will decide the assignment of each parameterized model to different segment of the player population.

On the right-hand-side of the tool, a chart representing the distribution of challenges over the player population enables the game designer to understand the effect of the selection criteria she has written. The game designer may go through some iterations of this specification process, and try the different assignment options, until she is:

- Certain that each player will receive the planned number of challenges (2, in our case),

- The challenge distribution reflects well the behaviors that the game wants to push at that time. For example in accord to the theme of the week.

At that point, she can decide to instantiate all challenges, which results in the APCGR of new game code and its deployment onto the run-time of our gamification framework. The new code applies uniquely to each individual player, and – as outlined above – differentiates her game experience using her current game state, as well as her accumulated game performance, and her track record in terms of gamified behaviors.

It is worth noticing how this semi–automatic process can be fairly time–consuming for game designers. In addition, they must gather significant experience with the gamified domain and the game itself, in order to produce challenge assignments that can be well accepted by the most players and enhance their game experience, and at the same time push them to further the underlying goals of the gamification campaign. Moreover, in games with large number of players an approach that strongly relies on this kind of expert judgment is hard to scale.

During the three weeks of the A/B test, our new APCG– and RS–based challenge generation system assigned 220 personalized challenges to 82 unique players (see Table 4.1). *RS players*, refer to the players who received RS challenges, thus for simplicity we use RS players throughput this dissertation. Among those RS players, 60 were active in those three weeks, with 164 assigned challenges. The Challenge Generator used the *percentageIncrement* and *absoluteIncrement* challenge templates, and applied them to an array of transport modes, covered by the following mobility indica-

Table 4.1: Numbers of RS and Non-RS Players, and the Challenges

	RS (in 3 weeks)	Non-RS (in 12 weeks)
# of Unique Players	82	397
# of Active Players	60	372
# Challenges	220	3333

tors: *{Train_ Trips, Bus_ Trips, ZeroImpact_ Trips, Walk_ Trips, Walk_ Km, Bike_ Km, BikeSharing_ Km}*.

RS players were randomly selected from a subset of the whole players' population, from which we excluded players who had very high performance in the previous week, whom we called "*weekly champions*", and players who were *not active* in the previous week. This selection of RS players was repeated every week, such that the players who were selected to receive RS challenges in a certain week, might or might not receive RS challenges in the following weeks of the A/B test. We reason this randomly selection to sustain the population of RS players for all the three weeks, since the numbers of active players change over the weeks (it may increase or decrease for any reason), the proportion of the RS players may decrease for the following week to receive the new RS challenges. Those reasons could be the freshmen players who are entered in the middle of the game – for increasing the number of players – or external reasons (described in Chapter 3.2.3)– for participating less number of players w.r.t the previous game week.

4.3 APCGR Evaluation Objectives

To assess the effectiveness of the automatically generated and personalized challenges, we have looked at three complementary aspects, which are reflected in the following evaluation objectives:

- **Objective 1:** *How does the player acceptance rate for RS challenges compare to the acceptance rate of the same types of challenges assigned via expert judgment (Non-RS challenges)?*

Objective 1 (Ob1) discusses to the issue of user experience, since recommending playable units of contents that may be diverse, but are all in all well accepted, by each individual player, is the pre–requisite to be able to leverage them as a mechanism for enhanced engagement and retainment. As a proxy measure of acceptance, we consider challenge completion rate. For example, the proportion of challenges that players completed successfully, and as a proxy measure of not-accepted, we take into account the challenges which are not completer or not even started.

- **Objective 2:** *How does the improvement recorded on the target indicators for the RS challenges compare to the improvement recorded for the challenges assigned via expert judgment (Non-RS challenges)?*

Objective 2 (Ob2) points to the efficacy of procedurally generated challenges as persuasive mechanisms that can appreciably impact players' behavior in a gamification campaign. To measure challenge–induced behavioral change, we consider those challenges whose goal requires to improve by a certain percentage the player's performance for a given mobility mode with respect to the previous game week.

- **Objective 3:** *How do the rewards computed for RS–generated challenges compare to those of challenges assigned via expert judgment, in terms of their adequacy to the challenge difficulty and the amount of behavioral improvement requested?*

Objective 3 (Ob3) speaks to whether our system assigns commensurate and balanced rewards for the challenges that automatically proposes to players. This is an important consideration for game designers, as well as any entity promoting a gamification campaign. On the one hand, the reward should be valuable enough to induce the player to make an effort and complete the challenge; on the other hand, it should not be excessive, because that leads to inadequate, sub–optimal exchanges between behavioral improvement and incentives, and is effectively equivalent to assigning challenges that are too easy. To measure this adequacy, we compute the ratio between the amount of improvement generated by challenges, and the amount of rewards "paid" by the game to players for challenge completion.

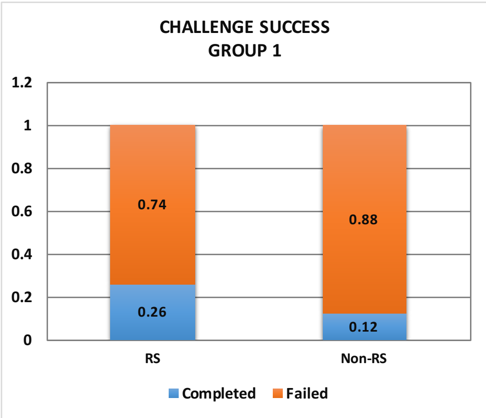
In the next Section we have demonstrated the evaluation of the objectives, as well as reporting in detail the results that we obtained in all experiments.

4.4 APCGR Results

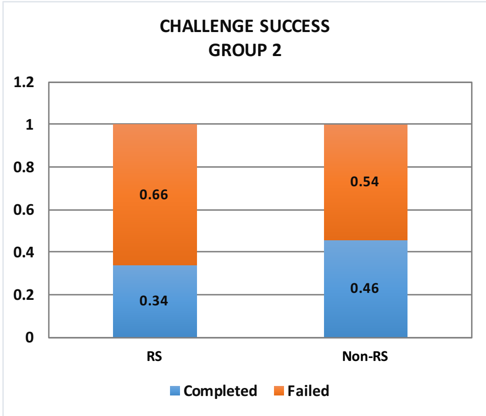
4.4.1 Objective 1 Evaluation

To evaluate Objective 1 (Ob1), we have compared the proportion of players' success in automatically generated and recommended challenges vs. challenges administered through expert judgment. In this analysis, we consider the 82 unique RS players, who received 220 challenges from our system during our A/B test in weeks 10, 11, and 12 of the Trento Play&Go game, and compare their challenges' completion rate to that of the following groups.

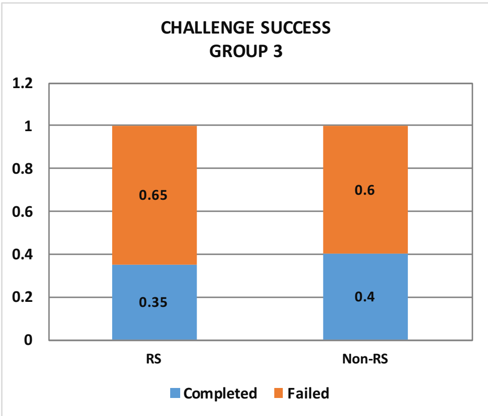
- Group 1: Non–RS players active at any point in the game, who were given in game weeks 10, 11 and 12 mobility challenges that are analogous to the RS challenges. These types of challenges are: *percentageIncrement* or *absoluteIncrement*, targeting the same set of transport modes covered in RS challenges. In total, players in Group 1 were given 1296 such challenges, and the results of this comparison is shown in Figure 4.6a.



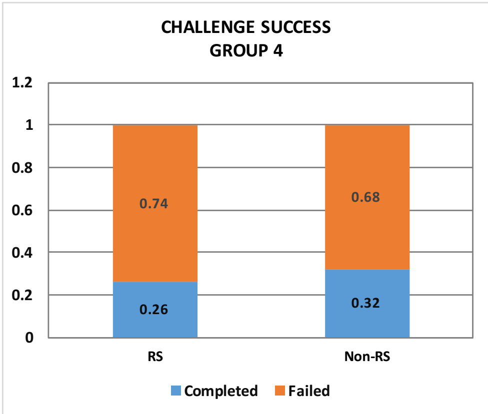
(a) Group 1



(b) Group 2



(c) Group 3



(d) Group 4

Figure 4.6: The Overall Challenge Acceptance - Group 1, 2, 3 and 4.

- Group 2: A subset of Group 1, including non–RS players who were active specifically in the game weeks 10 to 12, which were covered during the A/B test. They received 333 mobility challenges analogous to RS challenges. Figure 4.6b represents the evaluation of this comparison.

- Group 3: A subset of Group 2, from which we excluded the top performers (typically top 10 to 12 players) of the week before that of challenge assignment (that is, weeks 9 to 11). This is the group that serves effectively as the **control group in our A/B test**, since – by eliminating these "weekly champions" – Group 3 reflects the same population from which we drew the RS players each week, as explained in Section 4.2. Players in Group 3 were assigned 280 mobility challenges analogous to RS challenges.

Table 4.2: Proportion Tests for Challenge Success Rates.

	Group 1				**Group 2**				**Group 3**				**Group 4**			
T-Ch	RS		Non-RS		RS		Non-RS		RS		Non-RS		RS		Non-RS	
Status	C	T	C	T	C	T	C	T	C	T	C	T	C	T	C	T
# Challenges	58	220	153	1296	58	164	153	333	58	164	113	280	58	220	49	151
AC Ratio	26%		12%		35%		46%		35%		40%		26%		32%	

T-Ch: Type of challenges such as *RS* and *Non-RS* challenges.
AC Ratio: Acceptance Ratio.
C: Total numbers of the *completed* challenges.
T: Total numbers of the *recommended* challenges.

- Group 4: Finally, we examined the same 82 RS players "against themselves", that is, with respect to those challenges, which they received in weeks 10 to 12, but which were assigned to them by expert judgment, as opposed through our system. This challenge set included 151 mobility challenges analogous to the RS challenges. This examination is illustrated in Figure 4.6d.

In Table 4.2 we detail the results of a statistical test of proportion for the challenge success rate of RS players on RS challenges vs. the four groups defined above.

Notice that, vis–a–vis Group 1, we tested whether RS players had a *higher* success rate than the other players in general. That test is useful mostly as a sanity check, since a considerable number of players in Group 1 may not have been active through the three weeks. Given that, it should not be surprising that RS challenges enjoy a better, and highly statistically significant, completion rate. Instead, vis–a–vis the other groups, our hypothesis is that the RS challenge assignments should not differ statistically from the challenges assigned via expert judgment; therefore we tested whether RS players had a statistically equivalent success rate compared to the other players.

It needs to be mentioned that in the tests vs. Group 2 and Group 3, we had to eliminate 56 RS challenges, which were proposed to RS players who then chose not to be active (did not participate at all) to the game in the corresponding week. We did that to keep the testing conditions and the data sets congruent, since Group 2 and Group 3 contain only non–RS players who were *active* in the game weeks of the A/B test.

We have exploited the equivalency testing to indicate that the differences that do exist between the above groups are small enough for practical purposes [139]. To this end, we took the advantage of two one-sided test ($TOST$)[2] procedure to illustrate that the means in two population are close enough to be considered equivalent [140, 141]. Table 4.3 shows the properties of the test to assess the equivalency of the above groups.

We have set the confidence level of the interval (CI) value to 0.95 and $\alpha = 0.05$, which indicate a range of computed values that likely contain the true value of the parameter with a certain level of confidence 95% [142]. To

[2]We used tost function which is available in TOSTER package in r: `https://cran.r-project.org/web/packages/equivalence/equivalence.pdf`

Table 4.3: Equivalence Test in Different Groups.

	Group 1 CI=0.95, $\alpha = 0.05$		Group 2 CI=0.95, $\alpha = 0.05$		Group 3 CI=0.95, $\alpha = 0.05$		Group 4 CI=0.95, $\alpha = 0.05$	
Epsilon: ϵ	**Tost p-value**	**H0**	**Tost p-value**	**H0**	**Tost p-value**	**H0**	**Tost p-value**	**H0**
0.50	5.5652e-25	rejected	3.11723e-1	rejected	2.42267e-1	rejected	2.48866e-1	rejected
0.25	0.0004510	rejected	0.0010168	rejected	1.6678e-05	rejected	0.000103	rejected
0.15	0.4435465	not rejected	0.1706046	not rejected	0.01809221	rejected	0.04531761	rejected
0.1	0.9280577	not rejected	0.549738	not rejected	0.1466714	not rejected	0.2520635	not rejected

select the largest difference between the population' means (Epsilon ϵ) that can reasonably be considered equivalent, prior study or the knowledge of experts are needed. This is because, choosing the right number comes down to what the experts think makes a reasonable case, which differs from domain to domain. Hence, to know the right value, we run the equivalence test with epsilon (ϵ) in the range $[0.5 - 0.1]$, where the close this value to 0, the large of sample size is needed to attain a proper narrow confidence interval in order to conclude that the obtained estimate is statistically equivalent.

As it shown in the table, epsilon with 0.5 and 0.25 are not enough strong to know which proportions of the challenge success are equivalent, since the test rejects the null hypothesis in all cases that concludes all groups are equivalent, which is not practically correct.

We proceeded the test by setting the *Epsilon* with tinier value $\epsilon = 0.15$ and observed the reasonable results. In this setting, only non-Rs challenges in Group 2 performs better, since we fail to reject the null hypothesis by $p - value = 0.170606$ at 95% confidence. It is worth remarking that the player's population in Group 2 is not fully equivalent to the population from which we drew our RS players, since Group 2 *does include* weekly champions. Therefore, we had expected that Group 2 might perform overall somewhat better; in fact, weekly champions were responsible for 40 out of 153 challenges successes recorded for Group 2. The equivalence tests of RS challenges vs. Group 3 (i.e., the A/B set up) – as well as Group 4 – show that when samples are drawn from fully equivalent player populations, we can reject the null hypothesis at 95% confidence, which highlight that RS and non-RS challenges in these two groups are statistically equivalent, or in other words, the difference is minor.

This equivalency conclusion turned to existing a relevant difference between RS and non-RS challenges in Groups 2 to 4, when we decrease the magnitude of the region of similarity e.g., $\epsilon = 0.1$, by failing to reject the null-hypothesis.

Those equivalency tests, therefore, provide no support for a statistical difference for player groups of this magnitude, if we could define a reasonable interval of equivalency. Therefore, we can answer Ob1 by stating that:

> The difference between the challenge success rate of RS and non-RS challenges are statistically trivial if the margin of equivalency is defined reasonably.

4.4.2 Objective 2 Evaluation

To evaluate Objective 2 (Ob2), we define Improvement (Impr) induced during a challenge relatively to the player's performance of the previous game period (in our case,"*a Week*"). Given a performance indicator congruent with the challenge goal (in our mobility game: "either number of trips or amount of Km traveled in a certain transport mode in a week"), we normalized the definition of improvement by Equation 4.1.

$$Impr = (counter - base)/base \tag{4.1}$$

where *counter* is the numeric value of the performance indicator that starts from 0 to $+\infty$, and *base* is the value of the same indicator sampled at the moment in which the challenge has been administered to the player (i.e., at the end of the previous game week). According to the formula above, improvement (Imp) is a real number in the range $[-1, +\infty]$ (Figure 4.8 and 4.9 y axes), where the range between -1 and 0 indicates that the player has performed *worse* during the challenge period, compared to the previous game week. More specifically, -1 means *nothing* was done by the player related to the challenge goal, while 0 means that the player had *no improvement*. It means, she repeated the exact same performance of the previous game week (highlighted in yellow in Figures 4.9 and 4.8).

We have considered weekly challenges of type *percentageIncrement*, since their goal is exactly to improve by some given percentage one's performance on a mobility indicator with respect to the previous week. Out of the 164 challenges automatically administered by our system during the A/B test to those players who were active in game weeks 10, to 12, 129 were of type *percentageIncrement*. Hence, the rest were of type *absoluteIncrement* and, in particular, those were all what we call "try mode" challenges, that is, they asked to players to do at least a single trip in a mode they had not used at all the previous week. Although 17 out of those 35 challenges were completed, their definition is clearly not conducive to measure relative improvement in a way that is congruent with our definition above, and also with the *percentageImprovement* challenges. Therefore, we have excluded them from our analysis of Ob2, and considered non-RS challenges from the control Group 3 "Only *percentageIncrement* (249)".

The *percentageIncrement* RS challenges were subdivided in the following way:

- Improvement on number of trips per week: 71 challenges including, *Train_ Trips* with 12, *Bus_ Trips* with 36, and *ZeroImpact_ Trips* with 23.

- Improvement on amount of Km per week: 58 challenges consisting, *Walk_ Km* with 44 and *Bike_ Km* with 14.

