João Felipe Humenhuk

Automatic Churn Labeling and Prediction in MOBA Games

Curitiba - PR, Brasil

2021

João Felipe Humenhuk

Automatic Churn Labeling and Prediction in MOBA Games

Master's Dissertation Project presented to the Graduate Program in Informatics of the Pontificia Universidade Católica do Paraná as a partial requirement to obtain the title of Master in Computer Science.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Advisor: Emerson Cabrera Paraiso

Curitiba - PR, Brasil

2021

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Edilene de Oliveira dos Santos CRB-9/1636

H921a
2021

Humenhuk, João Felipe
Automatic Churn Labeling and Prediction in MOBA Games / João Felipe Humenhuk ; advisor: Emerson Cabrera Paraiso. -- 2021
81 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2021
Bibliografia: f. 77-81

1. Informática. 2. Jogos para computador. 3. Inteligência artificial. 4. Redes neurais (Computação). 5. Previsão. I. Paraiso, Emerson Cabrera. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título

. CDD. 20.ed. – 004

PUCPR
GRUPO MARISTA

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

41-2021

DECLARAÇÃO

Declaro para os devidos fins que o aluno **JOÃO FELIPE HUMENHUK**, defendeu sua dissertação de Mestrado intitulada **"AUTOMATIC CHURN LABELING AND PREDICTION IN MOBA GAMES"**, na área de concentração Ciência da Computação, no dia 08 de abril de 2021, no qual foi aprovado.

Declaro ainda que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade, firmo a presente declaração.

Curitiba, 08 de junho de 2021.

Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica do Paraná

Acknowledgements

First, I would like to thank all my family and friends that helped me through the whole process with their love and care. Especially, I would like to thank my parents, Ana and João, and my girlfriend Adriane that always cheered me up and made obstacles easier to overcome. I also want to thank my advisor Prof. Dr. Emerson Cabrera Paraiso, the Master's qualification and defense examining board members Prof. Dr. Bruno Feijo, Prof. Dr. Fabrício Enembreck, Prof. Dr. Júlio César Nievola, Prof. Dr. Vinicius Mourão Alves de Souza, and my lab-mate Luiz for all the precious feedback. Last but not less valuable, I would like to thank Siemens Ltd for their support in this research and Riot Games for providing access to their game data. This work was possible only due to the help of all of you.

Abstract

Churn can be interpreted as customer defection and can be considered the most critical challenge in the Game Analytics domain because of its impact on the game industry profit. When predicting churn, the first step is defining what is considered churn, which can change depending on the approaches and players' behaviors. From an extensive literature review, we have identified three frequent limitations in the churn prediction task: the nonexistence of true labels, the static definition of churn, and limitations on the adopted labeling approaches. Two novel methods and one novel labeling approach were proposed to overcome the identified challenges. Finished the process of defining what churn is and labeling the players according to it, starts the churn prediction task. There are two key factors associated with players' retention: in-game achievements and social influences. Both elements are addressed in this work by joining a metric called Commitment, and a social metric, which encompasses the churn influence a player receives from other players, and the graph structure in which a player belongs. The Commitment metric demands a score representing each players' performance. Since there is no replicable method capable of computing such a required score feature, a new process is suggested together with a way to evaluate it. This process returns a value representing the players' performance in a Multiplayer Online Battle Arena match. By deploying the proposed labeling methods and technique in two games, it was possible to conclude that the novel labeling approach can better identify the churn definition changes, and the methods can justify when the redefinition of churn and the classifier's retraining should happen. These results are valuable for the game context, with the potential to be extended to other contexts that carry the same notion of churn (e.g., mobile applications and social networks), because it delivers more reliable labels and classification performance. Regarding the players' evaluation process, aside from the calculation of Commitment, it is possible to use the score obtained in other tasks, such as systems that help the players improve their performance and more accurate Dynamic Difficult Adjustment of Artificial Intelligence Agents. Lastly, the proposed churn prediction method outperformed the F1-Score of related works, highlighting the proposed metrics' importance.

Keywords: game analytics, churn labeling, players evaluation, churn prediction, and MOBA.

Resumo

Churn pode ser interpretado como deserção de clientes e pode ser considerado o desafio mais crítico na área de *Game Analytics* por causa do seu impacto no lucro da indústria de jogos. Ao prever *churn*, o primeiro passo é definir o quê pode ser considerado *churn*. Essa definição pode mudar dependendo do comportamendo dos jogadores e das abordagens utilizadas. Ao realizar uma extensiva revisão da literatura, foram identificados três limitações frequentes na tarefa de predição de *churn*: a inexistência de *true labels*, a definição estática de *churn*, e limitações nas técnicas de rotulamento adotadas. Dois novos método e uma nova abordagem de rotulamento foram propostos para superar os desafios encontrados. Terminado o processo de definição de *churn* e rotulamento dos jogadores de acordo com esta definição, começa a tarefa de predição de *churn*. Existem dois fatores-chave associados com a retenção de jogadores: conquistas obtidas dentro do jogo e influências sociais. Ambos os elementos são tratados neste trabalho por meio do uso conjunto de uma métrica chamada *Commitment*, e de uma métrica social, a qual engloba a influência de *churn* que um jogador recebe de outros jogadores, e a estrutura do grafo ao qual ele pertence. A métrica *Commitment* exige um valor que representa o desempenho de cada jogador. Por conta da inexistência de um método replicável que exerça esta função, um processo apto a retornar um valor representando o desempenho de um jogador em uma partida de um jogo *Multiplayer Online Battle Arena* é proposto junto com um procedimento de avaliação do método. Ao aplicar a técnica e os métodos de rotulamento propostos em dois jogos, foi possível concluir que a nova abordagem de rotulamento pode melhor identificar as mudanças na definição de *churn*, e os métodos podem justificar quando uma redefinição do que é *churn* e o re-treino do classificador deveriam acontecer. Estes resultados são valiosos no contexto de jogos, com o potencial de serem extendidos para outros contextos que possuem a mesma noção de *churn* (e.g., aplicações para celular e redes sociais), por conta da entrega de rótulos e de uma performance do classificador mais confiáveis. Em relação ao processo de avaliação de jogadores, além de poder ser utilizado para o cálculo do *Commitment*, é possível usá-lo em outras tarefas, como em sistemas que ajudam jogadores a melhorarem suas habilidades e sistemas mais precisos de Ajuste Dinâmico de Dificuldade de Agentes de Inteligência Artifical. Para finalizar, o método de predição de *churn* proposto superou o *F1-Score* de trabalhos relacionados, destacando a importância das métricas propostas.

Palavras-chave: *game analytics*, rotulamento de *churn*, avaliação de jogadores, predição de *churn*, e MOBA.

List of Figures

List of Tables

List of abbreviations and acronyms

ADC	Attack Damage Carry
AIA	Artificial Intelligence Agent
ANN	Artificial Neural Network
API	Application Programming Interface
CDCR	Churn Definition Change Rate
DDA	Dynamic Difficult Adjustment
FV	Fixed Value
GaaS	Game as a Service
HA	Howling Abyss
HMM	Hidden Markov Model
ID	Identification
IFV	Individual Fixed Value
LOL	League Of Legends
MLP	Multilayer Perceptron
MMORPG	Massive Multiplayer Online Role-Playing Game
MOBA	Multiplayer Online Battle Arena
RQ	Research Question
SotA	State of the Art
SR	Summoners' Rift
SW	Sliding Window
WOW	World of Warcraft

Contents

1 Introduction

The profitability of the digital game (only “game” hereinafter) industry has attracted researches in the domain, being separated by games’ purposes (e.g., health, educational, entertainment). Unlike Serious Games that intend to change players’ behaviors or augment their knowledge about subjects, Entertainment Games provide an ambient to amuse and distract its players, who play it voluntarily. In the Entertainment Games category, there are numerous genres (e.g., First Person Shooter, Massive Multiplayer Online Role-Playing Game (MMORPG), Multiplayer Online Battle Arena (MOBA), etc.) that differ from each other in its game design, for instance, its objective, mechanics and artistic design. Aside from these differences, each game can also be classified regarding its business model, for instance, games Pay-To-Play and Free-To-Play, and their interactions with the game producers. With the increase of the market competition, the producers need to monitor the users’ behavior, gather information about possible issues, and launch improvements to the games. These needs changed the way games were maintained, from a perspective in which game development is considered finished after its release to a continuous development perspective without an apparent dead end, the idea of Game as a Service (GaaS) (CLARK, 2014). The GaaS management approach considers that the players’ motivations can be identified and maintained over time through Game Analytics analysis associated with the release of new game content. Such new management can keep the active users playing longer and motivate new ones to start playing, resulting in the profit increase (EL-NASR; DRACHEN; CANOSSA, 2016).

The GaaS policy’s profitability has many sources, such as in-game purchases, subscriptions, and the game purchase itself. The game purchase source relies on the players’ initial interest in playing the game, whereas the other sources are related to the continuous enjoyment in playing the game (KUMMER; NIEVOLA; PARAISO, 2017b). The GaaS approaches carry additional game management challenges, such as user profiling, game upgrades management, and churn prediction. Churn is described as customer defection and can be considered the most critical challenge because the industry profit comes from the users playing the game (HADIJI et al., 2014). Besides, it is cheaper to maintain active players for up to six months than acquire new ones (KARNSTEDT et al., 2010; YUAN et al., 2017).

Although churn is an old problem present in different businesses (e.g., telecommunications, TV, banking, or games), the first propositions of models to predict it started at the end of the twentieth-century (MASAND et al., 1999; MOZER et al., 2000). Churn in entertainment games differs from other contexts due to its nature of voluntary usage, which is an additional challenge, as it implies that a player can stop or return to play at any time and without notice (a non-contractual bond) (TAMADDONI; STAKHOVYCH; EWING, 2016). Aside from that, each game can have a different defined period to classify a player as a churner or non-churner, and that period could change over time. These unique characteristics increase the problem's difficulty and make the models of more well-studied domains, like telecommunications, unfeasible.

When predicting churn, the first challenge regards the labeling of the players. To understand and overcome the existent challenges in players' churn labeling, we elaborate two Research Questions (RQs). The first step is to identify the challenges, represented by RQ1: "What are the limitations present on the churn labeling task in games?". This question is answered based on related works, including methods, procedures, encountered challenges, and future works. This analysis identified three possible problems regarding a true label, a static churn definition, and limitations of the approaches used. The second step aims at verifying if they are indeed issues through the following RQ2: "Should the churn definition be updated?". A novel evaluation method is proposed to answer it, capable of quantitatively comparing the labeling approaches regarding the impact of a change in the churn definition over time. A new labeling approach that improves the most used ones' weaknesses and a method ("automatic labeler" henceforth) capable of automatically and periodically defining what behavior can be considered churn, based on data, are proposed to solve the encountered limitations. Acknowledge "behavior" as the players' frequency, days played and not played. To ensure that the method is context-free, no other metrics like performance or disengagement are approached since they can restrain a desirable general notion of churn or cause bias in the prediction.

Aside from the churn labeling, to predict if a player will churn, it is necessary to verify the factors that maintain him/her playing. As concluded by (PARK et al., 2017), the key factors for player retention are related to achievements and social interactions within the game. Considering their importance for player retention, these features are significant inputs to predict churn because they can represent the players' reasons to continue playing.

To capture the achievement feature, a metric called Commitment was used, which considers how long a player plays the game and how he/she plays it, joining engagement and performance (KUMMER; NIEVOLA; PARAISO, 2017a). Another essential characteristic of this metric that makes it useful for the proposed method is that it adapts to the users and can evolve together with the game.

Even though stated by a lot of studies, such as (GAJADHAR; KORT; IJSSELSTEIJN, 2008) and (CHEN et al., 2006), that social features carry essential information, most studies focus on the achievement aspect, considering the players' engagement, and only a few studies acknowledge the social aspects. A proposed systematic mapping identified 13 works that studies churn prediction in games, where, interestingly, only three of them considered social aspects, and none utilized it in the MOBA context.

Considering the effect of the social features on the model's performance, the little research done in MOBA games, its popularity, and its collaborative design, this research focused on predicting churn in a MOBA game. A MOBA game can be described as a team versus team strategic game. League of Legends (LOL), the most played MOBA game, has one of the highest quantities of people playing the game and watching championships (MORA-CANTALLOPS; SICILIA, 2018). Due to its success and facility in the data extraction process, the LOL's data was used in this work.

The Commitment metric requires a way to evaluate the players' performance. Even though numerous systems and modifications of these systems were proposed (ELO, 1978; GLICKMAN, 1995; HERBRICH; MINKA; GRAEPEL, 2007), their purpose was to give the players a score corresponding to their skill level. Since each character plays a different role in a MOBA game and each match is unique, these systems lack a way of evaluating a player based on its actions in a specific match, not its history.

Some works focus on the MOBA genre, proposing winning prediction on the competitive scenario (HODGE et al., 2019), Dynamic Difficult Adjustment (DDA) (SILVA; SILVA; CHAIMOWICZ, 2017), analysis of players' performance (CAPLAR; SUZNJEVIC; MATIJASEVIC, 2013), player skill evaluation (PRAKANNOPPAKUN; SINTHUPINYO, 2016), and game-play styles (FERGUSON et al., 2020) ranking. They all have some way to evaluate the players, but none give a value representing each player's performance for each match, disregarding history. To the best of our knowledge, only (PRAKANNOPPAKUN; SINTHUPINYO, 2016) achieve this task by deploying an Artificial Neural Network (ANN) with all characters' attributes in the game and training it to predict a bigger score for the winning team than the losing team. Their work proposes an innovative idea to solve the player evaluation problem in MOBA games but lacks details about the method and experiments that could improve its performance. It also lacks analysis regarding another MOBA game and an approach to evaluate the model's performance, disregarding the player's rank.

Wishing to address the problems encountered in the players' performance evaluation method, this work aims to improve the method proposed by performing modifications and experiments in a larger dataset of another MOBA game. It also clarifies missing details about the method execution and presents an evaluation procedure to compare the changes.

In summary, this research address all the major problems encountered when predicting churn in a MOBA game. First, regarding the players' churn labeling, other researchers from industry or academia can apply, explore, and take advantage of the proposed automatic labeler, new labeling approach, and the evaluation method on their acting contexts. The deployment of the three techniques (i.e. evaluate the players' behavior individually, define churn and label each player, and assess the need to relabel the players and retrain the classifier) and their results are illustrated in Figure 1. The new labeling

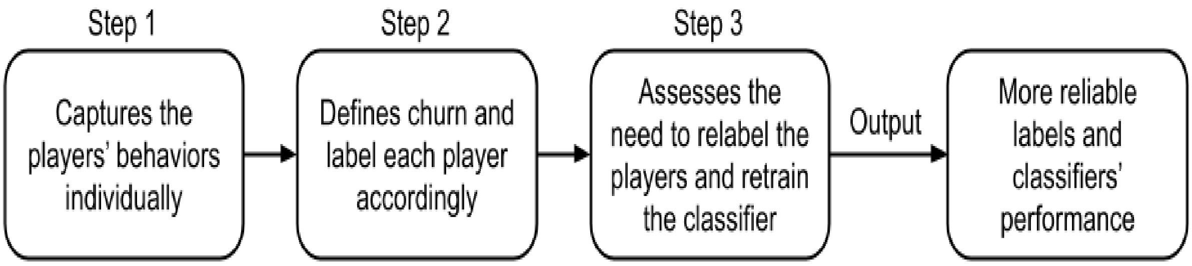
Step 1
Captures the players' behaviors individually
Step 2
Defines churn and label each player accordingly
Step 3
Assesses the need to relabel the players and retrain the classifier
Output
More reliable labels and classifiers' performance

Figure 1 – Linkage between the three techniques

approach (Step 1) gives more representative players' behavior information to the automatic labeler (Step 2). Using information that better represents the players, the automatic labeler provides more reliable labels to the evaluation method (Step 3). With more confident labels, the evaluation method can better assess the need to relabel the players and retrain the classifier. Using the outputs delivered from the three techniques' linkage, it is possible to acquire more attested labels and classifiers' performance. Aside from the proposed labeling approach, the evaluation method can be used over new churn labeling approaches, and the whole process can be applied in other games that were not approached in this work.