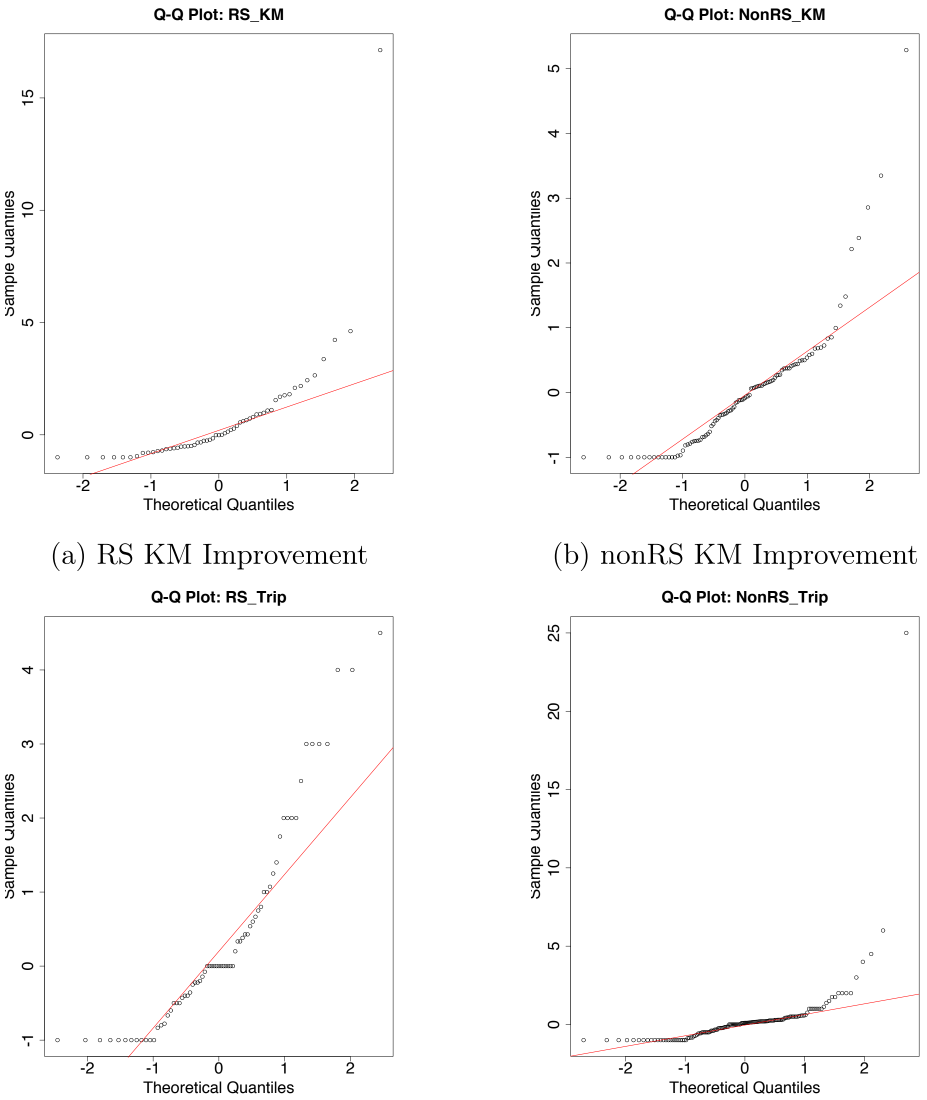
- The challenge generator also produced candidate *percentageIncrement* challenges for modes *Walk_ Trips* and *BikeSharing_ Km*, but they were never picked by the RS.

We compare the improvement induced by those RS challenges against equivalent non–RS challenges. For example, *percentageImprovement* challenges covering the same set of transport modes, which were administered during the three weeks (10, 11 and 12). There were 145 such challenges predicating on number of Trips, and 104 on amount of Km, for a total of 249 non-RS challenges. In order to assess the difference in improvement between the RS and non–RS regimes, we sorted the normalized improvement metric for each challenge in ascending order, and plot this sequence of values at equally spaced intervals. We show those plots in Figures 4.8, and 4.9, separately for trips–related and Km–related challenges respectively. In those Figures, the y axis represents the improvement metric, and the x axis the percentage of challenges having that improvement value, or less; the blue curve plots the results of the non–RS challenges, while the orange curve those of RS challenges. The amount of improvement collectively induced by those challenges can be visualized as the area in the chart comprised between those monotonic non-decreasing curves and the y=0 axis (i.e., the *no improvement axis*). In fact, actual (positive) improvement occurs only in the area to the right of the curve intercept with the y=0 axis (highlighted in green), while the area to the left of the intercept represents those cases in which we observed worse performance than the challenge baseline; that can be seen as a *negative improvement* area (highlighted in red). In both Figures, it is easy to appreciate that, while the negative improvement areas of the RS and non-RS curves are almost completely overlapping, the positive improvement areas under the blue curve are larger and contain almost everywhere the corresponding areas under the black curve.

To go beyond this intuitive assessment, we can also quantify the amount of improvement, by using a method for numerical integration of those curves, and estimate the size of those areas[3]. We call this metric the Area Under the improvement Curve, or AUiC[4]. We denote instead AUiC+ the estimate of the area of positive improvement only. Values for AUiC and AUiC+ for non-RS

[3]For those estimates, we took advantage of a function made available to the R statistical suite by [143].

[4]In order to avoid confusion with AUC, which usually indicates the Area Under the Curve in a Receiver Operating Characteristic (ROC) plot.



(a) RS KM Improvement

(b) nonRS KM Improvement

(c) RS Trips Improvement

(d) nonRS Trips Improvement

Figure 4.7: QQ Plot: KM and Trips in Group 3

and RS challenges are reported in Table 4.4 And Table 4.5 (separately for trips–related and Km-related challenges, as well as for each single transport mode). In almost all cases, the values of both AUiC and AUiC+ metrics are quite larger for RS challenges, with the single exception of *Zero impact* challenges, for which the AUiC value is somewhat larger in the non-RS case, and the AUiC+ values are very similar.

It is noticeable how in several cases the total AUiC in the non–RS case is negative; that is due to many non–RS players not doing enough to reach their baseline of the previous week, or choosing not to take up that specific challenge at all, thus offsetting the actual improvement (AUiC+) from other players in the same group.

		RS challenges					
		n (won)	rewards paid	AUiC	AUiC+	players improved	reward / AUiC+ per capita
Trips	**Tot**	**71 (27)**	**5650**	**0.385**	**0.643**	**30**	**293**
	Train	12 (7)	1470	0.672	0.897	7	234
	Bus	36 (13)	2860	0.204	0.536	13	410
	Zero impact	23 (7)	1320	0.365	0.527	10	250
	Walk	0	0	n/a	n/a	0	n/a
Km	**Tot**	**58 (23)**	**5410**	**0.305**	**0.612**	**26**	**340**
	Walk	44 (15)	3520	0.222	0.545	17	380
	Bike	14 (8)	1890	0.516	0.729	9	288
	Bike Sharing	0	0	n/a	n/a	0	n/a

Table 4.4: RS Challenge Improvement

		Non–RS challenges					
		n (won)	rewards paid	AUiC	AUiC+	players improved	reward / AUiC+ per capita
Trips	**Tot**	**145 (69)**	**15600**	**0.101**	**0.366**	**76**	**561**
	Train	31 (13)	3400	-0.237	0.147	13	1779
	Bus	52 (19)	5250	-0.135	0.264	21	947
	Zero impact	57 (34)	6200	0.466	0.532	39	299
	Walk	5 (3)	750			3	
Km	**Tot**	**104 (42)**	**11300**	**-0.017**	**0.301**	**48**	**782**
	Walk	80 (30)	8150	-0.006	0.314	36	721
	Bike	17 (8)	2100	-0.089	0.204	8	1287
	Bike Sharing	7 (4)	1050			4	

Table 4.5: Non–RS Challenge Improvements.

We can also look at the significance of the improvement differences signified by the AUiC and AUiC+ metric. Since the distribution of the data will determine the statistical procedure, we looked the QQ plot for that group (Group 3) to know the normality of the data [144]. Figures 4.7 shows that the data is not normally distributed, even they are not close to normal. Hence, we selected and applied the Wilcoxon test [145] to the RS and non–RS improvement data for trip–related and Km–related challenges (the same data sets visualized in Figures 4.8, and 4.9, respectively). Wilcoxon is a non–parametric test of the probability that randomly selected values from one data set could also belong to a second data set; we used it to investigate whether the improvements from RS challenges are statistically larger than those from non–RS challenges.

The results of the Wilcoxon test for Km–related improvement are:

```
W = 3280.5 p-value = 0.1317
```

The result of the Wilcoxon test for trip–related improvement are:

```
W = 5344 p-value = 0.2941
```

In both cases the p–value is too large to indicate that the RS improvement (and therefore the corresponding AUiC score) are significantly larger than the non–RS improvement. However, the observant reader looking at Figures 4.8, and 4.9 may notice that the plots are very similar in the negative improvement areas (for example, 15% to 20% of the data consists of -1 values), and start to divaricate after they reach the intercept with the y=0 axis. In fact, if we repeat the Wilcoxon test only for the data that represents positive improvement, we obtain, for Km–related challenges:

```
W = 571 p-value = 0.0003194
```

and for trip–related challenges:

```
W = 979.5 p-value = 6.549e-06
```

That means that the difference in the AUiC+ scores are highly statistically significant. In turn, that suggests that automatically generated challenges may have worked better for those players who embraced their assigned challenge goal, elevated their game, and put in place a positive effort to improve to some extent on their baseline mobility behavior. One possible interpretation is that the selection of transport modes and goals proposed by our recommendation system could have been better suited to the personal inclination of the individual players than the selection proposed through expert judgment to the non–RS players, making more congenial for many RS players to approach, reach or even go beyond the goals set in their personalized challenges.

To have an in-depth look at the normalized improvement of the players' performances, we have focused on each individual mode, and then described them in: *Week by Week Overall*, and finally demonstrated and compared all the modes vs. together *Week by Week* (proportion).

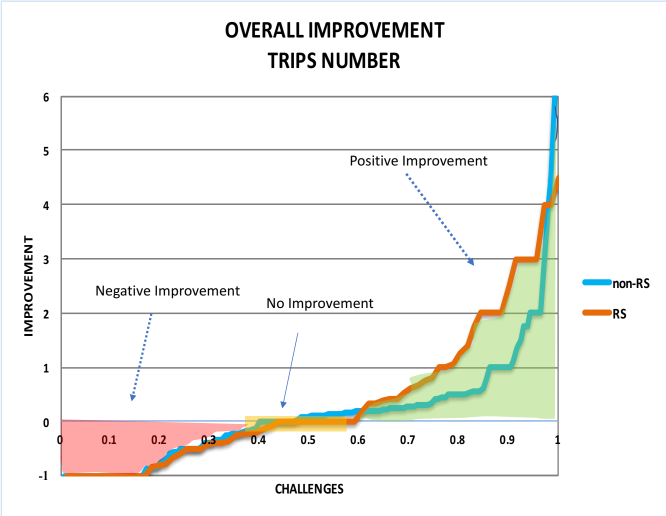


Figure 4.8: Improvement Induced by Challenges: Trips Number (all modes).

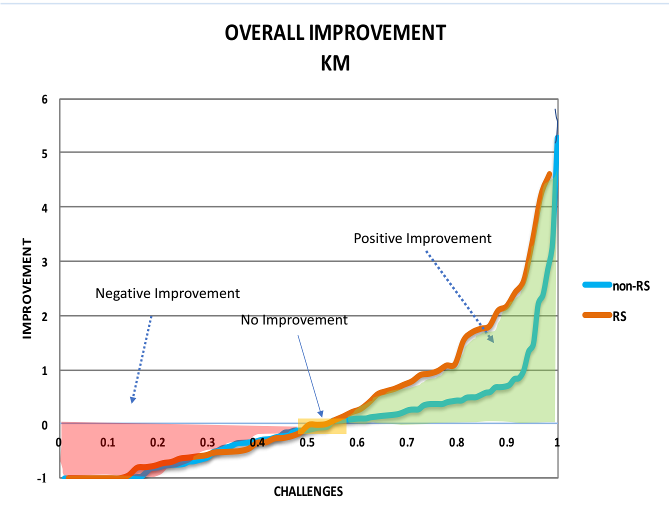
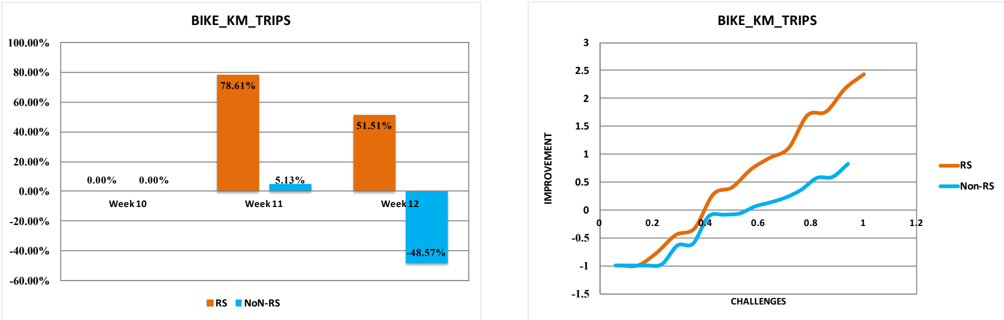


Figure 4.9: Improvement Induced by Challenges: Km Travelled (all modes).

- Bike Trip: As it is shown in Figure 4.10a, no performance was recorded in week 10, neither for RS challenges, nor non-RS challenges which express the players did not receive any Bike-related challenges in the week that may have happened due to the different theme week. While in the next week (week 11) there is a huge improvement in RS challenges by more than 78% vs. only 5.13% in non-RS challenges. However, the improvement in RS challenges declined from 78% to 50% in week 12, the gap between the improvement of the two methods increased, because the large numbers of players who received non-RS challenges did not repeat even their previous performances, thus the negative performance recorded for the non-RS challenges. The normalization of these perfor-

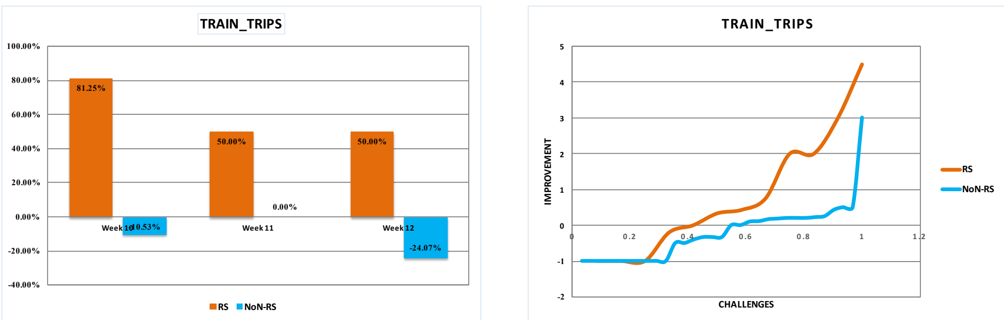


(a) Week by Week (b) Normalized Improvement

Figure 4.10: The Bike Improvement

mances for both RS-challenges and non-RS challenges is illustrated in Figure 4.10b, which shows the positive effect of RS challenges on players to progress their activities in that particular mode.

- Train Trip: Figure 4.11a and 4.11b represent the improvement of the train trips for the RS challenges vs. non-RS challenges which were recorded in the three weeks of the game. Tracing all the three weeks, a remarkable improvement observed in week 10 by more than 81% w.r.t the previous records for the similar means for the RS challenges. However, in the subsequent weeks less improvement were recorded, the amount of improvement (with 50%) in the week 11 and 12 are analogues and still significant. In contrast with the RS challenges, the negative performance were observed in week 10 and the last week of the game for the non-RS challenges. The normalized improvement also shows the significant gap between the effect of the two types of challenges on the players' progress by quantifying the differences with $AUiC$=*0.672* for RS challenges vs. $AUiC$=*-0.237* for non-RS challenges.