By deploying the improved players' performance evaluation method, other authors can utilize it in different tasks such as win prediction, systems to help players improve their performance, more accurate Dynamic Difficult Adjustment (DDA) of Artificial Intelligence Agents (AIA), algorithms that can help analysts in a competitive environment, and game-play styles ranking.

Lastly, the combination of the social metric and the Commitment showed its value to the churn prediction task. Both metrics can be used as a new baseline by other authors that aim to predict churn using the achievement and social features.

Using the whole proposed method, the industry can better understand its users, helping with strategic decisions about retaining campaigns and incresing its profit.

1.1 Motivation

Due to the high impact on the game industry's profit, predicting churn is an essential task but still has unsolved challenges. It usually has an assumption on the period that a player stops playing, and does not return, to label this player as a churner. This churn definition could change because a game can evolve and modify through time together with the players' behaviors (ZHU; LI; ZHAO, 2010), but this comportment is disregarded. Generally, works assume empirically this period, but, as done by (YANG et al., 2019), it can be defined using the game's data, which is more reliable. Even though their work improves the actual method used to decide this duration, it was done manually, and the definition is static, meaning that it does not include that a game and/or its players' behaviors can change over time.

Conscious of these problems, this research's first contribution is the understanding and detailing of the recurrent challenges present in the players' churn labeling task. The next contribution is the solution to the issues encountered, achieved by the proposition of an automatic labeler. This algorithm can automatically and periodically determine and update the churn definition used to label each player as a churner or non-churner. Since a way to evaluate the change in the churn definition is necessary and nonexistent, a procedure that can quantitatively compare different churn labeling approaches is also proposed. The evaluation procedure results can be used as indicators that a change in the churn definition and the classifier's retraining are needed. Lastly, since a couple of weaknesses were found in commonly used labeling techniques, a novel labeling approach that improves the flaws found is proposed.

Aside from the churn definition, the features used in related works focus on only one of the two key factors, the achievement, and there are only three works that combined both. Even though these studies applied both key factors in the churn prediction task, they all used achievement measures that do not evolve with the game and do not deal with the player base as one. This consideration is essential because numerous games change the game's evolution rate, and these changes will modify the metric's behavior. Commitment is a metric proposed by (KUMMER; NIEVOLA; PARAISO, 2017a) that will be used to address this problem. Commitment considers players' engagement and performance regarding a game's data set and delimits a range for high, average, or low degree of players' Commitment. Since the Commitment metric needs a way to evaluate the players' performance and there is no replicable method able to return a value that represents each player's performance in a MOBA game, such a method is detailed and appraised in a novel evaluation approach.

Aside from the achievement feature, all the related works that consider social features do not use it in the MOBA context, a highly social and cooperative game genre. Bearing this in mind, this work presents a social metric that joins features from several related works, encompassing churn influence received from other players and graph structures in the MOBA context.

A more detailed view of each component of the proposed methods can be seen in Chapter 5, where it will be described the automatic labeler, the new labeling approach, the labeling evaluation method, how the players' performance evaluation and each feature is calculated, and the deployment of the classifier for the churn prediction task.

1.2 Objectives

This research aims to propose a method that predicts churn in MOBA games based on a novel labeling procedure. To achieve it, the following specific objectives are required:

- To develop an algorithm responsible for periodically label the players as churners or non-churners according to the data;
- To develop a method that outputs a value representing a player's performance in a MOBA game match;
- To verify the applicability of Commitment in a MOBA game;
- To verify the applicability of social metrics in a MOBA game;
- To develop a method capable of predicting churn;
- To evaluate the proposed method;

1.3 Research Questions

This research answers the following questions:

- RQ1) What are the issues present on the churn labeling task in games?
- RQ2) Should the churn definition be updated?

1.4 Working Hypotheses

The hypotheses of this research are:

- H1) The automatic labeler provides more reliable labels than static definitions;
- H2) The proposed labeling approach better represents the players' churn behavior;
- H3) Commitment can be used in MOBA games;
- H4) Social features improve the churn prediction classifier's performance in a MOBA game.

1.5 Scientific Contribution and Technology Transfer

The research's scientific contribution can be separated into four groups: datasets, churn labeling, players' performance evaluation in MOBA games, and the importance of the Commitment and social features in the churn prediction task in a MOBA game. The first contribution is the availability of two datasets. One contains various stats from 180,468 match outcomes of the game LOL, while the other has 23 months of 2,400 players' log history.

Regarding the churn labeling, an automatic labeler, a novel labeling approach, and a labeling evaluation method were proposed. The automatic labeler solves issues regarding static, manual, and empirical churn definitions, delivering more reliable labels. The novel labeling approach improves the weaknesses fold in commonly used labeling techniques, implicating a more accurate representation of the players' churn behavior. Finally, the labeling evaluation method returns a value indicating the impact of a change in the churn definition caused in the labels. This value can be used to indicate that the churn definition should be updated and the model retrained. All the contributions can be used by other authors in different game genres and have the potential to be extended to other domains (e.g., mobile applications and social networks).

The third group of contributions regards the players' performance evaluation in a MOBA match. To the best of our knowledge, only one article proposes an idea to solve this challenge. Still, it lacks experiments in a larger dataset, and a way to evaluate the method disregarding the players' rank, which considers its history. In this research, all the method implementation details are described, experiments were made to improve the model's performance, and an evaluation procedure was proposed to appraise the results obtained. Other authors can replicate the method with these contributions, utilize it in many different applications, suggest changes, and compare them with the baseline proposed.

Lastly, this research shows the churn classification task results in a MOBA game using State of the Art (SotA) articles' metrics and the ones suggested in this work. The model deployed can be used as a baseline for other works that aim to predict churn in MOBA games using achievement and social features.

The industry can use all the contributions listed above to obtain a more reliable model's performance in the churn prediction task, better churn behavior representation, an indicative of redefining what is churn and retrain the classifier, and a baseline model for predicting churn in a MOBA game. Aside from the churn prediction task, the players' performance evaluation method can give each player a score in a MOBA match, and this score can be used in many commercial applications like in systems aiding players improve their performance, in more informative analysis of competitive matches or practices, among others.

1.6 Scope

This research's scope encompasses the extraction and preparation of the data and the creation of a method capable of automatically and periodically defining and updating what behavior can be considered churn. It also includes the churn prediction in a MOBA game using achievement and social features. The actions that could be taken to maintain possible churners are not accountable in this research.

Although this research is limited to evaluating the method only on a MOBA game, the proposed labeling approach and methods have the potential to be extended to other game genres and churn domains.

1.7 Document Organization

This document is organized into eight chapters. Chapter 2 presents the basic concepts to understand this work. Chapter 3 contains the state of the art methods to label churn, evaluate players' performance, and predict churn in games. Chapter 4 describes the methodology employed. Chapter 5 details the proposed method. Chapter 6 reports the experiments performed in this research. In Chapter 7 the results obtained from the experiments are showed and discussed. Lastly, Chapter 8 contains a summary of the work done, the answers to the research questions and hypotheses, the contributions, and future works.

2 Basic concepts

This chapter explains the basic concepts needed for the understanding of this work, assuming previous knowledge of supervised learning and classification algorithms. First, Section 2.1 explains the differences between two game categories, specifies which was chosen, and describes various genres among this category. Next, in Section 2.2 the genre chosen to be used in this work is detailed, describing its origin, characteristics, objectives, and how it is played. Section 2.3 explains the domain of player retention and presents the key-factors for retaining players, which can be used as features in the churn prediction task. Bearing in mind the two key-factors for maintaining players, achievement and social, two metrics are needed to encompass both. In Section 2.4 the metric Commitment, used to cover the achievement aspect, is explained, and in Section 2.5 all the necessary information regards graphs and some of its metrics are present. The information regarding graphs is required because the other key-factor, social, will be constructed using metrics from a graph built using game usage data. Last, in Section 2.6, the main objective of this research, predict churn, is explained, and the main differences from churn in other domains and games are presented together with the two most common techniques to predict it.

2.1 Digital Games

Digital games can be designed with different purposes and objectives. For instance, games that focus on improving players' health or knowledge are considered Serious Games, while games that focus on distracting, entertaining, and amusing their players, are called Entertainment Games. The main difference, aside from their purposes, is that Entertainment games are played voluntarily. A player can start or stop playing whenever he/she wants. This voluntary aspect is essential because this research objective is to predict churn considering features that maintain the user playing, and therefore the obligation facet is not accounted for.

Depending on a game's design, it is classified as a specific or a mixture of genres. There are numerous genres, like Role-Playing Game, MMORPG, MOBA, First Person Shooter, Platform, etc. Each one with its characteristics. Amid the genres, MOBA games have little research about players' performance evaluation and churn prediction. Aside from related works, it has high popularity and demand for social interactions to achieve a team's common goal, victory. This works' objectives and the MOBA's characteristics, make MOBA a great genre to be studied.

2.2 MOBA games

A Multiplayer Online Battle Arena (MOBA) game is an Entertainment Game and can be described as a team versus team strategic game. Each team, generally composed of five players, has the objective of destroying the opponents' Nexus, the most vital structure of a team and the closest one to its base. Each player controls a character containing unique abilities and status, used to fight the enemies' characters. When the life points of a character reach zero, it is considered dead and will need to wait several seconds until it respawns in the base and can be used again. After a team destroys the opponents' Nexus, it is considered the winner of the match. All the resources gathered (e.g., experience points, items, gold, powers, etc.) are erased, meaning that each game is isolated from the others.

There are numerous MOBA games such as Defense of the Ancients 2, League of Legends (LOL), Heroes of Newerth, among others. They change in graphics, characters, and other game designs, but all have the same principles defining a MOBA game.

LOL was launched in 2009 by Riot Games[1], is the most popular MOBA game both in playing and watching, have more than 140 champions[2], playable characters, and an Application Programming Interface (API)[3] that can be used to extract usage game data. Due to its stability, popularity (MORA-CANTALLOPS; SICILIA, 2018), and easy access to usage game data, LOL data was used in this research. For the rest of the dissertation, LOL game design was used as the standard to facilitate the explanation, but it could be adapted to any game in the same genre.

Each match in a MOBA game is independent, meaning the players' level, gold, and items are reset, so everyone starts the game with the same resources and status. This characteristic of having independent matches, or instances, categorizes the genre as an instanced game.

Before the match properly starts, the players can create a group with friends they want to play together in the same team, and the game will fill the remaining slots. A match can be divided into three consecutive phases, team formation, pick and ban, and gameplay.

In the team formation, players are assigned by the game to a role, varying between Top Lane, Jungle, Mid Lane, Attack Damage Carry (ADC), and Support. Each role has its function and designated position on the map for the beginning of the game, where it will confront an opponent from the other team. It is crucial to notice that, even though, in general, that is the case, there are numerous strategies and adaptations in the standard way the game is supposed to be played. After ten players are selected, separated into two

[1] <https://www.riotgames.com/en>

[2] <https://br.leagueoflegends.com/en-us/champions/>

[3] <https://developer.riotgames.com/apis>

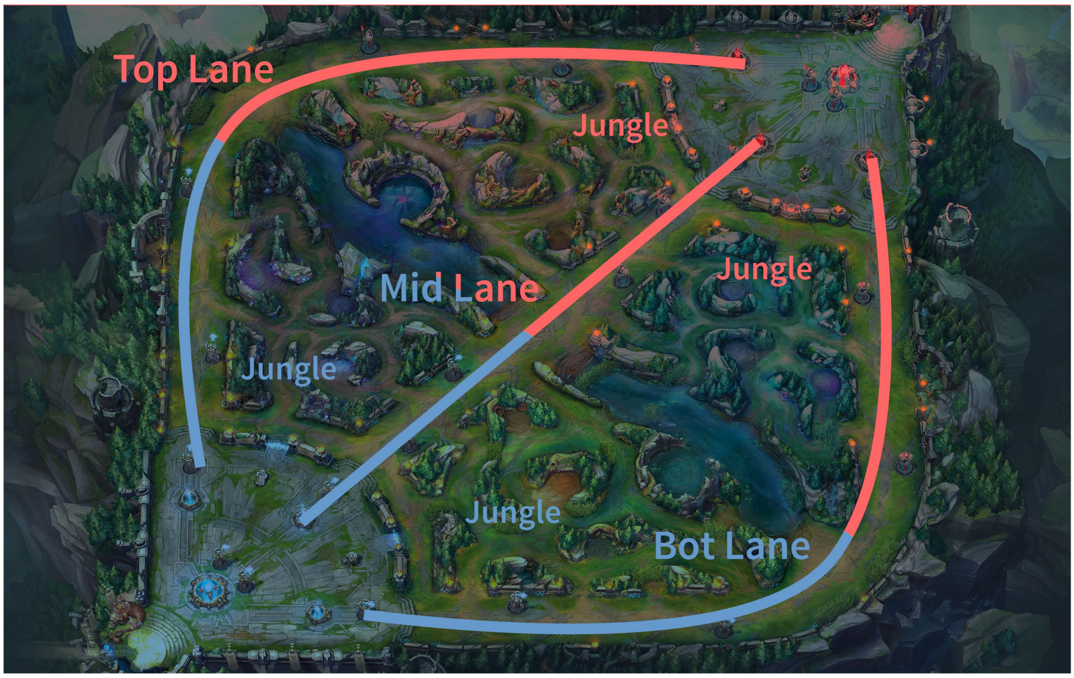

Figure 2 – LOL map, extracted from Jad Addams's article[4]

equal-sized teams, and have their assigned roles, the pick and ban phase starts.

The pick and ban phase encompasses the champion selection or prohibition of selecting. This phase ends quickly and finishes with all players having a chosen champion. A champion is a playable character that has its own set of abilities and attributes. Abilities can be used to deal damage, heal, or apply a variety of status (e.g., stun, root, knock up, etc.) to other champions, while attributes (e.g., health, health regeneration, armor, magic resist, among others) define the champions' strengths and weaknesses.

Lastly, the gameplay phase occurs on a symmetric map, represented by Figure 2, where all players, depending on their role, move to specific points in the map to confront the adversary assigned with the same position. The players can kill the opponents' minions, creatures that respawn every thirty seconds, to gather gold, also known as farming. Gold can be used to buy items in the shop that help raise your champions' attributes or give him/her new abilities. The game's goal is to conquer objectives in the map and destroy the enemy's Nexus cooperatively. These objectives can be towers of the enemy team or neutral objectives, monsters in the map that give the players particular increments in their attributes, new abilities, and gold.

After a team starts to get stronger, they can destroy the opposing team's structures until they reach the Nexus. Destroying the Nexus, the closest structure to the enemy's base, will declare the team that destroyed it the winner. After a match ends, the players can choose to play another match or leave the game. If they decide to play another match, the matchmaking system will select ten players, and the cycle repeats. In this next match,

[4] https://medium.com/@itsjadaknight/your-product-team-is-a-professional-esports-team-b43d5afa4a3

all the resources gathered from the last match are erased, meaning that every match is isolated, and the players can perform better or worse depending on what happens in the game.

2.3 Player Retention

Player retention is the task that aims to identify the key reasons to maintain an user playing. This task is highly connected to churn because the model responsible for predicting churn can have its features based on the reasons identified in this researches. As stated by (PARK et al., 2017), the key factors for player retention are achievement and social aspects.

An achievement feature can encompass many aspects, but its core is in the player gaining something, like an item, experience, success, etc. This feature can be measured by a player's action in the game or the results of a day or a match. Usually, this feature is calculated by the time spent playing and the player's performance that comprise both the time required to acquire these achievements and the evolution that came with these new resources.