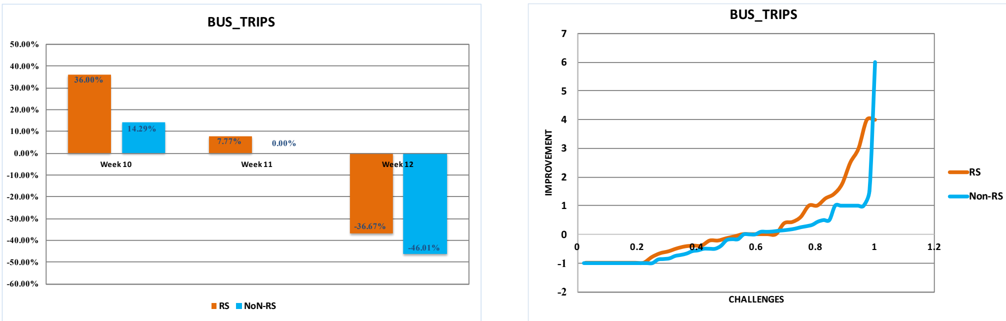


(a) Week by Week (b) Normalized Improvement

Figure 4.11: Train Improvement

- Bus Trip: Looking the players' performance in bus trips which is shown in Figure 4.12, we have found considerable improvements in RS and non-

RS challenges with 36% and 14.29% in the first week of the A/B test (see 4.12a), respectively. These improvements dramatically decreased to ∼2% and 0% in the next week of the game, for the same mode and the same types of the challenges, respectively (it is noticeable that in week 11 there were not any non-RS challenges related to the bus trips). Unlike the first week, in the last week of the game we recorded the negative improvements in both types. This might have happened in the case of the players were not well-enough motivated to perform at least their previous performances. By excluding the negative figures and normalizing the only improvement in the three weeks, we achieved *AUiC=0.210* for RS challenges vs. *AUiC=-0.135* for non-RS challenges which are plotted in Figure 4.12b.

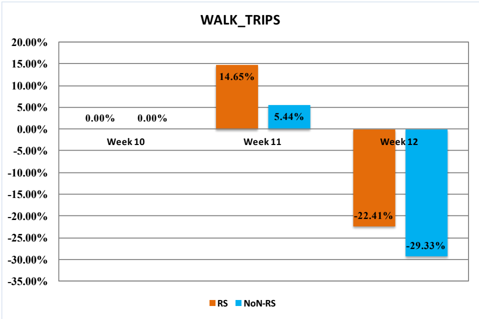


(a) Week by Week

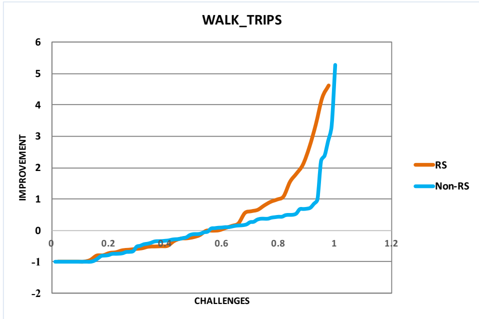
(b) Normalized Improvement

Figure 4.12: Bus Trips Improvement

- Walk: Unlike the improvement recorded for the different transportation modes mentioned above, in this mode there is only a positive improvement in the second week of the A/B test (week 11) that is represented in Figure 4.13a in week by week comparison, with around ∼15% vs. ∼6% in RS challenges and non-RS challenges, respectively. Referring the above comparisons, the similar behavior happened in this mode as well, by recording the negative performances for the last week of the A/B test that could have happened due to the several reasons which are detailed in the next Section. For this experiment we also quantify the difference by calculating the Area Under the improvement Curve (AUiC) as follows; *AUiC=0.222* for the RS challenges, and *AUiC=0.006* for the non-RS challenges. In addition, by taking into account only the players who improved their performances (w.r.t their previous week performances), we obtained *AUiC+=0.545* and *AUiC+=0.314* for RS challenges and non-RS challenges, respectively. These differences indicate that RS challenges are more effective than non-RS challenges, even in the case that we excluded the players who were not active in that particular mode.



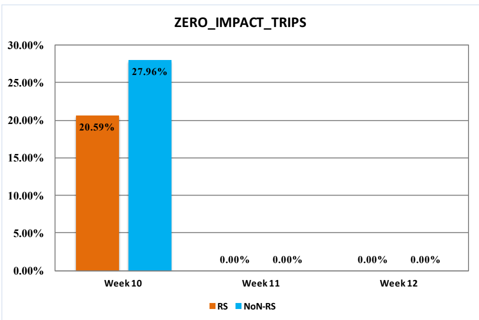
(a) Week by Week



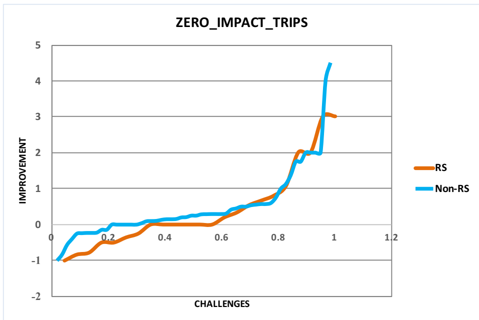
(b) Normalized Improvement

Figure 4.13: Walk Trips Improvement

- ZeroImpact Trips: The only case in which non–RS challenges perform better is ZeroImpact_Trips depicted in Figure 4.14a. This positive performance captured only in the first week of the A/B test. Since the AUiC+ values in this case are almost the same, the difference is mostly due to less of a negative AUiC contribution in the non–RS challenges case. This means that a larger number among RS players did less than their previous week baseline limited to those specific zero–impact challenges.



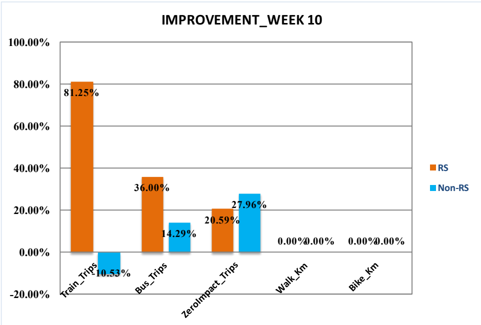
(a) Week by Week



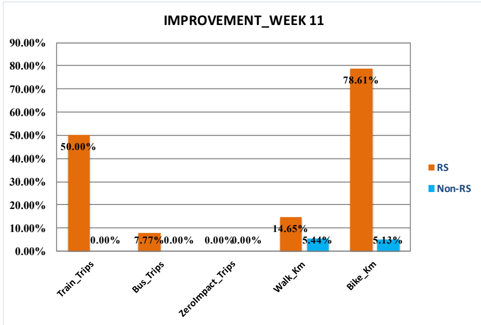
(b) Normalized Improvement

Figure 4.14: ZeroImpact Trips Improvement

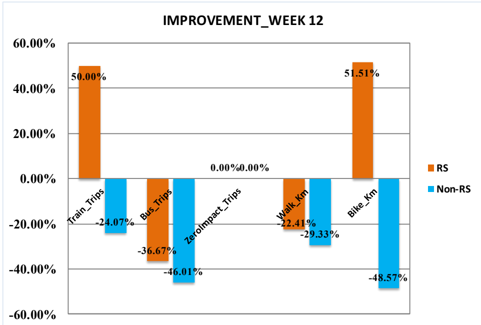
Finally, to compare all the transportation mode's contributions for both RS challenges vs. non-RS challenges we have plotted them week by week which are shown in Figure 4.15. It is worth mentioning that only ZeroImpact_Trips in week 10 from non-RS challenges outperformed the RS challenges, and the rest in all weeks (by excluding the cases that in both challenges negative contribution were recorded) RS challenges had more effect on persuading players to progress their performances. It is also shown that in week 12 except Bike and Train trips, we recorded negative performances for both RS challenges and non-RS challenges.



(a) Week 10



(b) Week 11



(c) Week 12

Figure 4.15: Weekly Improvement in the Three Weeks

On the basis of the data described above (see Table 4.4 and Table 4.5 as well), we can thus answer Ob2 by stating that:

Challenges that automatically generated and assigned by our system may be conducive to higher level of improvement than analogous challenges assigned through expert judgment.

4.4.3 Objective 3 Evaluation

To evaluate Objective 3 (Ob3), we put in relation the amount of improvement for the various challenge types to the rewards "paid" by the game for challenge completion.

Operationally, we characterize improvement via the data about AUiC+ in Table 4.4 and 4.5, since otherwise the total AUiC of non–RS challenges would often be negative. That means that we considered only the performance of those players who were able to achieve *some* amount of actual improvement in their challenges. Moreover, since the cardinality of the player sets yielding some improvement is different in RS vs. non–RS cases, we further normalize the data considering only the number of players who contributed to that improvement (i.e., the *players improved* column in Table 4.4); we thus compute the *per capita* reward paid by the system per unit of AUiC+ ($Reward_{pc}$). The corresponding formula is as follows:

$$Reward_{pc} = (Reward_{tot}/players_{imp})/AUiC+ \tag{4.2}$$

By looking at that data, we can clearly see that the challenge proposals by our system are across the board *more economical* in terms of rewards paid as incentives per unit of improvement. The difference is always in favor of RS challenges, and it is quite evident. In fact, the rewards paid to non–RS players per unit of improvement are – almost in all cases – about double of those paid to RS players, or more. For example, the per capita reward paid per unit of improvement in non–RS trip–based challenges is 1.91 times higher (561 / 293) than in the RS case, while in non–RS Km–based challenges is 2.3 times higher (782 / 340) than in the RS case.

Therefore, we can answer Ob3 by stating that:

> Challenges assigned by our system may yield better improvement for the same amount of per capita reward (or, they yield the same improvement for less reward.

Taking into account the results observed for Obj1, Obj2 and Obj3, we can claim that the proposed APCGR framework successfully addresses these objectives and thus contributes to solve the problems highlighted in Chapter 1.

In the next Section, we discuss our finding at extending our approach with an automated learning algorithm, which will augment our recommendation component, in order to optimize and tune the selection of challenges to the individual player based on her track record with challenges proposed in the past.

4.4.4 Lessons Learned

Empirical results from games promoting sustainable urban mobility [30, 109] have shown that such personalized units of playable content have a significant positive effect on players' engagement as well as retention over time. The fully automatic PCGR approach presented in Chapter 3.2 allows to propose to players mobility challenges that are tailored to their habits and preferences, with minimal additional design work.

We have evaluated our challenge generation and recommendation system through an A/B test conducted within a long–running sustainable mobility game involving more than 400 active players. The main objective of the test was to compare the approach for challenge generation presented in this paper (RS) with an earlier semi–automatic approach based on expert judgment (non-RS). We evaluated three key aspects:

- Comparison of the acceptance rate of challenges generated through the two approaches;
- Comparison of the improvement in mobility habits of the players obtained through RS vs. non-RS challenges; and
- Comparison of the cost, in terms of given in-game rewards, of the RS vs. the non-RS approach.

For all three aspects, the evaluation results show that the RS-based automated approach, thanks to challenges that are tailored to each player's profile, is not only comparable to the one based on expert judgment, but may even be more effective.

Our experiments, however, highlight some limitations and open several additional research questions, both in terms of the proposed approach and in terms of its evaluation.

Taking into account challenge acceptance (the first research question highlighted in this Chapter "Ob1"), measured in terms of completion rate, our experiments show that there is no significant difference between RS and non-RS challenges when samples are drawn form equivalent players' populations. However, the relatively low number of challenges (and players) in the control and intervention groups (164 RS challenges, 280 non-RS challenges) might be a potential threat to validity, and these encouraging results require confirmation in larger trials.

Regarding to the second research question "Ob2", the achievements in terms of behavioral improvement are very promising: conducted experiments show a statistically significant positive difference in the improvement induced by RS challenges with respect to non-RS challenges. However, both RS and non-RS challenge recommendation approaches fail to affect the behavior of a considerable segment of the players' population, represented by players that did not reach previous week baseline or, in some extreme cases, did not take up the proposed challenge at all. This aspect, although possibly influenced by other confounds, questions the effectiveness of the supported challenge mechanism itself (independently from the RS or non-RS system used to generate and recommend challenges) and needs to be further investigated.

A clear limitation of the proposed PCG framework, that might affect the impact in terms of engagement and behavioral change on broader demographics of players, concerns the fact that it considers the elevation of player's status as the key player objective. This objective guides the sorting and filtering algorithm, that promotes challenges whose reward maximizes the chances to elevate the player's status in the game (e.g., reaching a certain level in the game, rise in rankings, win a badge). Moreover, rewards themselves are currently limited to point bonuses.

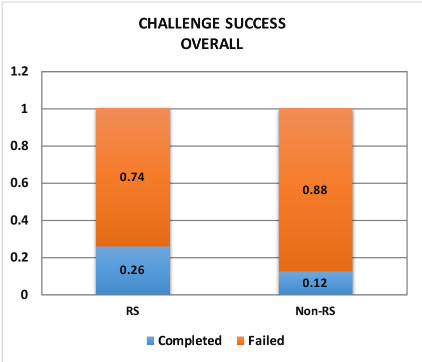
The other noticeable point that should be highlighted is the considerable gap between the knowledge of the framework w.r.t the real players' preferences/abilities in order to handle the given challenges. This room reflected to the acceptance rate of the given challenges (discussed in Ob1) and the negative contributions by the players (who were active in the first two week of the A/B test) in the last weeks of the game. This could have happened due to the lack of learning the player's feedback (not only considering the previous game week performance, but taking into account the whole player's performance that had been recorded during the game to capture a pattern that a particular player tends to tackle the multiple obstacles in the game) by facing the different recommended challenges, or well-motivating the players in the last week. This brings out a need for an investigation to further improve our framework, by injecting the ML technique to learn and model the actual players' feedback with respect to the introduced challenges. In short, integrating ML to the main framework can augment our Filtering & Sorting module, in order to optimize and tune the selection of challenges to the individual player based on her track record with challenges proposed in the past.

Another interesting extension of the proposed work could exploit player modeling and play-style analysis techniques to guide the generation and recommendation of challenges, better tailoring administered challenges, and corresponding rewards, to the player's motivation and objectives. In addition, follow-up work could consider various challenge-based game mechanics: single-player challenges (as the one used in this game), player-to-player challenges, as well as team-level challenges. This would allow to compare different motivational affordances, from purely competitive ones to more collaborative ones, and to evaluate their effectiveness on different player types.

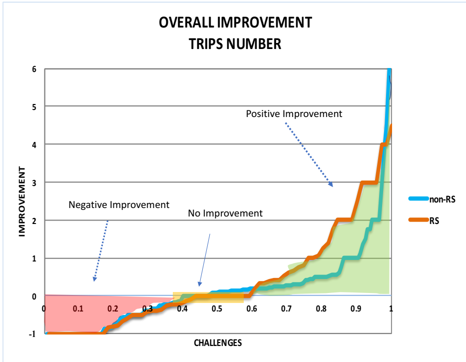
Concerning the third research question "Ob3" that was inquired relating to the cost of RS and non-RS challenge generation approaches, we observed that RS-based challenges are far less expensive (i.e., they induce the same improvement in mobility behavior for less reward) compared to the non-RS challenges. However, further investigation is needed to fully capture the actual relation between the reward associated to a challenge and the induced impact on player's behavioral outcomes.

In the near future, we plan to support other challenge-based mechanics and cover a wider range of reward types that, in combination with play-style analysis techniques, can be exploited to further customize the game content by taking into account players' types. Finally, we plan to apply our solution to other large-scale gamification experiments in the mobility domain and beyond.

In the next Section, we describe the study setup, implementation, and evaluation of the Machine Learning module, which the module is designed and integrated, into APCGR framework –particularly in Filtering & Sorting module– to optimize the challenge selection performance.



(a) Success Rate



(b) Trips Improvement

Figure 4.16: The Overall Challenge Success Rates and Trips Improvement

4.5 Machine Learning Experiment Setup and Evaluation Objectives

Although the achievements presented in previous Sections suggest that the proposed framework can positively affect on improving players' behavior towards the specified goals under the umbrella of the gamified system (which are shown in Figure 4.16a and 4.16b as an example), there is still room for improvement.

As it shown in Figure 4.16a, the overall challenge success rate of the challenges introduced by our proposed framework and the semi-automatic was 24% vs. 12%, respectively. This 24% success rate represents that the proposed PCG framework requires a better understanding of players' ability at handling the given challenges, which have been recommended during the game weeks. It worth-nothing that the success rate does not grantee the behavioral improvement all the time, but it positively reflects to the players' engagement and their demeanor progress.

To further improve our framework, we implemented the ML module, and evaluated it on the data-set that we collected from the Trento Play & Go. The data-set is statistics on the challenges that were given to players' during the twelve weeks game. It contains more than 6 thousand instances that each sample indicates one challenge that demanded the player to complete it within a certain constraint. Therefore, in this experiment we only focus on the urban mobility challenges, which asked players to change, sustain and improve their mobility habits in the particular mode. We structured our evaluation targeting two main objectives which are listed in the following:

- **Objective 4:** *How effective are the ML techniques in terms of predicting the challenge(s) that suit players' preferences?*

Objective 4 (Ob4) talks about the *Effectiveness* of the ML module in our APCGR framework. Specifically, we intend to compare different classification algorithms to determine the best algorithm that provides the better result in predicting the players' feedback's with respect to the given challenges.

- **Objective 5:** *How much time is needed to complete the whole ML process?*

Objective 5 (Ob5), points to the time duration that ML module requires to capture the challenges from the *Valuator* module and return them with one extra feature to the *Filtering* and *Sorting* module.

Over the next Section we have answered the above objectives by evaluating and demonstrating the *Effectiveness* and *Efficiency* of the framework within the Trento Play&Go experiment.

4.6 Machine Learning Evaluation and Results

For evaluating the proposed module, among the 12 weeks data, we have selected 132 players who were active in weeks 11 or 12. Then we look at the history of these players and restrict them to the ones who succeeded at least one challenge from week 2-10. Notably, we got rid of the first week data, as we did not provide any challenge in that week of the game. We used the above setting throughout the evaluation presented in this Chapter.

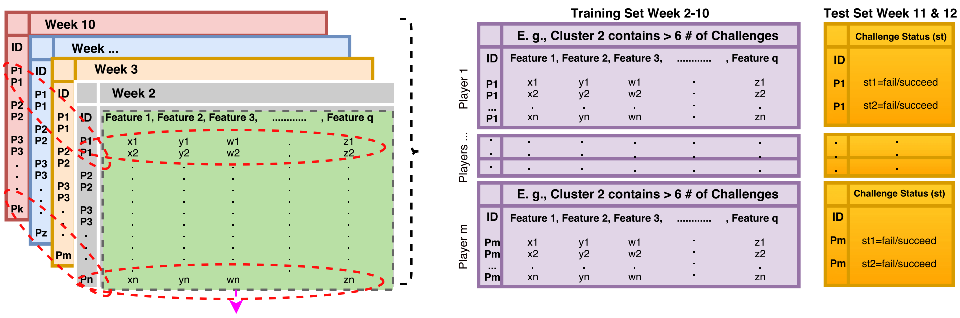


Figure 4.17: Conceptual View the Training and Test Set in Each Cluster

4.6.1 Objective 4 Evaluation

Effectiveness: In order to get objective 4 (Ob4) that is to observe what effect the different numbers of challenges and the ML techniques would have on the effectiveness of the ML module, we choose to run the experiment for varying numbers. In other words, we plan to know how many challenges (and thus, how many game weeks) are needed to predict challenge completion in an accurate way. To this end, we clustered the participants into 5 groups in a supervised way such that each group contains a particular number of challenges as follows:

Table 4.6: The Ratio of Players Play Style in All Sections

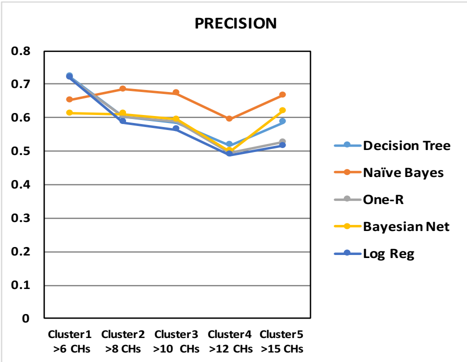
Clust / Dscrn	Clust 1≥6	Clust 2≥8	Clust 3≥10	Clust 4≥12	Clust 5≥15
#Players	89	74	67	42	15
#Challenges	1045	945	879	589	227

Clust: Cluster.
Dscrn: Description.

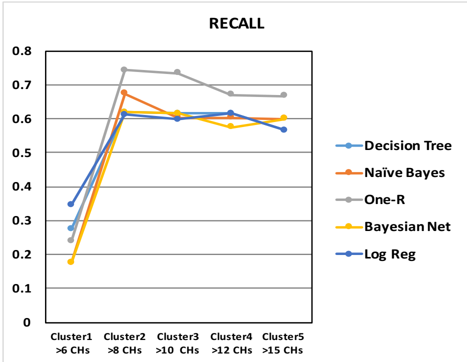
- Cluster 1: Players who have received 6 or more challenges;
- Cluster 2: Players who have received 8 or more challenges;
- Cluster 3: Players who have received 10 or more challenges;
- Cluster 4: Players who have received 12 or more challenges;
- Cluster 5: Players who have received 15 or more challenges.

It is noticeable that these clusters of players hold overlap data. It means, for example, cluster 2 that contains players who received more than 8 challenges, it also covers the players who are given more than 6 challenges. Consequently, the same overlapping data can be seen in other clusters. Having the above setting at constructing the clusters, the first cluster includes 89 players with 1045 challenges. 74 players with 945 numbers of challenges are located in the second cluster. The following cluster consists 67 participants who are given 879 challenges during the game. Cluster 4 with 25 less numbers of players vs. the third one contains 42 players with around 589 challenges. Finally, the last cluster with minimum number of players among the others includes 15 players with 227 challenges (see Table 4.6).

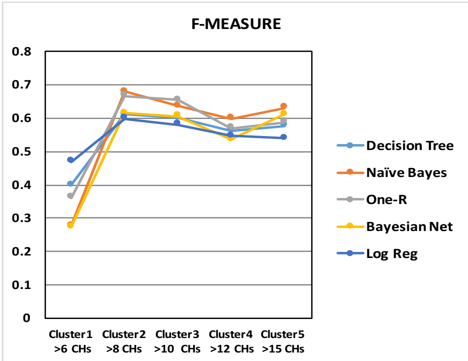
Considering the above settings, we took the individual player's instances that were recorded from week 2 to week 10 to construct the individual player's training set in each cluster (illustrated in Figure 4.17). Then, we performed five different ML classification techniques such as "Decision Tree, Naive Bayes, One-R, Bayesian Network and Logistic Regression (Log Reg)" to train the model. Once the model is trained, the last two weeks of the game data as the Test-set is used to validate the classification methods. In other words, every challenge in week 11 and 12 will be fed to the system to calculate the probability that this challenge will be succeeded or failed by the player. Thereafter, a class with higher PS value will be assigned as the prediction result for our binary classification problem "Succeed or Failed". Finally, this assignment will be compared to find an agreement with the actual result that the player has done within week 11 and 12. Then, the comprehensive performance evaluation of all classification algorithms in every cluster are calculated as follows; the sum of True Positive predictions (TP) of all players instances, the sum of False Positive (FP) predictions of all players, the sum of True Negative (TN)



(a) Precision



(b) Recall



(c) F-measure

Figure 4.18: Classification Performance in Different Clusters

predictions, and lastly the sum of False Negative (FN) predictions of all players, which are depicted in Figure 4.18. Thereafter, the following well-known measurements are used to accredit the experiment:

- **Precision**: Points the ratio of the number of relevant challenges retrieved and of the number of irrelevant and relevant challenges retrieved ($true\_positives/(true\_positives + false\_positives)$), which indicates how useful the results are;

- **Recall**: Indicates the ratio of relevant challenges retrieved and of the total numbers of relevant challenges in our data-set ($true\_positives/(true\_positives + false\_negatives)$), that represents how complete the results are;

- **F-measure**: Refers to the test accuracy, considering both precision and recall (weighted average) to compute the score.

Note: in this experiment, TP is the number of the failed challenges that are correctly predicted as such. TN indicates the number of the succeeded challenges, which are correctly predicted as such. FN refers to the number of the failed challenges that are predicted as succeeded, and FP points to the succeeded challenges that are predicted as the failed challenges.

As it shown in Figure 4.18, the results reported for cluster 1 indicate that this cluster does not provide enough data at training and modeling the players' feedback for prediction, since all algorithms provided a poor performance in the range of 25-50% in F-measure and Recall in this cluster.

In the second cluster almost all the algorithms recorded admissible results, where it can be seen a remarkable betterment from cluster 1 to cluster 2 in Recall and F-measure in all techniques (see Figure 4.18b and 4.18c). Since, the algorithms are applied on each individual player's data, and the overall results of all players are reported as the classification performance result, this improvement happened when we excluded 15 players who have recorded less than 8 challenges (totally 100 challenges) in their challenge pool.

We have used Zero Rule[5] classifier to obtain the accuracy base-line of each cluster, so that we are able to assess the performance of the other classification methods in our classification problem. Similar range of baselines (close to random 50%) have calculated such as 57.7%, 56.7%, 56.4% , 54.4% and 54.6% for clusters 1 to 5, respectively.

Having the above agreement accuracy for each cluster as the base-line (that highlights marginally more than a random classification and prediction), *Naive Bayes* performs better among the other implemented classification methods by 60% in accuracy –slightly better than baseline– and also ∼70%, 68% and 68% in precision, recall and F-measure respectively in the second cluster.

In the third cluster that includes 67 participants who have minimum 10 challenges, the performance smoothly decreased in all techniques, and it intensified in the forth cluster as well. This might have happened due to the fact that these clusters include players' data who are involved in the game for a longer time, so their behavior have been affected by the gamified system. Therefore, considering the whole players' history in this dynamic context, has a negative effect in modeling and consequently predicting their behavior for the next game week.

Thus, extracting and specifying a certain time-window (from the whole history) to train the model could help to enhance the performance of modeling and predicting. In addition, each player had a limited number of challenges (in different clusters) in her challenge's history that can be considered as the training set, hence every single challenge can influence the performance of the classification results.

According to the results that we have obtained in each cluster, and to have an in-depth look at the effectiveness of each individual technique, in this stage we focus on the cluster of players who have received more than 8 numbers of challenges.

[5]We exploited ZeroR classifier which is available in Weka to calculate the base-line for each cluster: `https://machinelearningmastery.com/estimate-baseline-performance-machine-learning-models-weka/`.

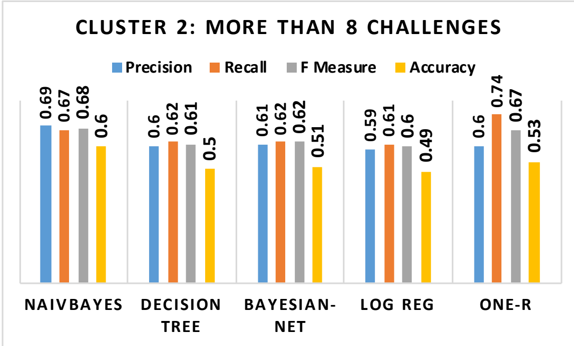


Figure 4.19: Classification Performance in Cluster 2

Table 4.7: Area Under the Curve Measurement in Cluster 2

MSRT \ ALG	Naive Bayes	Decision Tree	Bayesian-Net	One-R	Log Reg
AUC	0.604	0.462	0.469	0.483	0.426

ALG: Algorithm.
MSRT: Measurement.

Hence, taking into account only Cluster 2, comparison between the performance of classification techniques are shown in Figure 4.19, where recall, precision and F-measure are assessed in this comparison. In addition, we quantify the difference between the methods by the *Area Under the Curve (AUC)* [146] metric that represents a probability that the classifier will randomly chosen positive observation higher than randomly chosen negative observation.

In this task, the proportion of the failed challenges that are correctly predicted as such refers to the true positive rate (TRP) and plotted on the y-axis (as known recall). The false positive rate (FPR) refers to the proportion of the succeeded challenges that are predicted as the failed challenges and plotted on the x-axis. However, we use Table 4.7 to report the AUC result of the different classification algorithms, rather than plotting the graph.

As it shown in the bar-chart the performance of Decision Tree, Bayesian Network and Logistic Regression (Log Reg) are almost similar by maximum 60%, in precision, recall and F-measure. We detailed in Table 4.7 the obtained *AUC* in Cluster 2, that represents very poor figures for all classifiers "except Naive Bayes". 0.462 for Decision Tree, 0.469 for Bayesian Network, for Logistic Regression we gained 0.426, and finally for One-R we obtained 0.483.

Although, One-R shows a remarkable recall with ∼74%, it has the similar precision w.r.t the other algorithms. While Naive Bayes provides a better result in predicting the right challenges by ∼70% in recall, precision and F-measure with better *AUC=0.604*.

The above achievement supports the positive effect of integrating the Machine Learning module into the APCGR framework, in improving the challenge selection process in Filtering & Sorting module. We can thus answer Ob4 by:

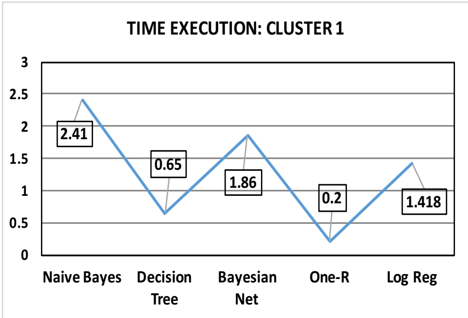
> Exploiting machine learning techniques –particularly *Naive Bayes*– will enable the proposed APCGR framework to optimize the success rate of the recommended challenges in the game, which have a potential ability to support players' motivation to improve their healthy behavioral in the gamified system.

4.6.2 Objective 5 Evaluation

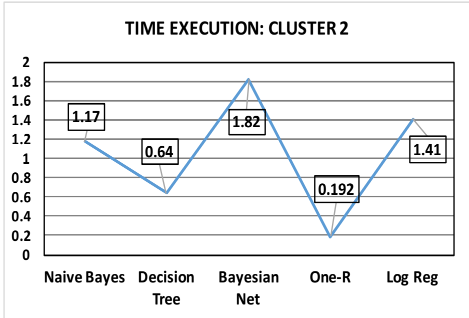
Efficiency: The fifth objective that we aim to reach concerns the execution time at performing the ML module in the framework introduced in Chapter 3.2. This is significantly important when the whole framework needs to be run and executed on-the-fly. Thus, to evaluate the experiment we designed a component as a *Counter* in the module to evaluate how fast the different ML techniques are, when they receive the list of challenges from the *Valuator* module till returning the same challenges with appending the *PS* value on each challenge. To this end, we articulate that the proposed module as the preliminary experiment is conducted in a machine with the following configuration: *2.7 GHz Intel Core i5 machine with 8 GB of 1867 MHz DDR3 RAM*, and the value that we obtained are expressed in *Millisecond (ms)* throughout the Chapter. It is noticeable that this module, similar to the PCG framework and the whole gamification platform will be run in the server that is specified for our Gamification project (Trento Play&Go) and located in the Smart Community Lab at Fondazione Bruno Kessler (FBK) in Trento (Italy). Therefore, the execution time in the server will be much less than these results that we obtained from this single machine.

In Figure 4.20, we report separately the time performances of the ML techniques: Naive Bayes, Decision Tree, Bayesian Net, One-R and Logistic Regression (Log Reg), on all clusters. It is worth notice that the execution time, by tracing all the clusters from Cluster 5 (see Figure 4.20e) to Cluster 1 (see Figure 4.20a), gets longer in overall from 0.356 ms to 1.307 ms, respectively. It is evident that this increasing of the execution time is due to the fact that the different numbers of players located in the clusters, which increase from 15 players with 227 challenges in Cluster 5 to 95 players given 1046 challenges in Cluster 1 (see Table 4.6).

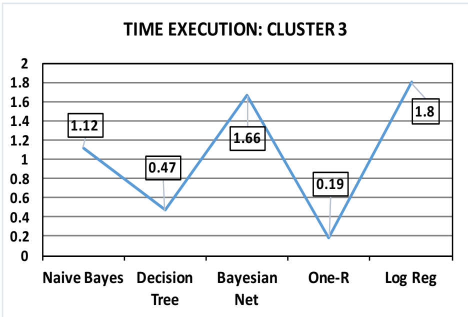
We have shown in Section 4.6.1 that the ML module can be activated (and performs better within the other clusters), when the player recorded at least 8 challenges in her profiles/history. Hence, to have an in-depth look at performance of the algorithm in terms of how fast they are, we focus on Cluster 2 containing 74 players having minimum 8 challenges.



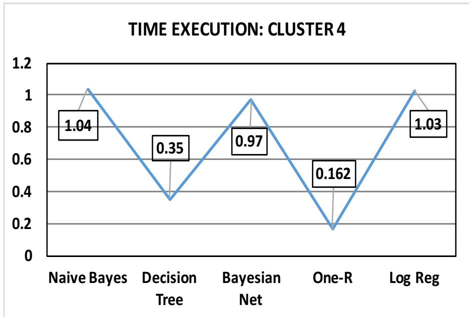
(a) Cluster 1.



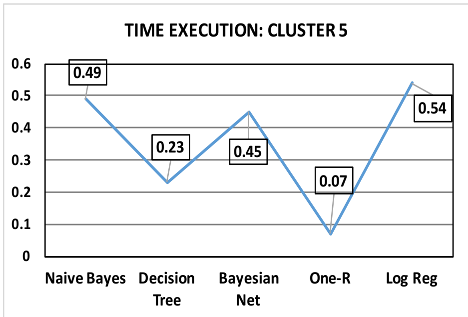
(b) Cluster 2.



(c) Cluster 3.



(d) Cluster 4.



(e) Cluster 5.

Figure 4.20: Time Execution in All Clusters

As it is clear in Figure 4.20b, Logistic Regression (Log Reg) and Bayesian Network need more time to complete their processes with 1.82 and 1.41 milliseconds, while One-R and Decision Tree are the best techniques in terms of time in completing the prediction process with 0.64 and 0.19 milliseconds. Naive Bayes that reports the most effective classification method –with our data– among the other techniques, needs 1.17 millisecond to finish its procedures. Although, time consumption is a crucial element in any framework that designers always try to assess and minimize it, in this generation of game content (challenges) will be important when the generation is on-the-fly.