Differently, the social aspects regard to all actions done in a group of people, in our case, cooperative tasks in a game. Players can have different tie strengths, depending on the time used to execute tasks with specific individuals. For instance, two close friends tend to play more together than two players that barely know each other (PIRKER et al., 2018). Aside from this individual influence, another social aspect that will be accountable in this research is the characteristics of the group or groups a player is a part of, explained in more detail in Chapter 5, where a combination of social metrics that encompasses individual and group features are proposed.

Even though these features present to be of great importance, most works tend to analyze only the players' engagement, capturing the quantity of time played by a player. Although the time used to play is important, and it is a usual metric for evaluating a game's success, it lacks the consideration of how the players play. Regarding the social metrics, few articles assess it in players' churn prediction and none using data from a MOBA game. This work solves both issues by deploying Commitment and the social metrics in a MOBA game dataset.

2.4 Commitment

Commitment is a metric proposed by (KUMMER; NIEVOLA; PARAISO, 2017a) that represents how attached a player is to a game and considers two features, time spent playing and score achieved. The time spent playing is easily obtained, but the score is not an easy task and can have many variables. In this research, an approach similar to the method proposed by (PRAKANNOPPAKUN; SINTHUPINYO, 2016) was utilized to overcome this problem. Since the article of Prakannoppakun and Sinthupinyo lacks details about the deployment of the method, experiments to improve its performance, and an evaluation procedure, this work proposes and appraises all the issues in a larger dataset of a different MOBA game.

The Commitment metric's main assumption is that if a player likes a game, he/she will play for more time and, consequently, improve his/her abilities. This idea is used to cluster the players into three groups and later to label them into three levels of commitment, low, average, and high. Since this labeling considers the player data set's values, the metric adapts to the players' behavior over time. These characteristics have been proven to improve models' performance, making it valuable to identify risk situations, such as when a game enters the niche stage, and predict churn (KUMMER; NIEVOLA; PARAISO, 2018).

2.5 Graph

A graph can be described as $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$. $\mathbf{V}$ being the vertex set and $\mathbf{E}$ the edge set. Let $\mathbf{V} = \{v_1, \cdot\cdot\cdot, v_i, \cdot\cdot\cdot, v_n\}$ where N is the number of vertex and $\mathbf{e}_{ij} = (\mathbf{v}_i, \mathbf{v}_j) \in \mathbf{E}$ (OU et al., 2016). In this work a directed graph was used, meaning the edges contain a direction, for instance, $\mathbf{e}_{ij}$ and $\mathbf{e}_{ji}$ are not the same. Another difference from the standard graph definition presented is the usage of weights $\mathbf{W} \in \mathbf{R}$, where every edge has a weight $\mathbf{W}(\mathbf{e})$, $\mathbf{e} \in \mathbf{E}$.

A graph can store the interactions between players, using vertex and edges to keep the information of the players who played together and the amount of interaction between them as the weights of the edges, as illustrated in Figure 3. With a directed weighted graph constructed from the players' daily data usage, a few metrics can be extracted to obtain the features needed.

In this work, five metrics were extracted from the graph, the churn influence, number of neighbors, average neighbors' degree, transitivity, and clustering coefficient. For the churn influence, the node's edges' weights are used with a positive value if the neighbor did not churn (Players 3 and 4 in the example) or negative otherwise (Player 1), as presented in Figure 4 if Player 2 is being analyzed. The Player 2 node will be used in all examples. Using this example, the churn influence can be calculated as 0.2. The number of neighbors is represented by the node degree, which is the number of connections a node is connected to, for instance, the number of edges that contain that node $\mathbf{E} \supset \mathbf{v}_i$. It is important to notice that the number of neighbors only consider the out degree, meaning the vertex that start at the node being analyzed. An example can be seen in Figure 5,

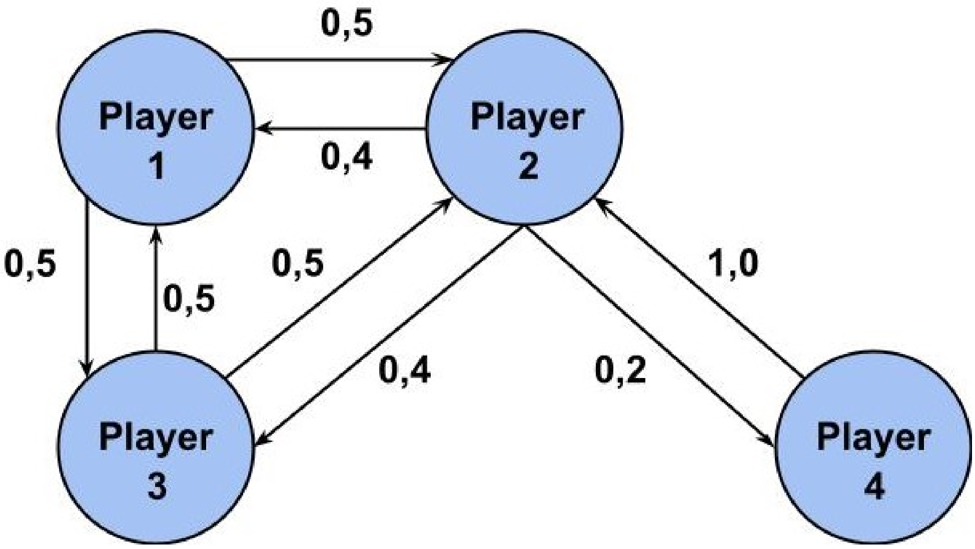

Figure 3 – Players' interactions storage in a graph

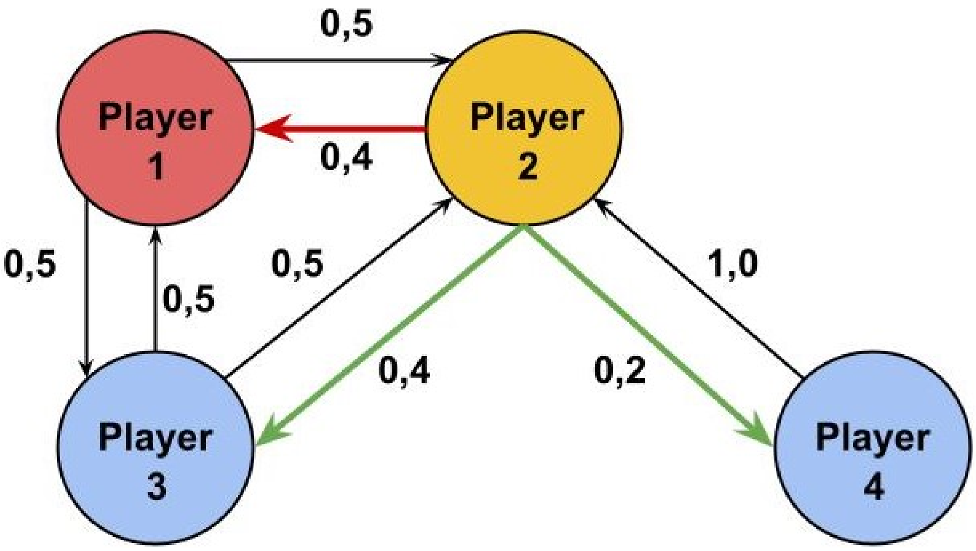

Figure 4 – Churn influence example

where the number of neighbors is equal to three.

Following the idea of node degree, the average neighbors' degree can be calculated using the degree average of all nodes connected to a specific node. In the example, illustrated in Figure 6, the metric value is 1.67.