In the light of the experiment and concerning the trade-off between *Effectiveness* and *Efficiency*, we are able to response to Ob5 by stating that:

> Naive Bayes algorithm seems the best classification method that fits the framework to model and predict the outcome (Succeeded or Failed) of the generated challenges. This prediction can be exploited to further improve the challenge selection process in the *Filtering & Sorting* module.

The above sections illustrated the evaluation and assessment of the proposed APCGR and the integrated ML module, which were implemented, in an Urban Mobility gamification system. The assessment was constructed based on two important concepts; Effectiveness and Efficiency. These two concepts have been reflected into five objectives, that we have addressed throughout the various sub-sections.

4.6.3 Lessons Learned

Section 3.3 presented the work of integrating Gamification and Machine Learning for customizing players experience in the gamified system that serve a starting spot for further exploration in personalization in this context. We have seen in [30] that game content, in particular *Challenges* have a positive effect to foster and motivate players to change their habits toward the specified goals. This behavioral change can be shaped better with exploiting the advantage of machine learning in generating and personalizing the game content, specially for who were not well-motivated to at least try one of the given challenges during the game. This allows us to gamify citizens/players without whipping them by avoiding the generation of non-interesting challenges, but instead, we could learn what the player(s) does not like and make them as the ground truth to generate new game content. Having in mind the above advantages of using ML, we have integrated the ML module in the framework and evaluated it considering the following objectives:

- How effective the ML techniques are to predict the challenge(s) that suit the players' preferences?
- How much time is needed the module completes the whole ML process?

Taking into account these two evaluation objectives, we have found that using ML in our proposed APCGR framework in gamification can be effective for optimizing the challenge selection process. We have obtained almost 70% correctly prediction that makes the ability for the proposed system to assign game content by having more influence on players' engagement in the game.

We have examined the ML module on the data that we collected during the game such that we considered players' experiences from week 2 to 10 to train the model (*Training Set*), and the rest of the data (week 11 and 12) are considered for Test-set to evaluate the performance of the module.

The performance of the classification models are assessed with the actual activities that the players have done during the game. The results that we have obtained and presented in Section 4.5 show that the ML module can be used to improve the challenge selection process in *Filtering* & *Sorting* module.

So far in Chapter 4 we have reported the performance of a system framework for procedural generation of playable content in gamification. We have evaluated the framework in a gamified urban mobility campaign, Trento Play&Go, that ran for 12 weeks in 2016. In addition, a machine learning module is built to optimize the process of challenge selection, when enough data is collected from players during the game period. In the next section, we discuss the achievement from the results of applying play style extraction module on educational gamification context, limitation of the proposed approach and the possible extension to solve the highlighted limitations.

In the next Section, we present the evaluation of the proposed Play Style Extraction module, which was implemented in an educational gamification context.

4.7 Solving the Incognitum Scenario

Solving the Incognitum is a first-person, point-and-click 3D interactive e-learning environment that was developed to teach the relationship between fossil and geological time zone. The game environment is built based on the largest museum of natural history in the early 19th century known as *Charles W. Peale's Museum of Art*[6], which is located in Philadelphia city (USA). During the game, players can interact with museum's exhibits such as minerals, strata deposits, fossils and portraits of renowned historical figures related to the exhibits (a screen-shot of the game and its environment is shown in Figure 3.9). In addition, players are able to collect the jewelry of the museum by correctly answering the questions they face in various sections of the game. Questions are mainly associated to fossils, minerals or other information related to geological time, location, etc.

Game content and the whole environment have been constructed to backup the various learning and play styles based on Achievement Goal Theory (AGT) [147]. Comprehensively, the aim of this interactive educational game is to demonstrate the following three main concepts [148]:

- Law of Superposition[7],
- Mastodon Life[8], and

[6] https://www.philamuseum.org
[7] https://www.britannica.com/science/law-of-superposition
[8] https://en.wikipedia.org/wiki/Mastodon

- Geological Time[9]

The game consists of 6 sections such that players after passing a short section as "*Tutorial*" are able to know what the game is and how to play it. They then can explore the virtual space in the main four sections of the game called "*Quests*" to assess various exhibits, as well as answering multiple questions related to them in order to reach and complete the objective of the game. The game ends with an *Exploration* section, where players are allowed to continue the exploration of the game.

In this study there were 75 freshmen students (in Digital Media department) who volunteered to participate to the game to study the correlation between play style and player type and their influence on learning in *Solving the Incognitum* game. In this experiment, volunteers were not told that they must complete the game, hence they could have quit the game whenever they prefer or get bored of playing the game.

Before starting the game, participants were asked to complete a pre-knowledge test consisting of 12 questions about the essential earth science context. They were also given up to 1 hour with an instruction to play the game.

The procedure used to collect the essential data from the game is monitoring the screen of the game and recording all the game play data while players played the game. After finishing the game, participants were again demanded to fill out two types of questionnaires: first a post-knowledge acquisition questionnaire consisting of 15 questions (AGT survey [149]), and the second a survey contains an 11-question about player type [147]. Thereafter, to validate and guarantee the reliability of the survey questions Cronbach's alpha [150] was used (for more information about questioners see [148]).

4.8 Player Modeling Experiment Setup

We implemented the proposed module on the data collected from *Solving the Incognitum* [107] described in the above Section (4.8). The data-set that contains 75 participants gameplay's data was used in [107], in which a player modeling framework was constructed based on episodic segmentation of gameplay traces and sequential machine learning. Basically, researchers represented how the players change their play styles during the game. In that experiment two researchers by recording (screen recording) the participants' activities tried to extract their play styles. Due to the change of play style of players during the game, they segmented the game into four sections (Quests) and manually labeled the players' play styles by looking their behavior and activity regarding to visited items, navigation in the game, use of the concept of map, questionnaires, etc. Hence, Explorer, Goal-seeker (or Achiever), Lost

[9] `https://www.britannica.com/science/geologic-time`

and Careless types of play styles are assigned to those players. The proportion of the assigned play styles are shown in Table 4.8 that indicate players do not fall into a specific style during the game, and they dynamically change their play styles over the time.

Table 4.8: The Assigned Play Style Proportion During the Game (4 Sections)

Sections / Play style	Section 1	Section 2	Section 3	Section 4
Explorer	50%	25%	29%	13%
UnIntrested (Careless)	25%	18%	20%	16%
Goal Seeker (Achiever)	∼17%	40%	38%	63%
Other (Lost)	∼8%	17%	13%	∼8%

Borrowing the strategies "Time-based windows and Segmentation" that we used in [107], we evaluate our proposed automatic play style cognition on player's gameplay data that was recorded from her behavior in the game. Three players are excluded via *InterquartileRange* and *Removewithvalues* functions explained in Section 3.4.4, because of having extreme and outlier values. In addition, we excluded 17 players' data instances (in order for validating the proposed approach), who are *Not* labeled due to miss-information, monitoring, questionnaire. Basically, the experts could not labeled those 17 players because of the mentioned issues.

Behavior of each individual player is represented by around 70 features in each section of the game which vary widely, from how the player moves in the game to visiting location and items, the number of that she answered (including correct and wrong), to the time that she spent to read and answer the questions. This information is captured individually while she plays the game.

4.9 Player Modeling Evaluation Objectives

Having in mind the different sections of the game (detailed in Section 4.8), we took into account only 4 sections (Quests) of the game and 2 minutes time-windows to implement the proposed approach. To assess the performance of the proposed module, we have looked at two complementary aspects, which are reflected in the following objectives:

- **Objective 6:** *To what extent the play styles that are automatically extracted for a player are similar to the labels selected and administered by means of research members for the same player in the same section* *of the game? and with how much confidence players played toward the recognized play styles?*

- **Objective 7:** *To what extent can we predict the play style considering the previous styles?*

We applied the proposed framework on the four main Quests (sections) and considered two minutes time-windows. The list of features which are important to characterize the play styles of interest (see Chapter 3.4.1) were selected among the 70 features that are logged in the data-set using the domain knowledge; however, there are other methods that can help to characterize those styles (e.g., survey, questionnaires). Thus, in this experiment we utilize only the extracted features from the data-set without demanding any question from players to model their behavior.

Three main categories of features were extracted such as *Time spent* in the game, number of *visited game content*, and the game content that reflects the *player's feedback* from the game, given a total 11 features for each player. Features were selected in basis of usability for characterizing players' pattern in the game. In other world, the listed features deliver a map that the proposed system exploits to extract the play styles following the defined Rule Set. Given the above rationale, the selected features are depicted in Table 4.9:

Table 4.9: Selected Features

Category	#	**Features**	**Description**
Time Spent	1	Total Time for Reading	Refers to the total time that the player spent to read the content of the game. The average time taken to read the game content during the game was recorded around 2.42 sec.
	2	Navigation Time	Points to the total time that the player spent navigating around the game.
	3	Time for Map	Talks about the total time that she spent reading the Map in the game.
	4	Reading Min	Refers to the minimum time that the player spent to read/visit a single component of the game around 0.670 sec.
	5	Reading Max	Shows the maximum time that the player spent to read/visit a single component of the game. The maximum time record during the game is 3.77 sec.
Visiting game content	6	Total Item Visited	Depicts the total number of items that the player visited.
	7	Item Visited New	Points to the number of new item visited in a section.
	8	Item Revisited	Illustrates the total number of component that she re-visited in the game.
	9	Questions Visited	Talks about the total number of visited questions.
Player's feedback	10	Questions Right Ratio	Refers the percentage of correct answers that the player provided.
	11	Questions Wrong Ratio	Shows the percentage of incorrect answer that the player responded.

Rule Set:

We take into account the features relating to Time and Visiting the game component to interpret how *fast* and *slow* players play in the game, and consider the feedback of the player to how *precise* they play in the game to characterize play styles. The above features are considered as the *Rule Set* (ground truth) to characterize the style as described in Section 3.1.

The propose module is used to construct Table 4.11 that shows the ratio of players' play styles and how their play styles changed over time in various sections of the game, according to the collected data.

As shown in previous work [107], play style is not constant and players change their play styles over the course of the game (see Table 4.8). Dynamicity of play styles were also obtained by implementing the proposed automatic approach on the collected data, however different results in ratio of play styles are evident, which are discussed in the next Section. For example, in section 2, there was a 36% of Explorers, whereas in section 3 there was around 20% of Explorers, and the number of players classified as Careless changed from 16% in section 1 to around 13% in section 2.

4.10 Player Modeling Results

4.10.1 Objective 6 Evaluation

In order to reach the Objective 6 (Ob6), we have implemented two comparison tests. First, which is "*Peer to Peer*" comparison, we conduct an A/B test between the results that we obtained in [107] (for simplicity, we call Group A from now on) against the results that we obtained using the proposed module in this dissertation (for simplicity, we call Group B from now on). Secondly, we conduct the same A/B test between Group A against Group B by eliminating the players who played in Margin. In this experiment, the players whose confidence values are less than 10% are considered and highlighted as Margin players (players whose behavior cannot be characterized by a specific play style).

Table 4.10: Proportion Comparison

	Section 1		**Section 2**		**Section 3**		**Section 4**	
	A	**B**	**A**	**B**	**A**	**B**	**A**	**B**
Explorer	50%	35%	25%	36%	29%	20%	13%	13%
Careless	25%	16%	18%	13%	20%	29%	16%	20%
Achiever	17%	27%	40%	27%	38%	38%	63%	51%
Lost	~8%	22%	17%	24%	13%	13%	~8%	16%
SqDif	Δ =0.377		Δ = 0.036		Δ = 0.0162		Δ = 0.425	

SqDif: Square Difference.

Table 4.11: The Ratio of Players Play Style With and Without Confidence Value in All Sections

		Absolute Comparison			With Confidence value: >10%		
	Play Styles	# of Correctly Labeled (per type and total)		# of Players	# of Correctly Labeled		# of Players
Section 1	Explorer	18	33	55	18	33	48
	Achiever	7			7		
	Careless	7			7		
	Other (lost)	1			1		
Section 2	Explorer	9	29	55	5	21	40
	Achiever	8			13		
	Careless	6			1		
	Other (lost)	6			2		
Section 3	Explorer	10	36	55	10	35	48
	Achiever	16			15		
	Careless	8			8		
	Other (lost)	2			2		
Section 4	Explorer	4	32	55	1	22	27
	Achiever	23			20		
	Careless	4			1		
	Other (lost)	1			0		
Total # of Players		130		220	111		161
Overall Accuracy		60%			~70%		
Performance Measurement		Recall=0.53 Precision=0.54 F-measure=0.53			Recall=0.60 Precision=0.61 F-measure=0.60		

For the first experiment, we have conducted an A/B test between the labels that the proposed module assigned to players (Table 4.11) vs. the ones obtained and reported in [107] that are shown in Table 4.8. Then, we compared *peer-to-peer* play style of the players that are assigned in each section of the game. In other words, the play style of every single player in Group A is compared to the same player's play style in Group B. We refer to this test as *Absolute Comparison*. As it is detailed in Table 4.11, overall 60% of the labels are identical between Group A and Group B, that demonstrates a moderate labelization accuracy in Group B using the proposed framework. The resulting low accuracy is not surprising: surveys results, administered to players after the game session, were exploited in the manual labeling of play styles as additional information and turned out to be essential for the labeling process in all cases of ambiguous play styles

In the second comparison experiment, we took into account the confidence values calculated by Equation 3.4 for each player in all sections of the game. Hence, we excluded the players who have less than X% confidence value from Group B and mutually from Group A. Then, we carried out an A/B test on the two groups. A part from the comparison test that provides a value (a percentage of agreement or accuracy) that could be used to validate the proposed module compared to manually extraction of play styles, these confidence values reveal that with how much certainty a player behaves toward a specific style.

Having the above statement, thus the system is able to highlight players who played in margin by adjusting a threshold value to meet (e.g., Conf=10%). Although, the numbers in absolute comparison shows a poor similarity between the results, by excluding those who played in margin from Group A and B, the accuracy of correctly labelization increased up to 70%. Table 4.11

indicates the percentages of play styles which are identified with confidence and without confidence value in Group B.

We can also look at the significance of the correctly labeled play styles between the groups (with confidence value, for simplicity we call Group C as for now, and Group B that defined above) vs. the ground truth (Group A that manually extracted and labeled to each player). To that end, we applied the *Chi-Square*[10] on these groups. Once between Group B vs. Group A and once between Group C and Group A.

Looking to the classification measurement such as Recall, Precision and F-measure in this task, it is evident that there is a considerable difference between the implementation of the proposed approach on the two population "Group B and C". The result of F-measure, which is the weight average of recall and precision, shows 8% improvement from the first population (B) to the second (A). The similar betterment obtained in recall and precision from Group B to A. We took into account these improvement (around 10%) in all metrics a good result, since we have multiple classes. This shows that when outlier samples are drawn from the populations, the improvement in correctly play style labelization is significant.

The plot in Figure 4.21. presents the distribution of the extracted play styles with their confidence values in the four sections of the game. y axis indicates the confidence values in ranges $[-\infty, +50]$ in which the negative values refer to the fact that recognized styles are far from the characteristics of the styles compared to the *Rule Set*. While, the more positive value shows players are close to the style that are labeled. Thus, we reached to this statement that the more close the confidence value to zero, the more that player plays ambiguously.

Looking at the confidence values of all possible play styles that every individual player recorded in each session of the game, it is undoubtedly evident that players relatively trended toward behaving in a particular style in each game section. For example, player "Bob" behaved ∼30% as Careless, and ∼20% as Explorer in section one, and played 20% as Achiever, and 35% as Explorer in the next section of the game. As it is mentioned in Section 3.4.2, the module picked up and assigned the play style that has higher confidence value. For instance, considering the Bob behavior in section one, the module assigned Careless play style with ∼30% against Explorer by ∼20% confidence value.

This relatively behaving in the game brings up a critical challenge in adapting game content in basis of the play style.

The above proportion therefore provide no support for a statistical difference for assigned play style of the players groups of this magnitude. We can thus answer Ob6 by:

[10] We took the advantage of *chisq.test()* function from *FunChisq* package in r for the statistical test.

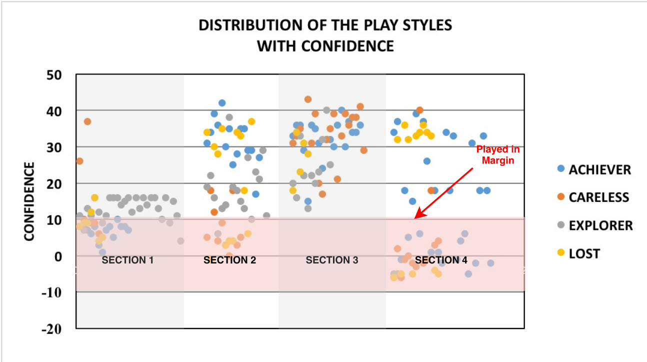


Figure 4.21: Confidence Value of the Extracted Play Styles.