Regarding the next metric, transitivity, first it is necessary to understand what a triangle is. Triangles are the number of groups of three nodes fully connected, demonstrated by Figure 7. The implementations and definitions of both metrics followed the ones described in the NetworkX Python library[5]. Transitivity is the number of possible triangles in a graph. In this research, the graph used is a subgraph containing only the node being evaluated and its neighbors. To calculate the number of possible triangles, the

[5] <https://networkx.org/documentation/stable/reference/index.html>

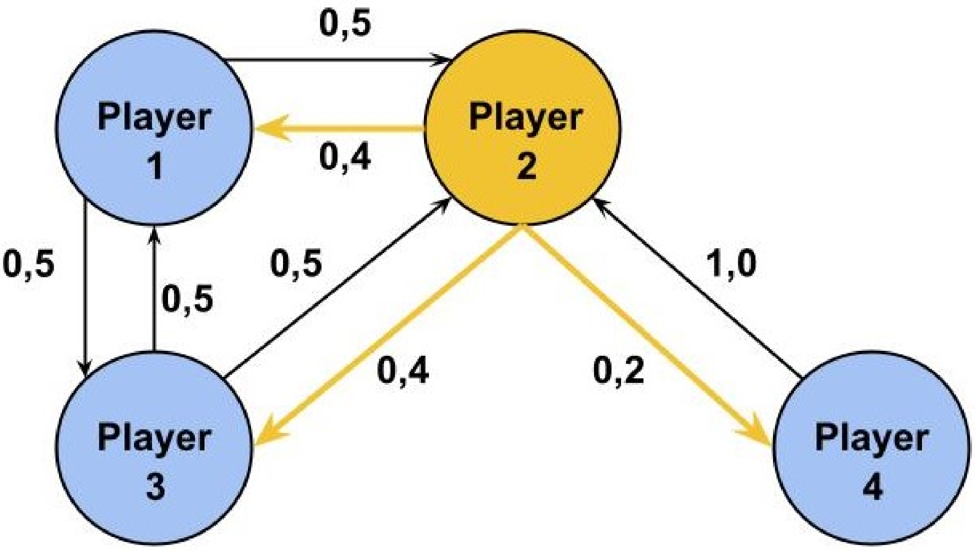

Figure 5 – Number of neighbors example

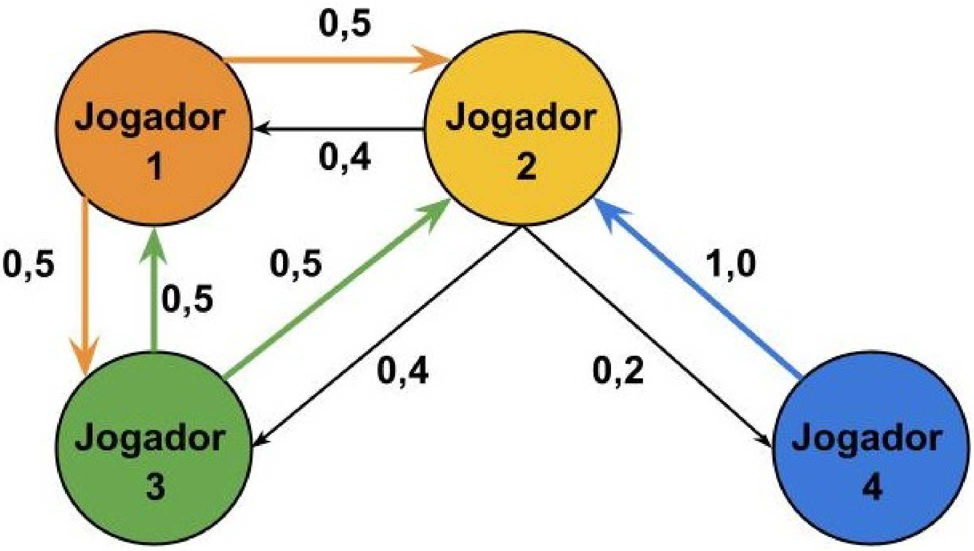

Figure 6 – Average neighbors' degree example

implementation uses Equation 2.1, where triads are two edges with a shared vertex, as presented in Figure 8. For this example, there are two triangles and ten triads. Deploying Equation 2.1, we have a transitivity of 0.6.

$$\textbf{Transitivity} = 3\frac{\#triangles}{\#triads} \tag{2.1}$$

Since the graph used is directed and weighted, the clustering coefficient is defined as the subgraph edge weights' geometric average. The calculation is demonstrated by Equation 2.2, where u is the node being analyzed, $deg^{tot}(u)$ is the sum of in-degree and out-degree of u and $deg^{\leftrightarrow}(u)$ is the reciprocal degree of u. Using the same example of the previous metrics, the geometric average is equal to 7.44 and there are six $deg^{tot}(u)$ and

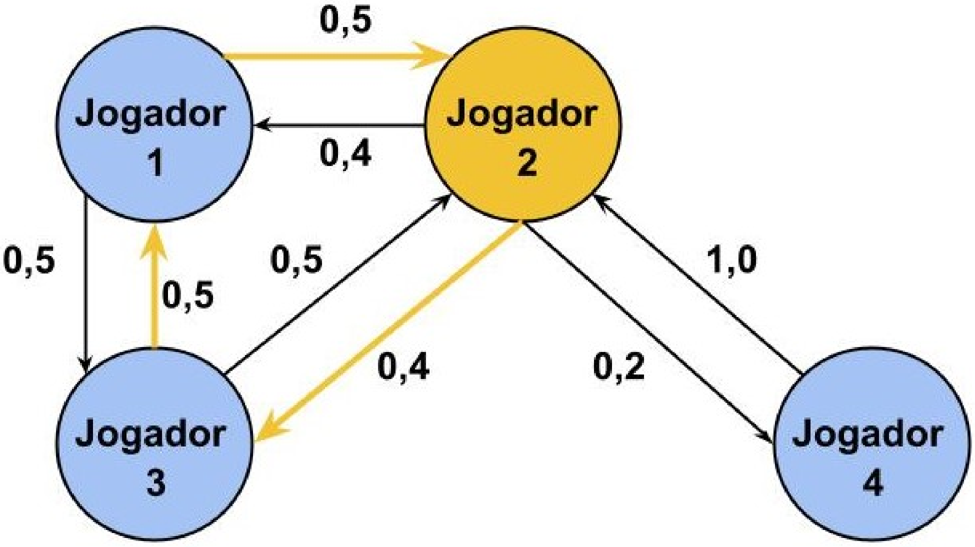

Figure 7 – Example of a triangle in a graph

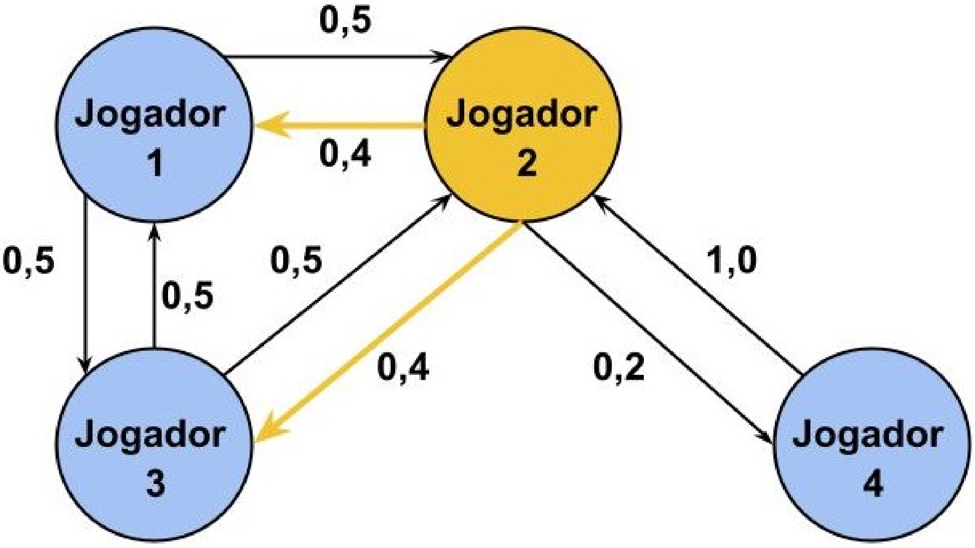

Figure 8 – Example of a triad in a graph

three $deg^{\leftrightarrow}(u)$. Using this information and Equation 2.2, we obtain a clustering coefficient of 0.15.

$$c_u = \frac{\sum_{vw}(\hat{w}_{uv}\hat{w}_{uw}\hat{w}_{vw})^{\frac{1}{3}}}{(deg^{tot}(u)(deg^{tot}(u)-1)-2deg^{\leftrightarrow}(u))*2} \tag{2.2}$$

2.6 Churn

Churn can be described as the act of a user abandoning a service. This service can be a banking account, a phone number, a game, a subscription, or any other service type. Users are the service providers' primary revenue, so predicting if a user will churn is an important task. When approaching churn in the telecommunications domain, banking, or most of the services, the client needs that service. There is a substantial barrier to acquire a service as well as to abandon it. However, when dealing with Entertainment Games that players play voluntarily, this barrier is practically nonexistent. There is no explicit warning when leaving it. This characteristic and our less invasive approach make it a challenging problem, where all the information we can extract to predict churn is located in the game usage data.

When predicting churn, there are two main approaches, binary classification, which classify a player as a churner or non-churner, and survival analysis, which considers the churn as an event and tries to predict when this event will occur, giving the producers a time window to try to retain that player.

On the one hand, the binary classification approach is similar to any Machine Learning problem of binary classification. On the other hand, statistical approaches are used in the survival analysis to find the survival probability and the hazard probability. The survival probability is the percentage that an individual survives from the time origin to a specified future point in time, meaning the event's non-occurrence. Contrary, the hazard probability is the percentage that an event will occur. In our problem, this event can be the user playing or not playing another match, and the probabilities are calculated using past events. This research focuses in the binary classification, leaving the application of the survival analysis approach to future works. The choice for the binary classification was made because more works utilize it, meaning that implementations and comparisons could be done more easily.

Aside from the approaches used, it is interesting to notice that the datasets are usually unbalanced, meaning that techniques for unbalanced data can be used, and evaluation metrics that can address this issue, such as the F1-Score, need to be used. Lastly, the players' behavior patterns can have different periods, which means that different sizes of time windows are necessary to encompass its comprehension.

3 State of the Art

The related works are split into three sections to better separate each issue addressed by this research. First, the related works regarding churn labeling are detailed in Section 3.1, explaining how they labeled their dataset, the main labeling approaches used, the issues encountered, and how they can be solved. Next, Section 3.2 shows the articles concerning players' performance evaluation, highlighting their inapplicability in MOBA games, proposing improvements to the only method that can be used for MOBA games, and evaluating the changes using a novel evaluation method. Later, all works regarding churn prediction in games found in the systematic mapping are described. This analysis revealed that only a few studies utilized social aspects in their features, only one work used a MOBA dataset, and none combined both. Finally, all the gaps encountered are summarized in Section 3.4.

3.1 Churn Labeling

Numerous works predict churn. Some utilize only achievements aspects (HADIJI et al., 2014; BERTENS; GUITART; PERIÁÑEZ, 2017; KUMMER; NIEVOLA; PARAISO, 2018; MILOŠEVIĆ; ŽIVIĆ; ANDJELKOVIĆ, 2017; PERIÁÑEZ et al., 2016; TSYMBALOV, 2016), others use social (BACKIEL et al., 2015; BARAS; RONEN; YOM-TOV, 2014; BORBORA et al., 2011; BORBORA, 2015; DASGUPTA et al., 2008; DROFTINA; ŠTULAR; KOŠIR, 2015; KAWALE; PAL; SRIVASTAVA, 2009; LIU et al., 2019; ÓSKARSDÓTTIR et al., 2016; PHADKE et al., 2013; SARAVANAN; RAAJAA, 2012), some see the problem as time-series (BORBORA; SRIVASTAVA, 2012; CASTRO; TSUZUKI, 2015; KIM et al., 2017; KRISTENSEN; BURELLI, 2019; RUNGE et al., 2014; TAMASSIA et al., 2016; YANG et al., 2019), another utilize Natural Language Processing (KILIMCI; YÖRÜK; AKYOKUS, 2020). Regarding the game domain, they use the data from different game genres such as Multiplayer Online Battle Arena (MOBA) and Massively Multiplayer Online Role-Playing Game (MMORPG), from different platforms, like desktop and mobile, and target diverse types of players. All of them provide useful information about algorithms and techniques to classify the players as churners or non-churners. Still, they do not focus on the labeling process, using a static definition of who should be considered a churner for the whole dataset. This issue raises some concerns about the reliability of the performances obtained because the definition of churn, and, consequently, the labels, could change over time. Still, it is assumed that they remain the same.

Focusing on labeling, two works studied their behavior using different techniques. Approaching with an economic view, (CLEMENTE-CÍSCAR; MATÍAS; GINER-BOSCH,

2014) utilizes the idea of loyal customers proposed by (BUCKINX; POEL, 2005) to calculate the usefulness of the churn definitions by analyzing the economic loss of the churn preventing campaigns. Even though it is an exciting approach and directly impacts the game producers' profit, the game's financial data is necessary, which is generally not available, invalidating its usage in most situations.

The work of Rothmeier and colleagues (ROTHMEIER et al., 2020) provided insights about the different approaches used to label the players. They explained and tested four techniques by comparing the final result obtained from various algorithms in the churn prediction task. The four approaches are divided into two categories, the ones that utilize the players' log history, namely, Naive and Sliding Window, and the other two that use the idea of disengagement, where a significant reduction in playtime characterizes disengagement, later implicating in churn (XIE et al., 2015). The disengagement-based approaches have an exciting concept, but since they utilize time spent playing to label the players, data usually used as a feature in the classification algorithm, we choose to exclude these approaches because they could implicate bias regarding the classification. The other reason to exclude is the fact that in the experiments performed by (ROTHMEIER et al., 2020) it was shown that the Naive performed very well in all the algorithms tested and excelled in two, removing the necessity of studying approaches that can lead to a bias in the classification and acquired a similar result.

Even though they tested different churn labeling approaches, the experiments focused on the classifiers' results, not the labeling itself. The problem of evaluating the labeling process resides in the nonexistence of true labels, resulting in the lack of values to be used for comparison. True labels only exist in the cases of games that finished their usage life cycles (KUMMER; NIEVOLA; PARAISO, 2017b), where the notion of churn has a final form for each player. Note that as each game can have distinct players' behaviors, transferring learning from one game to another without a possible bias is impossible. Therefore, true labels cannot be considered since they only exist when the churn prediction is not a need anymore, as the game operation was finished. This means that churn labeling approaches must encompass the ability to adapt to a dynamic churn behavior during a game life cycle, which is firmly attached to the players' behaviors that change overtime (ZHU; LI; ZHAO, 2010; COOK, 2007). This fact highlights that when a game adopts a static definition of churn, it ignores the changes in players' behavior and keeps predicting churn based on a possible no coherent concept according to current data. An entailed problem is that a good accuracy of a classifier can hide this situation, as what is predicted with high confidence could not be linked to the actual notion of churn, leading to poor churn management.

A way to evaluate the labels could be done by creating two perspectives to analyze and compare, accomplished by separating the players' log data into two parts and deploying a chosen labeling approach in both parts, which results in two definitions of churn. It is essential to notice that the final result of this first step is not to acquire labels for each part but to define what is churn (e.g., after n consecutive days a player did not enter the game, he/she is considered a churner). After splitting the data and acquiring a churn definition, it is possible to apply both definitions in the second part, which contains the most recent data, and compare the obtained labels. Considering the nature of the players' behavior, the labels are believed to change in a given moment because, as observed and described by (ZHU; LI; ZHAO, 2010; COOK, 2007), the players' life cycle travels a linear path represented by different motivational stages. These behavioral dynamics cause a change in churn definition, which can be measured by disagreements between the resultant labels in the second step because the analysis considers both definition perspectives from past and current log data.

There are three approaches used in related works to label players as churners or non-churners, each one will be explained, and their weaknesses will be discussed below (Subsections 3.1.1, 3.1.2, and 3.1.3).