There is no significant difference in proposed automatic extraction play styles compared to the ground truth that administered by expert researchers, when margin players are excluded.

4.10.2 Objective 7 Evaluation

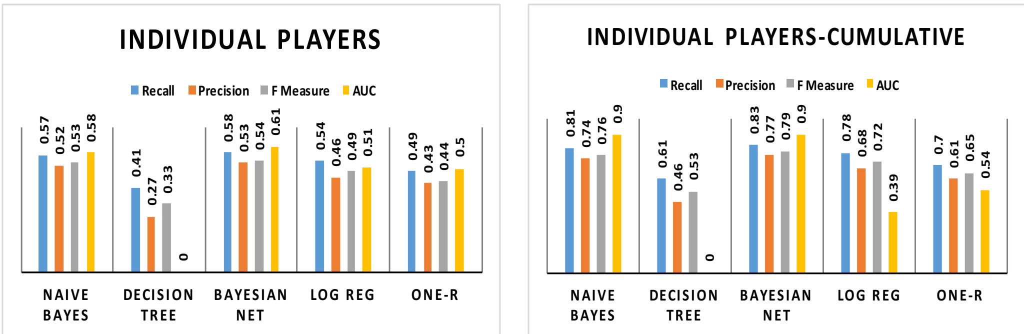
To evaluate objective 7 (Ob7), we take into account only the styles which we used in absolute A/B test comparison (*one single style*) without considering their confidence values. Thus, we have constructed three different experiments as follows:

- *Section by Section (Individually)*; in this experiment, we aim to predict the play style of individual player in each section of the game "Separately".
- *Cumulative*; in this evaluation, to classify play style for a given player, we consider all the player's data from all sections prior to the one at hand. And finally,
- *Time Window*; to conduct this experiment we ignore sections, and just consider the data collected from the past 2 minutes whenever we want to classify play style.

For each experiment, we constructed different Training and Test sets which are detailed in its section, and in all experiments we try to predict the play style that already labeled in previous sections or time-windows.

Section by Section (Individually):

To evaluate this experiment, we took into account the data of each player separately (player's performances only in that section) and evaluated modeling and prediction pipelines. Due to the fact that the number of samples for each player is quite limited, we also took into account the data of the first section of the game (Tutorial). Hence, to construct the Training and Test set with maximum 5 instances for every player, the iterative leave-one-out cross validation is used to evaluate the classification problem. Each round, one sample is left out to be considered as the *Test set* and the remaining samples are used to learn the model. Finally, the average of all players' classification performances is calculated and reported as the result of the classification approach. In other words, the comprehensive performance evaluation of each classification methods in all players are integrated as follows; the sum of True Positive predictions of all players as TP, the sum of False Positive predictions of all players as FP, the sum of True Negative predictions of all players as TN, and lastly the sum of False Negative predictions of all players as FN.



(a) Individual Experiment (b) Cumulative Individual Experiment

Figure 4.22: Classification Results for Individual and Cumulative Individual Player

To train the model, we employed 5 different ML classification techniques such as "Naive Bayes, Decision Tree, Bayesian Network, Logistic Regression (LOG REG) and One-R" and compared vs. each other to find out which method can provide the best result w.r.t our data. Once the model is trained, the Test-set is used to validate the classification method. Thereafter, well-known metrics such as Precision, Recall and F-measure are used to accredit the experiment.

The classification result of 5 approaches are depicted in Figure 4.22a, where *Bayesian Network* with 58% recall, 53% precision and F-measure 54% and Naive Bayes by 57% recall, 52% precision and 53% F-measure (almost similar efficiency) accomplished better among the other approaches, which considered an admissible result by comparing with $\sim 36\%$ accuracy as the base-line. We have calculated the baseline using Zero Rule[11] classifier available in Weka [151].

In addition, we quantify the difference between the methods by the *Area Under the Curve (AUC)* [146] metric which represents a probability that the

[11]We exploited ZeroR classifier which is available in Weka to calculate the base-line for each cluster: `https://machinelearningmastery.com/estimate-baseline-performance-machine-learning-models-weka/`.

classifier will randomly chosen positive observation higher than randomly chosen negative observation. Hence, looking at AUC value of Bayesian Network and Naive Bayes, Bayesian approach performed marginally better with AUC of 61% vs. Naive Bayes with 58%. Following the above two models, Decision Tree, Regression Logistic and One-R classification approaches also obtained a poor results "almost close to the baseline".

These low classification results on the players' individual recorded data is not surprising, due to the lack of well-enough instances in Training set to properly model the individual behavior. Another reason for this low performance could be the change of play styles sourced from players behavior during the game.

Cumulative Experiment:

To evaluate this experiment, we took into account the same sections used in the Individual experiment. Thus, the cumulative data-set of each personal player contains five samples that logged in her profile/history as follows; *section 0-0; section 0-1; section 0-2; section 0-3* and *section 0-4*, which each section indicates player's behavior up to that section. E.g., *section 0-0* represents the behavior that a player (Alice) has done in section 0, *section 0-1* refers to the activity that Alice behaved in section 0 and section 1, and so on. The conceptual view of this cumulative experiment is depicted in Figure 4.23.

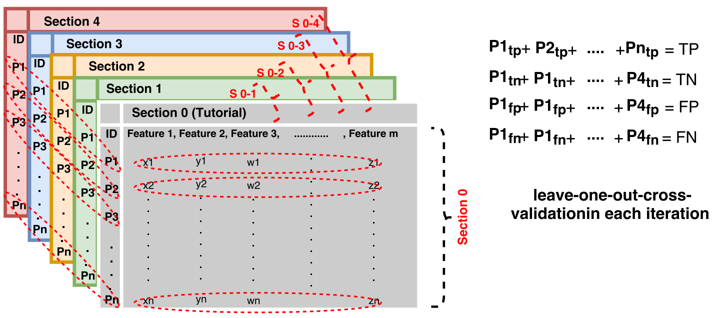


Figure 4.23: The Structure of Training and Test Set in Individual Cumulative Experiment.

We performed leave-one-out cross-validation on players' cumulative data using Training and Test set and implemented the above 5 classification approaches. Similar to the *Individual Experiment*, the sum of True Positive, False Positive, False Negative and True Negative off all cumulative sections are integrated to calculate the comprehensive performance of the classification method. Figure 4.22b represents recall, precision, F-measure and the area under the curve (AUC) values obtained through cross-validation on the Training set using the validation set (denoted as Test set).

Table 4.12: Classification Performances-Individually

Metrics –>	Recall		Precision		F-measure		AUC	
Methods	Ind	CuM	Ind	CuM	Ind	CuM	Ind	CuM
Naive Bayes	57%	81%	52%	74%	53%	76%	58%	90%
Decision Tree	41%	61%	27%	46%	33%	53%	-	-
Bayesian Network	58%	83%	53%	77%	54%	79%	61%	90%
Logistic Regression	55%	78%	46%	68%	50%	72%	51%	39%
One-R	50%	70%	43%	61%	44%	65%	50%	54%

Ind: Individual result.
CuM: Cumulative result.

In this experiment a significant improvement in classification results is observed. For the sake of clarity, one-to-one comparison between the results from Section by Section and Cumulative experiments are depicted in Table 4.12. E.g., Bayesian Network provided a remarkable improvement with recall 83%, precision 77% and ∼80% in F-measure, which compared to the individual experiment around 20% is improved in all metrics. A similar betterment is also achieved by Naive Bayes approach with 81% in recall, 74% precision and F-measure with 76%. Although the numbers in Logistic Regression showed a significant improvement compared to the achievement in Individual experiment, the approach is still poor in AUC value with 39% (a bit more than the baseline).

This significant improvement may be due to that the cumulative experiment could exploit data richer which properly characterize players' play styles. The results presented in the above three experiments highlight that *Bayesian Network* performs better w.r.t our small data-set of samples. Hence, in the next experiment we determine to implement Bayesian Network to model players' behavior with the time frame.

2-minutes Time Windows:

In this analysis we ignored the 4 main sections of the game, and segment the whole game into equal 2-minutes time windows (see Figure 4.24). Thus, to classify play style we just consider the data collected from the past 2 minutes. As it is mentioned in Section 4.8, each player had been allocated for 60 minutes to play the game. But many players quit the game before completing the whole game (for any reason), so we are limited to only 12 2-minutes windows. We then exclude players who have less then 6 time windows (basically players who played minimum 12 minutes in the game). Hence, leave-one-out cross-validation is used to construct Training and Test sets on each player's time windows.

Similar to Section by Section (individually) experiment the outcome of test samples for each player (we refer to TP, FP, TN and FN) are integrated together to compute the overall result. Naive Bayes, Decision Tree, Bayesian

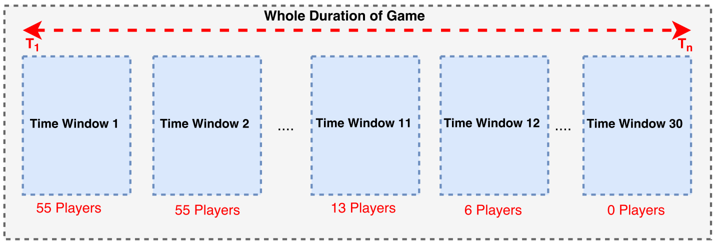


Figure 4.24: Conceptual view of 2 Minutes Windows Segmentation.

Network, Logistic Regression and One-R classification approaches are implemented on each time window. Indeed, in this experiment we intend to represent the performance of the prediction while players change their play styles in different time windows.

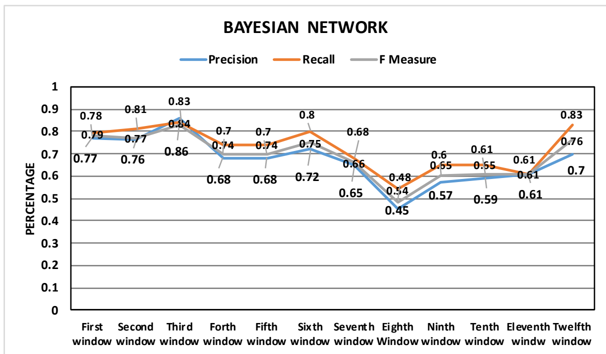


Figure 4.25: Classification Result for 2-minutes Windows.

Results, illustrated in Figure 4.25, show the flow of changes in classification performance through the 12 time windows.

We can observe reliable performances till window 6, with ranges of 78% to 80% for recall, 79% to 75% for precision, 77% to 72% for F measure. While, the classification performance smoothly started to decline from overall 72% (F-measure) in window 6 to overall 61% (f-measure) in window 11, in which window 8 recorded the worst numbers by 54% recall, 45% precision and 48% F-measure. This decrement of performance might be due to the less number of samples (maximum 6 time windows) that those 13 players recorded in their profiles.

Classification validity measures achieved in window 12 (which is the last time window) show an admissible performance with 76% in all metrics. Since, sample size is a significantly important feature in any empirical study (specially modeling in machine learning), this result in that last time window indicates, considering more time windows to learn and model the styles, led to obtain a good prediction.

Taking into account the above three evaluations, we can thus answer Ob7 by:

The proposed prediction approach proved to be able to predict play styles with good performances, particularly when cumulative strategy is exploited. However, lower performances have been obtained whenever dealing with poor or limited data. Moreover, Section by Section and Time Window strategies, present issues when dealing with players behaving relatively towards specific play styles.

4.10.3 Lessons Learned

In Section 3.4.2, we have provided a recognition module that automates the procedure of extracting play style of players in gamification context. The proposed recognition module constructed in basis of a Utility Function that is able to extract play styles by scoring players' behavior affiliated to a specific style in the game. The module enables the main framework to contextualize game content (e.g., generating personalize game content) with respect to players' play styles.

Evaluation results from the data which was captured from an educational interactive game-based learning called *Solving the Incognitum* [107], has presented such player modeling and play style extraction could have a potential positive influence to advance game customization that lead to boost players' engagement in the game. The automatic play style recognition framework presented in Chapter 3.2 (Section 3.4.2) is capable of modeling players' relatively behavior in each stage of the game that allows the system to tailor the game w.r.t the recognized style.

We have evaluated the play style extraction module through an A/B test on the data collected from an interactive educational game-based environment. The main objective of this preliminary work and experiment was to compare the play styles that the automatic framework assigned to players with the play styles that manually administered (for the same player) by expert researchers reported in [107]. Thus, the following evaluation objectives led our study to evaluate the framework:

- To what extent the play styles which are extracted for a player are similar to the play style administered manually for the same player? and with how much con dence players played toward the recognized play styles?
- To what extend can we predict the play style considering the previous styles?

Taking into account Ob6, the evaluation result indicted that there is no significant difference between the two approaches. Around 70% accuracy of agreement is obtained between the labels, which are extracted by the proposed automatic framework against the play styles that labeled to the same players by the expert researchers, when players who played in the margin of classi cation are removed from the whole players population (59 out of 220 samples in whole sections of the game).

The results reported for Ob7 specified that cumulative individual player's data is preferred in considering the model player's behavior by achieving around 80% accuracy in the test set vs. Individual Experiment and Time Windows with overall 60% accuracy in the test set.

The small number of contributors might be sourced such a poor classification performance (Individual and Time Windows experiment). Moreover, the issue could be intensified by noisy data that players recorded during the game. This highlights the need of larger population to learn and model players' play styles in the game. The clear constraint of the proposed play style recognition is that the framework is a Rule Based system that limits the system to extract play styles which are out of the Rule Set.

Having in mind the positive influence of personalized challenges to improve players' engagement in the game reported by Ob2 in Section 4.4 (in this gamification study that was introduced in Chapter 3, players' improvement in a specific mobility means is considered as players' engagement in the game), integrating this recognition module to the main framework can act as a booster to enhance players' engagement.

A possible extension of the proposed framework can improve the Rule Set. Instead of relying on the Rule Set that we manually constructed, unsupervised machine learning algorithms (for instance, Expectation Maximization (EM), K-means algorithms or/and self-organizing feature map (SOFM) [116, 152, 153]) could be implemented on the data to cluster them based on their similarities and recognize the relations between the features to discover a pattern for each individual player's behavior. Another interesting extension of the proposed framework can be implemented and led by the following question:

- *How can prediction contribute to game adaptation?*

Answering the question needs to look at the result of Ob6 and Ob7, in which we have reported how players played the game, and to what extend the system predicted the play style for the next game session. Although, in responding Ob6, we achieved that players relatively played toward a specific play style that make personalization challenging to individual player, we hypothesize that game adaptation module is capable of generating game content (w.r.t the play style) greater than randomly generation (without considering play style).

In spite of the fact, that the result of Ob6 allows the system to use the extracted style to tailor the game (without exploiting machine learning techniques to predict the style) to the individual player, exploiting Ob7 supports the system to consider past players' experience in customizing the game. This personalization can be implemented on different game content on-the-fly.

For example, for Achievers game content (e.g., considering the Questions introduced in the game as Challenges that players should handle and complete them) can be tailored in terms of difficulty, or the numbers of questions. In another context like our Urban Mobility system, could be tailored on the other type of game content such as badge collections, points, etc. While for the Lost players game content can be customized to reduce the complexity of the game element when she plays the game.

As the part of our future work, we plan to implement the proposed module in our Urban Mobility gamification with larger participants to extract and model players' behavior. The integration of play style recognition and machine learning modules will augment the performance of generation of personalized game content (challenges). It contributes to enhancing the performance of recommending the personalized challenges to the individual players that may increase the rate of challenge success, as well as players' improvement in the game.

4.11 Summary

Persuasive technology "Gamification" can be more forceful for changing players' behavior towards the healthy style, if the right strategy is used to persuade players –particularly– in a large and long gamified system. The evaluation of APCGR and its integration were an attempt to assess the validity of the proposed solution (in an urban mobility system) to overcome the problems, which were highlighted in Chapter 1. Concerning the problems, our APCGR evaluation showed that the effectiveness of such technology can be enhanced by dynamically and automatically applying personalization in a large-scale gamification, rather than using the pre-defined game content and one-size-fits-all strategy in generating the content (challenges) during the game. As the assessment of the integrated ML module into APCGR framework showed (Section 4.5), utilizing the machine learning algorithms optimize the the challenge selection process that can contribute to enhance players' motivation in the gamified system.

The positive effect of tailoring the game to players' play style has been proved, in increasing players' motivation in digital games. Thus, considering this potential, with the aim of advancing personalization in gamification, we evaluate the play style recognition module (in an education gamified domain) as the first step of game adaptation based on the players' play style.

Automatically and dynamically deriving the players' play style is an important feature and step that can be exploited by future releases of our framework for tailoring the game content towards the play style. Our experiment and assessment explicitly showed that the players changed their play styles during the game. These play style changes also express the need for dynamic personalization in the curse of the game.

In summary, a one-size-fits-all is not a good strategy to take in order to generate game content in gamification. Rather, our solution illustrated that using PCG and recommender system to generate and personalize the game content –to individual players–, is more effective in changing players' behavior towards the specific goal.

This dynamic personalization was constructed based on players' game status and objective, preference, skill, and the objective of the gamified system, and we postponed the investigation on game adaptation w.r.t the play style for the future work.

In the next Chapter, we conclude this dissertation by wrapping the problems that we highlighted to defeat in this dissertation, solutions that we provided to handle the issues, and contributions that we achieved in this empirical PhD project. Then, the possible extensions of the proposed APCGR system – in particular recognition module– that can be investigated for the future work are described.

Chapter 5

Conclusion and Future Works

5.1 Summary of Research

Recent years have witnessed an increasing amount of investigation in gamification to use this powerful tactic to motivate people towards the more sustainable and healthy behaviors. Nevertheless, gamification performance tends to abate in the medium to long-term. This can be caused from two common issues: the dominance of pre-defined game element that are defined at design time, and a one-size-fits-all strategy in generating game content during the game. To overcome this problem we developed a technological solution that enables the dynamic generation and personalization of game content in a large scale gamification domain.

This dissertation presents the design of an automatic PCG framework that enables the game content generation and personalization with respect to the various aspect of players including preferences, skills, abilities, etc. through a large scale field experiment in the context of Urban Mobility system. The results clearly show that implementation of our proposed automatic PCG framework in a large scale gamification experiment, *is not only feasible*, but even is more *Effective* and *Efficient* against to the manual game content generation to incentivize participants to improve their healthy behavior in the gamified system.

In particular, this dissertation focused on developing a PCG Recommender System framework, in which artificial intelligence techniques are used to tailor the game content to each individual player aimed to keep them engaged in the game in the following steps:

- End-to-end automation of generating personalized game content (challenges), to be assigned to each individual player in gamification;
- Devising and integrating a machine learning module into APCGR framework to optimize the challenge selection procedure (extended version of the framework). And;
- Enabling player modeling concept in gamification to advance personalization (the second extension of the framework).

In Chapter 3, we presented the design and integration of procedural content generation strategies, recommender system, machine learning and player modeling to build a framework with the aim of overcoming the highlighted gap in gamification context. The ML module is designed, as a complementary work, to enable the framework by exploiting machine learning techniques to optimize the challenge selection process in *Filtering* & *Sorting* module aimed to improve personalization in gamification –in general– and –in particular– increase the success rate of the given challenges over the course of the game.

Finally, in Section 3.4, we demonstrated the first and essential step towards the implementation of player modeling concept in gamification context. We developed a score-based module that can automatically extract players' play style during the game. Chapter 4 reported the implementation and evaluation of APCGR and its' integrated modules; Machine Learning and Player Modeling.

We have examined our solution in two different contexts of urban mobility system and game-based educational. Our proposed framework has been evaluated, with two important concerns in mind: Effectiveness and Efficiency, which are reflected in seven objectives defined through the experimental evaluations.

5.2 Summary of Objectives and Finding

In Table 5.1, we directly point to the findings through the different evaluations, which correspond to the defined objectives, as well as the effectiveness and efficiency of the approach.

5.2.1 Effectiveness

This aspect was assessed through five objectives. Ob1 and Ob2 mainly refer to the effectiveness of the proposed APCGR approach in changing the players' behavior against the static approach. The evaluation results showed for all two objectives the RS-based automated method, thanks to challenges that are tailored to each player's profile, is not only comparable to the one based on expert judgment, but is even more effective.

Table 5.1: Overview of Objectives and Findings

Research Question	Aspect	#	Objectives	Finding
How can we construct an automatic framework that effectively, efficiently and dynamically provides personalized game content that can boost players' engagement, as good as manually content generation in a gamified system?	Effectiveness	Ob1	Comparison of the success rate of the generated challenges by both approaches?	There is no significant difference in acceptance between RS challenges and Non-RS challenges.
		Ob2	Comparison of the improvement in mobility habits of the players obtained through RS vs. non-RS challenges?	Challenges that automatically generated and assigned by APCGR led to higher level of improvement than analogous challenges Non-RS.
		Ob4	How effective the ML techniques are in terms of predicting the challenge(s) that suit the players' preferences?	Exploiting machine learning techniques enables the proposed APCGR framework to optimize the success rate of the recommended challenges that can help to improve their behavior.
		Ob6	To what extent the play styles that are automatically extracted for a player are similar to the labels chosen by the experts?	There is no significant difference in proposed automatic extraction play styles compared to the ground truth that administered by expert researchers, when margin players are excluded.
		Ob7	To what extent can we predict the play style considering the previous styles?	The proposed prediction approach proved to be able to predict play styles with good performances, particularly when cumulative strategy is exploited. However, lower performances have been obtained whenever dealing with poor or limited data.
	Efficiency	Ob3	Comparison of the paid rewards for the unit of improvement by RS vs. Non-RS approach	Challenges assigned by our system yield better improvement for the same (even less) amount of per capita reward.
		Ob5	How much time is needed the ML module completes the whole ML process?	Naive Bayes algorithm is the best classification method that fits the framework to model and predict the outcome of the generated challenges.

Ob4 and Ob7 generally point to the effectiveness of Machine Learning techniques to predict the players' feedback and play style during the game. The former "Ob4" talks the effectiveness of the integrated ML module to predict the success rate of the recommended challenges. The results from the evaluation of this module on the data collected from Trento Play&Go showed that exploiting the framework with ML can help to optimize the challenge selection process. We have almost obtained 70% correctly prediction that makes the ability for the proposed system to assign game content by taking into account various features (e.g., the level of difficulty that enables the concept of flow in the gamified system, rewards, improvement, etc.) aims to increase the performance of challenge recommendation that can a have a positive influence on players' engagement in the game. The latter "Ob7" points the effectiveness of the ML techniques to predict players play style for the next section of the game. The results showed that the designed system can predict the players' play styles with high performances, particularly when cumulative strategy is exploited.

Finally, Ob6 refers to the capability of the proposed play style recognition to automatically drive the play style. Our evaluation illustrated that there is no significant difference in proposed automatic extraction play styles compared to the ground truth that administered by the expert researchers, when margin players are excluded.

5.2.2 Efficiency

Efficiency was evaluated with two different objectives in phase 2 and 3. Ob3 and Ob5 are defined to evaluate the efficiency of the proposed APCGR and "Machine Learning" module. In order to reach Ob3, we compared the paid rewards for the unit of improvement by our proposed APCGR system vs. the manual approach. The results clearly confirmed that the challenge proposals by our system are across the board more economical in terms of rewards paid as incentives per unit of improvement (see Table 4.4). Time consumption was our concern to evaluate our proposed ML module in Ob5. Taking into account the effectiveness of the examined algorithms, Naive Bayes works better with our data.

In this work, we demonstrated that integrating recommender system and PCG strategy at generating and personalizing game content has a positive influence on players' engagement in the gamified system. The success of our proposed framework for tailoring the game content showed a great potential to utilize the tailored content as an effective tool for promoting green behavior. Moreover, it indicates that such integration can be easily employed in various gamification context to meet the gamified system's objectives by utilizing AI and recommender systems' techniques. However, our experiments highlight some limitations and open several additional research questions, both in terms of the proposed approach, as well as its evaluation. A clear constraint of the proposed framework that might affect the system's impact on engagement and behavioral change from a broad demographics of players, concerns the fact that it considers the elevation of player's status as the key player's objective.

5.3 Future Work

The work in this PhD thesis represents an initial stride towards mainly automating the process of generating, tailoring and recommending game content to the individual players, and validating the effectiveness and efficiency of this automation in gamification context. Although this work demonstrated many interesting and significant findings, it also brings up many research opportunities for further investigation.

In the future work, we will explore a number of extensions to the play style extraction framework in the context of gamification –in general– and –in particular– urban mobility system discussed in this dissertation. In the following we briefly discuss the way that the study could be extended at extracting players' play style in a gamified system.

5.3.1 Exploiting Unsupervised and Supervised ML algorithms to Extract Players' Play Style

As the first step of the future work, we started to implement player modeling concept in our Urban Mobility scenario to discern the play style of the players who participate in gamification. This enables the framework to advance the generation of personalized challenges (introduced in Chapter 3.2) not only based on players' game status, preferences or game objectives, but also taking into account players' characteristics that show how they actually behave in the game. Tailoring game content w.r.t the players' play style could have a value in reinforcing participants to be more immersed in the system.

To this end, we aim to build up the *Score framework* that we proposed and introduced in Chapter 3.4. Indeed, the framework is a rule-based approach since it can extract the play style based on the rules that are defined in the Rule Set (Section 4.10.1). This manually defining the rules limits the module in recognizing only the play styles, which are already defined in the Rule Set in a supervised fashion. To overcome this limitation such that the system can automatically and effectively recognize all the possible demeanors during the game, we investigate to design a framework that exploits the power of unsupervised and supervised ML algorithms. This would further establish the generalizability of our proposed play style-driven approach for personalizing gamified system.

As mentioned above, this extension of the play style recognition module has already commenced. To give a brief overview, we have constructed the first prototype of our framework mainly in a two-tier architecture of *Cluster-*

ing and *Rule Extraction.* Clustering concerns to automatically detect hidden patterns in human mobility data using unsupervised algorithms (for instance, Expectation Maximization (EM), K-means algorithms or/and self-organizing feature map (SOFM) [116, 152, 153]). In the second module, for the sake of interpret-ability, rule induction algorithm exploited to learn the extracted rules for a given cluster label. This module is in charge of using supervised algorithm (e.g., JRip that is a rule learner algorithm [154]) to find the rules between features and the labels, which are assigned in clustering step. In fact interpreting is the real motivation for us to use rule learning.

Although there are several statistics' techniques that can be used to assess clustering and rule extraction performances, the lack of "ground truth data" (labeled by human) to validate clustering and rule extraction challenge this framework that is constructed based on the unsupervised algorithm.

5.3.2 Design a Semi-Supervised Module to Discover Play Style of Players Modeling in Gamification

Another interesting future extension of the proposed play style recognition module could exploit semi-supervised algorithm to extract players' pattern from their behavior in the context of gamification. As it is always arduous and costly to label a large-scale data by hand (e.g., in our urban mobility scenario we can construct a survey or questionnaire based on Hexad Scale [98] in order to score and find their play styles during the game), semi-supervised algorithms can be used to find the players' pattern from a small portion of the data, (which are labeled by hand e.g., the questionnaire) in order to apply on the rest of the data.

5.3.3 Personalized Game Content For the Group of Players

The PCG framework developed and presented in Chapter 3.2 automates the generation of game content and personalizes to each individual player. We have shown that this automation is not only feasible but it is also effective and efficient in making individual extrinsic and intrinsic motivation [10] to improve players' behavior towards the gamified system's direction. An interesting extension would be to develop a framework that can set-up a group of players (like teams; from the players who are involved in the gamified system) in gamification so that the system is able to recommend personalized challenges to those groups. This may contribute to foster players' motivation in a group competition under the gamification goal.

Bibliography

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

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

[3] Sunny Consolvo, Katherine Everitt, Ian Smith, and James A Landay. Design requirements for technologies that encourage physical activity. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 457–466. ACM, 2006.

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

[5] Stephen Purpura, Victoria Schwanda, Kaiton Williams, William Stubler, and Phoebe Sengers. Fit4life: the design of a persuasive technology promoting healthy behavior and ideal weight. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 423–432. ACM, 2011.

[6] Tammy Toscos, Anne Faber, Shunying An, and Mona Praful Gandhi. Chick clique: persuasive technology to motivate teenage girls to exercise. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1873–1878. ACM, 2006.

[7] Tyler Pace, Shruti Ramalingam, and David Roedl. Celerometer and idling reminder: persuasive technology for school bus eco-driving. In

CHI'07 Extended Abstracts on Human Factors in Computing Systems, pages 2085–2090. ACM, 2007.

[8] Valerie Sugarman and Edward Lank. Designing persuasive technology to manage peak electricity demand in ontario homes. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1975–1984. ACM, 2015.

[9] Katie Seaborn and Deborah I. Fels. Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, 74:14 – 31, 2015.

[10] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *2014 47th Hawaii International Conference on System Sciences*. IEEE, 2014.

[11] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek '11, pages 9–15, New York, NY, USA, 2011. ACM.

[12] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. Gamification. using game-design elements in non-gaming contexts. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 2425–2428, New York, NY, USA, 2011. ACM.

[13] Thilo Kunkel and Daniel Funk. Sport fan engagement through gamification in a digital media environment. In *Presented at the sport marketing association conference*, 2014.

[14] Ralf Terlutter and Michael L Capella. The gamification of advertising: analysis and research directions of in-game advertising, advergames, and advertising in social network games. *Journal of Advertising*, 42(2-3):95–112, 2013.

[15] Jing Du, Yue Feng, and Changchang Zhou. Gamification for behavior change of occupants in campus buildings to affect improved energy efficiency. 2014.

[16] Raman Kazhamiakin, Annapaola Marconi, Alberto Martinelli, Marco Pistore, and Giuseppe Valetto. A gamification framework for the long-term engagement of smart citizens. In *Smart Cities Conference (ISC2), 2016 IEEE International*, pages 1–7. IEEE, 2016.

[17] Carl DiSalvo, Phoebe Sengers, and Hrönn Brynjarsdóttir. Mapping the landscape of sustainable hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1975–1984. ACM, 2010.

[18] Elaine M Huang. Building outwards from sustainable hci. *ACM Interactions*, 18(3):14–17, 2011.

[19] Lorenz M. Hilty and Bernard Aebischer, editors. *ICT Innovations for Sustainability*, volume 310 of *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2015.

[20] Pascal Lessel, Maximilian Altmeyer, and Antonio Krüger. Analysis of recycling capabilities of individuals and crowds to encourage and educate people to separate their garbage playfully. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, New York, NY, USA, 2015. ACM.

[21] Jonathan Simon, Marco Jahn, and Amro Al-Akkad. Saving energy at work: the design of a pervasive game for office spaces. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. ACM.

[22] Robert S Brewer, Nervo Verdezoto, Thomas Holst, and Mia Kruse Rasmussen. Tough shift: Exploring the complexities of shifting residential electricity use through a casual mobile game. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. ACM, 2015.

[23] Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A. Landay. Ubigreen: Investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1043–1052, New York, NY, USA, 2009. ACM.

[24] R. Kazhamiakin, A. Marconi, M. Perillo, M. Pistore, G. Valetto, L. Piras, F. Avesani, and N. Perri. Using gamification to incentivize sustainable urban mobility. In *2015 IEEE First International Smart Cities Conference (ISC2)*, pages 1–6, Oct 2015.

[25] Rosta Farzan, Joan M DiMicco, David R Millen, Casey Dugan, Werner Geyer, and Elizabeth A Brownholtz. Results from deploying a participation incentive mechanism within the enterprise. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 563–572. ACM, 2008.

[26] Juho Hamari, Jonna Koivisto, and Tuomas Pakkanen. Do persuasive technologies persuade?-a review of empirical studies. In *International Conference on Persuasive Technology*, pages 118–136. Springer, 2014.

[27] Paul Weiser, Dominik Bucher, Francesca Cellina, and Vanessa De Luca. A Taxonomy of Motivational Affordances for Meaningful Gamified and Persuasive Technologies. In *Proceedings of the 3rd International Conference on ICT for Sustainability (ICT4S)*, volume 22 of *Advances in Computer Science Research*, pages 271–280, Paris, 2015. Atlantis Press.

[28] Yefeng Liu, Todorka Alexandrova, and Tatsuo Nakajima. Gamifying intelligent environments. In *Proceedings of the 2011 international ACM workshop on Ubiquitous meta user interfaces*, pages 7–12. ACM, 2011.

[29] Jennifer Thom, David Millen, and Joan DiMicco. Removing gamification from an enterprise sns. In *Proceedings of the acm 2012 conference on computer supported cooperative work*, pages 1067–1070. ACM, 2012.

[30] Reza Khoshkangini, Giuseppe Valetto, and Annapaola Marconi. Generating personalized challenges to enhance the persuasive power of gamification. *Int. Work. Pers. Persuas. Technol*, 2017.

[31] Pietro Di Lena, Silvia Mirri, Catia Prandi, Paola Salomoni, and Giovanni Delnevo. In-vehicle human machine interface: An approach to enhance eco-driving behaviors. In *Proceedings of the 2017 ACM Workshop on Interacting with Smart Objects*, pages 7–12. ACM, 2017.

[32] Ricardo Lopes and Rafael Bidarra. Adaptivity challenges in games and simulations: a survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(2):85–99, 2011.

[33] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9.

[34] Qing Li and Byeong Man Kim. An approach for combining content-based and collaborative filters. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pages 17–24, 2003.

[35] David Pizzi, Jean-Luc Lugrin, Alex Whittaker, and Marc Cavazza. Automatic generation of game level solutions as storyboards. *IEEE Transactions on Computational Intelligence and AI in Games*, 2.

[36] Edirlei Soares de Lima, Bruno Feijó, and Antonio L Furtado. Hierarchical generation of dynamic and nondeterministic quests in games. In

Proceedings of the 11th Conference on Advances in Computer Entertainment Technology, page 24. ACM, 2014.

[37] Stephen Karpinskyj, Fabio Zambetta, and Lawrence Cavedon. Video game personalisation techniques: A comprehensive survey. *Entertainment Computing*, 5(4):211–218, 2014.

[38] Alexander Zook, Stephen Lee-Urban, Michael R Drinkwater, and Mark O Riedl. Skill-based mission generation: A data-driven temporal player modeling approach. In *Proceedings of the The third workshop on Procedural Content Generation in Games*, page 6. ACM, 2012.

[39] Alexander Zook, Stephen Lee-Urban, Mark O Riedl, Heather K Holden, Robert A Sottilare, and Keith W Brawner. Automated scenario generation: toward tailored and optimized military training in virtual environments. In *Proceedings of the international conference on the foundations of digital games*, pages 164–171. ACM, 2012.

[40] Mihaly Csikszentmihalyi. *Finding flow: The psychology of engagement with everyday life.* Basic Books, 1997.

[41] Jenova Chen. Flow in games (and everything else). *Communications of the ACM*, 50(4):31–34, 2007.

[42] Ben Cowley, Darryl Charles, Michaela Black, and Ray Hickey. Toward an understanding of flow in video games. *Comput. Entertain.*, 6(2):20:1–20:27, July 2008.

[43] Mihaly Csikszentmihalyi, Sami Abuhamdeh, and Jeanne Nakamura. Flow. In *Flow and the foundations of positive psychology*, pages 227–238. Springer, 2014.

[44] Seth Cooper, Christoph Sebastian Deterding, and Theo Tsapakos. Player rating systems for balancing human computation games. In *Proceedings of 1st International Joint Conference of DiGRA and FDG*. DIGRA Digital Games and Research Association, 2016.

[45] Arpad E Elo. *The rating of chessplayers, past and present.* Arco Pub., 1978.

[46] J. Togelius, R. De Nardi, and S. M. Lucas. Towards automatic personalised content creation for racing games. In *2007 IEEE Symposium on Computational Intelligence and Games*, pages 252–259, April 2007.

[47] Diana Lora, Antonio A Sánchez-Ruiz, Pedro A González-Calero, and Marco Antonio Gómez-Martín. Dynamic difficulty adjustment in tetris. In *FLAIRS Conference*, pages 335–339, 2016.

[48] Darryl Charles, A Kerr, M McNeill, M McAlister, M Black, J Kcklich, A Moore, and K Stringer. Player-centred game design: Player modelling and adaptive digital games. In *Proceedings of the Digital Games Research Conference*, volume 285, page 00100, 2005.

[49] Sauvik Das, Alexander Zook, and Mark O. Riedl. Examining game world topology personalization. In *Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3731–3734, 2015.

[50] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, 2011.

[51] Baptiste Monterrat, Elise Lavoué, and Sébastien George. Motivation for learning: Adaptive gamification for web-based learning environments. In *6th International Conference on Computer Supported Education (CSEDU 2014)*, pages 117–125, 2014.

[52] Luiz Rodrigues, Robson Parmezan Bonidia, and Jacques Duílio Brancher. A math educacional computer game using procedural content generation. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 756, 2017.

[53] Marko Balabanović and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, March 1997.

[54] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[55] Lian Duan, W Nick Street, and E Xu. Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems*, 5(2):169–181, 2011.

[56] J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM, 1999.

[57] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. Recommender system for predicting student performance. *Procedia Computer Science*, 1(2):2811–2819, 2010.

[58] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and

possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[59] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[60] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[61] Chumki Basu, Haym Hirsh, William Cohen, et al. Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai*, pages 714–720, 1998.

[62] Marco De Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Preference learning in recommender systems. *Preference Learning*, 41, 2009.

[63] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*, volume 60, 1999.

[64] Aristomenis S Lampropoulos, Paraskevi S Lampropoulou, and George A Tsihrintzis. A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis. *Multimedia Tools and Applications*, 59(1):241–258, 2012.

[65] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[66] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, January 2001.

[67] Chihiro Ono, Mori Kurokawa, Yoichi Motomura, and Hideki Asoh. A context-aware movie preference model using a bayesian network for recommendation and promotion. *User Modeling 2007*, pages 247–257, 2007.

[68] Mark Van Setten, Stanislav Pokraev, and Johan Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *Adaptive hypermedia and adaptive web-based systems*, pages 515–548. Springer, 2004.

[69] Simen Fivelstad Smaaberg, Nafiseh Shabib, and John Krogstie. A user-study on context-aware group recommendation for concerts. In *HT (Doctoral Consortium/Late-breaking Results/Workshops)*, volume 112, 2014.

[70] Cosimo Palmisano, Alexander Tuzhilin, and Michele Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, 20(11):1535–1549, 2008.

[71] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103–145, 2005.

[72] J. Bobadilla, F. Ortega, A. Hernando, and A. GutiÃľrrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109 – 132, 2013.

[73] Rafet Sifa, Christian Bauckhage, and Anders Drachen. Archetypal game recommender systems. In *LWA*, pages 45–56. Citeseer, 2014.

[74] Pavle Skocir, Luka Marusic, Marinko Marusic, and Ana Petric. *The MARS – A Multi-Agent Recommendation System for Games on Mobile Phones*, pages 104–113. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[75] W. L. Raffe, F. Zambetta, and X. Li. Neuroevolution of content layout in the pcg: Angry bots video game. In *2013 IEEE Congress on Evolutionary Computation*, pages 673–680, June 2013.

[76] B. Harrison, S. G. Ware, M. W. Fendt, and D. L. Roberts. A Survey and Analysis of Techniques for Player Behavior Prediction in Massively Multiplayer Online Role-Playing Games. *IEEE Transactions on Emerging Topics in Computing*, 3(2):260–274, June 2015.

[77] Erik Andersen, Sumit Gulwani, and Zoran Popovic. A trace-based framework for analyzing and synthesizing educational progressions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 773–782, New York, NY, USA, 2013. ACM.

[78] Eric Butler, Erik Andersen, Adam M. Smith, Sumit Gulwani, and Zoran Popović. Automatic game progression design through analysis of solution features. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2407–2416, New York, NY, USA, 2015. ACM.

[79] Fábio Silva, Cesar Analide, Luís Rosa, Gilberto Felgueiras, and Cedric Pimenta. Gamification, social networks and sustainable environments. *International Journal of Interactive Multimedia and Artificial Intelligence*, 2(4):52–59, 12/2013 2013.

[80] Mark O Riedl, Andrew Stern, Don Dini, and Jason Alderman. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, 4(2):23–42, 2008.

[81] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1, 2013.

[82] Adam M. Smith, Chris Lewis, Kenneth Hullet, and Anne Sullivan. An inclusive view of player modeling. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, FDG '11, pages 301–303, New York, NY, USA, 2011. ACM.

[83] Richard Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19, 1996.

[84] Nick Yee. The psychology of massively multi-user online role-playing games: Motivations, emotional investment, relationships and problematic usage. *Avatars at work and play*, pages 187–207, 2006.

[85] Dan Dixon. Player types and gamification. In *Proceedings of the CHI 2011 Workshop on Gamification*, 2011.

[86] Kirsi Pauliina Kallio, Frans Mäyrä, and Kirsikka Kaipainen. At least nine ways to play: Approaching gamer mentalities. *Games and Culture*, 6(4):327–353, 2011.

[87] Erik Van Der Werf, Jos WHM Uiterwijk, Eric Postma, and Jaap Van Den Herik. Local move prediction in go. In *International Conference on Computers and Games*, pages 393–412. Springer, 2002.

[88] Pavle Skocir, Luka Marusic, Marinko Marusic, and Ana Petric. The mars–a multi-agent recommendation system for games on mobile phones. In *Agent and Multi-Agent Systems. Technologies and Applications*, pages 104–113. Springer, 2012.

[89] Alessandro Canossa. *Meaning in Gameplay: Filtering Variables, Defining Metrics, Extracting Features and Creating Models for Gameplay Analysis*, pages 255–283. Springer London, London, 2013.

[90] Ji-Lung Hsieh and Chuen-Tsai Sun. Building a player strategy model by analyzing replays of real-time strategy games. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 3106–3111. IEEE, 2008.

[91] Ben G Weber, Michael Mateas, and Arnav Jhala. Using data mining to model player experience. In *FDG Workshop on Evaluating Player Experience in Games*, 2011.

[92] Anders Drachen, Alessandro Canossa, and Georgios N Yannakakis. Player modeling using self-organization in tomb raider: Underworld. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 1–8. IEEE, 2009.

[93] Christian Thurau, Christian Bauckhage, and Gerhard Sagerer. Combining self organizing maps and multilayer perceptrons to learn bot-behaviour for a commercial game. In *GAME-ON*, page 119, 2003.

[94] Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 443–448. SIAM, 2007.

[95] Martin Rajchl, John SH Baxter, A Jonathan McLeod, Jing Yuan, Wu Qiu, Terry M Peters, and Ali R Khan. Hierarchical max-flow segmentation framework for multi-atlas segmentation with kohonen self-organizing map based gaussian mixture modeling. *Medical image analysis*, 27:45–56, 2016.

[96] Richard A Bartle. *Designing virtual worlds.* New Riders, 2004.

[97] Andrzej Marczewski. User types. *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design*, 1:65–80, 2015.

[98] Gustavo F. Tondello, Rina R. Wehbe, Lisa Diamond, Marc Busch, Andrzej Marczewski, and Lennart E. Nacke. The gamification user types hexad scale. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '16, pages 229–243, New York, NY, USA, 2016. ACM.

[99] Lennart E Nacke, Chris Bateman, and Regan L Mandryk. Brainhex: preliminary results from a neurobiological gamer typology survey. In *International Conference on Entertainment Computing*, pages 288–293. Springer, 2011.

[100] Yan Xu, Erika Shehan Poole, Andrew D. Miller, Elsa Eiriksdottir, Dan Kestranek, Richard Catrambone, and Elizabeth D. Mynatt. This is not a one-horse race: Understanding player types in multiplayer pervasive health games for youth. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 843–852, New York, NY, USA, 2012. ACM.

[101] Lauren S Ferro, Steffen P Walz, and Stefan Greuter. Towards personalised, gamified systems: an investigation into game design, personality and player typologies. In *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death*, page 7. ACM, 2013.

[102] Baptiste Monterrat, Michel Desmarais, Elise Lavoué, and Sébastien George. A player model for adaptive gamification in learning environments. In *International Conference on Artificial Intelligence in Education*, pages 297–306. Springer, 2015.

[103] Richard E Ferdig. *Handbook of research on effective electronic gaming in education*. IGI Global, 2008.

[104] Carrie Heeter and B Winn. Implications of gender, player type and learning strategies for the design of games for learning. *Beyond Barbie to Mortal Combat: New perspectives on games, gender, and computing*, 2008.

[105] Jenova Chen. Flow in games (and everything else). *Commun. ACM*, 50(4):31–34, April 2007.

[106] Barry Smyth and Paul Cotter. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2):53–59, 2000.

[107] Josep Valls-Vargas, Santiago Ontanón, and Jichen Zhu. Exploring player trace segmentation for dynamic play style prediction. In *Proceedings of the Eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 93–99, 2015.

[108] Brad Paras. Game, motivation, and effective learning: An integrated model for educational game design. 2005.

[109] Raman Kazhamiakin, Annapaola Marconi, Alberto Martinelli, Marco Pistore, and Giuseppe Valetto. A gamification framework for the long-term engagement of smart citizens. In *Smart Cities Conference (ISC2), 2016 IEEE International*, pages 1–7.

[110] Stephen Karpinskyj, Fabio Zambetta, and Lawrence Cavedon. Video game personalisation techniques: A comprehensive survey. *Entertainment Computing*, 5(4):211–218, 2014.

[111] Christoph Klimmt, Tilo Hartmann, and Andreas Frey. Effectance and control as determinants of video game enjoyment. *Cyberpsychology & behavior*, 10(6):845–848, 2007.

[112] Stefan Engeser and Falko Rheinberg. Flow, performance and moderators of challenge-skill balance. *Motivation and Emotion*, 32(3):158–172, 2008.

[113] Derek Lomas, Kishan Patel, Jodi L. Forlizzi, and Kenneth R. Koedinger. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 89–98, New York, NY, USA, 2013. ACM.

[114] Alena Denisova and Paul Cairns. Adaptation in digital games: The effect of challenge adjustment on player performance and experience. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '15, pages 97–101, New York, NY, USA, 2015. ACM.

[115] Rowan Tulloch. Reconceptualising gamification: Play and pedagody. *Digital Culture & Education*, 6(4), 2014.

[116] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[117] Edward Deci and Richard M Ryan. *Intrinsic motivation and self-determination in human behavior*. Springer Science & Business Media, 1985.

[118] Michele D Dickey. Game design and learning: A conjectural analysis of how massively multiple online role-playing games (mmorpgs) foster intrinsic motivation. *Educational Technology Research and Development*, 55(3):253–273, 2007.

[119] Robin Hunicke. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 429–433. ACM, 2005.

[120] Changchun Liu, Pramila Agrawal, Nilanjan Sarkar, and Shuo Chen. Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. *International Journal of Human-Computer Interaction*, 25(6):506–529, 2009.

[121] Justin Chumbley and Mark Griffiths. Affect and the computer game player: the effect of gender, personality, and game reinforcement structure on affective responses to computer game-play. *CyberPsychology & Behavior*, 9(3):308–316, 2006.

[122] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[123] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[124] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[125] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: A survey and experimental evaluation. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 306–313. IEEE, 2002.

[126] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4):491–502, 2005.

[127] Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. Player Modeling. In Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius, editors, *Artificial and Computational Intelligence in Games*, volume 6 of *Dagstuhl Follow-Ups*, pages 45–59. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013.

[128] Tobias Mahlmann, Anders Drachen, Julian Togelius, Alessandro Canossa, and Georgios N Yannakakis. Predicting player behavior in tomb raider: Underworld. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 178–185. IEEE, 2010.

[129] Georgios N Yannakakis and Manolis Maragoudakis. Player modeling impact on player's entertainment in computer games. *Lecture notes in computer science*, 3538:74, 2005.

[130] Georgios N Yannakakis and John Hallam. A scheme for creating digital entertainment with substance. In *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 119–124, 2005.

[131] Michael D Kickmeier-Rust and Dietrich Albert. Educationally adaptive: Balancing serious games. *International Journal of Computer Science in Sport (International Association of Computer Science in Sport)*, 11(1), 2012.

[132] Ken Hartsook, Alexander Zook, Sauvik Das, and Mark O Riedl. Toward supporting stories with procedurally generated game worlds. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pages 297–304. IEEE, 2011.

[133] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[134] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[135] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[136] Pang-Ning Tan et al. *Introduction to data mining*. Pearson Education India, 2006.

[137] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[138] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[139] William C Blackwelder. âĂIJproving the null hypothesisâĂİ in clinical trials. *Controlled clinical trials*, 3(4):345–353, 1982.

[140] Donald J Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of pharmacokinetics and biopharmaceutics*, 15(6):657–680, 1987.

[141] Andrew P Robinson and Robert E Froese. Model validation using equivalence tests. *Ecological Modelling*, 176(3-4):349–358, 2004.

[142] Alan Agresti and Brent A Coull. Approximate is better than âĂIJexactâĂİ for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[143] Gerald Jurasinski, Franziska Koebsch, Anke Guenther, and Sascha Beetz. *flux: Flux rate calculation from dynamic closed chamber measurements*, 2014. R package version 0.3-0.

[144] William R Rice. Analyzing tables of statistical tests. *Evolution*, 43(1):223–225, 1989.

[145] Edmund A Gehan. A generalized wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52(1-2):203–224, 1965.

[146] G Jurasinski, F Koebsch, and U Hagemann. Flux rate calculation from dynamic closed chamber measurements, r package version 0.2-1, 2012.

[147] Aroutis N Foster. The process of learning in a simulation strategy game: Disciplinary knowledge construction. *Journal of Educational Computing Research*, 45(1):1–27, 2011.

[148] J Valls-Vargas, A Kahl, J Patterson, G Muschio, A Foster, and J Zhu. Designing and tracking play styles in solving the incognitum. In *Proceedings of the Games+ Learning+ Society Conference, to appear. Games+ Learning+ Society*, 2015.

[149] Andrew J Elliot and Holly A McGregor. A $2\times$ 2 achievement goal framework. *Journal of personality and social psychology*, 80(3):501, 2001.

[150] J Reynaldo A Santos. CronbachâĂŹs alpha: A tool for assessing the reliability of scales. *Journal of extension*, 37(2):1–5, 1999.

[151] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[152] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[153] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on neural networks*, 11(3):586–600, 2000.

[154] Jens Hühn and Eyke Hüllermeier. Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, 2009.