3.1.1 Fixed Value Approach

One of the most common ways to label the players as churners or non-churners is to define a number of days or a Fixed Value (FV). If in the most recent data, a player has not played consecutively for this amount of days, named its last absence, he/she is considered a churner, as demonstrated by Equation (3.1). This value can be defined empirically (ROTHENBUEHLER et al., 2015), but some authors utilized the players' history to achieve an FV that encompasses the behaviors of the player base majority, as performed by (PERIÁÑEZ et al., 2016), (RUNGE et al., 2014), and (YANG et al., 2019).

$$Label = \begin{cases} \text{Churner}, & \text{if } LastAbsence > FV \\ \text{Non-Churner}, & \text{otherwise} \end{cases} \tag{3.1}$$

Following the same idea of a data-driven approach, Equation (3.2) represents how the FV calculations are performed in related works, where an absence with a return is the number of consecutive days not played followed by a day played, the n represents the number of absences with a return and i the ith absence. As an example, in Figure 9, there are three distinct players and their respective play histories. Acknowledging the definition of an absence with a return, three absences can be identified in the example, 1, 1, and 2 days of absence with return. The player with Identification (ID) 1 has zero absences, followed by ID 2 with an absence of 1, and, lastly, ID 3 with two absences, one of one day and another of two days. Using Equation (3.2), the FV of one is obtained by averaging the three values found. With the FV of one and utilizing Equation (3.1), the

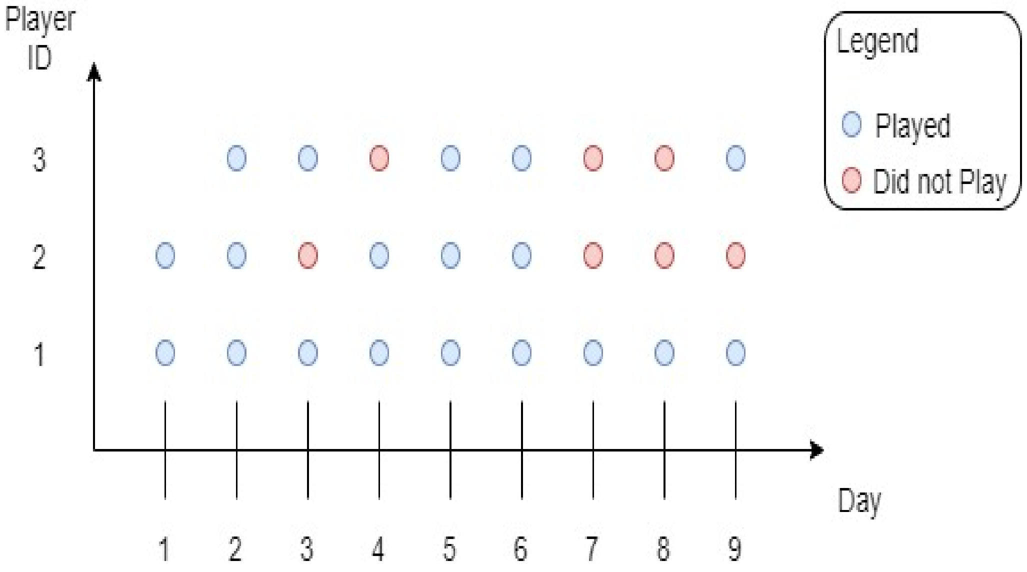

Figure 9 – Fixed Value approach example

players with the IDs 1 and 3 are not considered churners, because they have a last absence of zero, and the player 2, with its last absence of three days, is labeled as churner. If defined empirically, the FV used could not represent the player base, which was solved by using a data-driven approach. Still, since the players can have different behaviors, like playing only on the weekends or playing only on the week's days, the value could not be a reliable representation for each player.

$$FV = \frac{\sum_{i=0}^{n} AbsenceWithReturn_i}{n} \tag{3.2}$$

3.1.2 Naive Approach

The Naive approach consists of dividing the dataset into two roughly equal parts and verifying if the players were present in both parts, non-churner, only on the first, churner, or second, beginner. As stated by (ROTHMEIER et al., 2020), the addition of the third class provides the intention of improving class balance, but the data splitting technique has several drawbacks. It can be challenging to choose a specific timestamp; the chosen one can bring lots of data from some players but almost none from another; the beginners' exclusion from the non-churners class can conceal particular insights; it can lose important information about behavior changes in the second split. A usage example can be seen in the work of (DRACHEN et al., 2016), where the Naive approach was utilized on the first and second weeks of the players' log history to label the players.

A running example is exemplified by Figure 10, where the same dataset used in the previous example is separated into two roughly equal parts. Since all players are present in the first and second half, we can label them as non-churners.

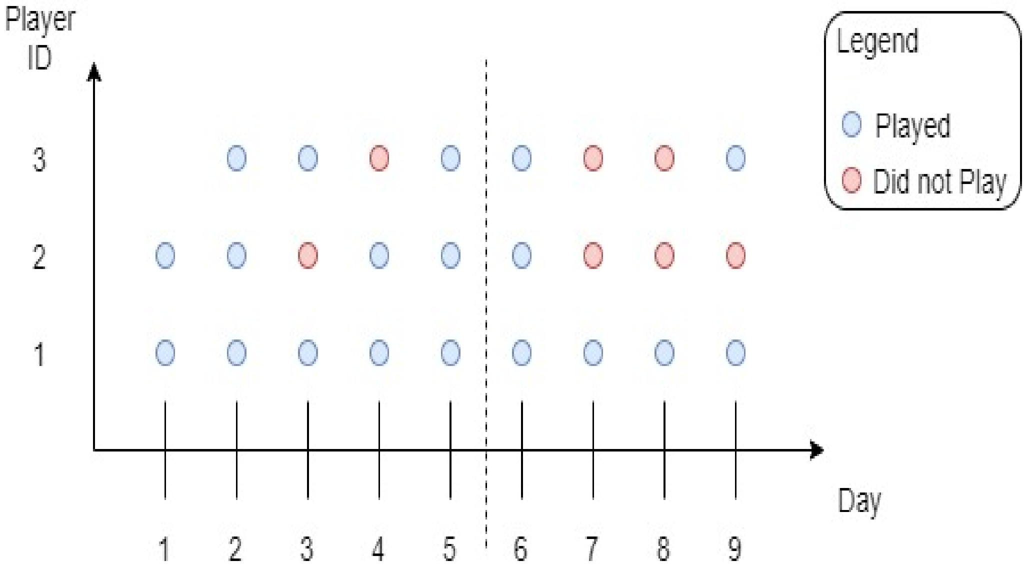

Figure 10 – Naive approach example

Aside from the drawbacks mentioned earlier, the Naive approach can take, in the worst case, double the size of the dataset to identify churners because it would take the same size of the dataset as the number of days not played. Another problem can be identified if an evaluation like the one proposed in Section 3.1 is executed because differently from the Fixed Value approach where a value that represents the player base is calculated, meaning that the definition of churn could change, for the Naive, the rule is always the same. This characteristic restricts a comparison between windows and invalidates the identification of a change in the churn definition. The inability to identify this change excludes the possibility of evaluating the approach's correctness, which is a considerable drawback.

3.1.3 Sliding Window Approach

Like the Naive, the Sliding Window (SW) approach follows the same rules, but the difference occurs when separating the dataset into two parts. Diverging from the Naive, the splitting does not need to divide the dataset equally, enabling fine-tuning to specific games and solving the issue of taking too long to identify churners. In Figure 11 it is possible to adjust the dashed line to the left or the right, increasing or decreasing the size of the windows, represented by M and N. Compared to the last example, we advanced one day and this changed the label of Player 2 to churn because it is absent in the second part of the split. Although fine-tuning is a great addition, it is challenging to choose the best sizes for M and N. Since the SW approach follows the same rules of the Naive, accounting for the players' presence in two separate windows, it also suffers from the same problem of not capturing the churn definition changes.

Bearing in mind all the presented approaches, there are two strategies to label the players, one that considers a fixed value, the average of absences, and another that uses

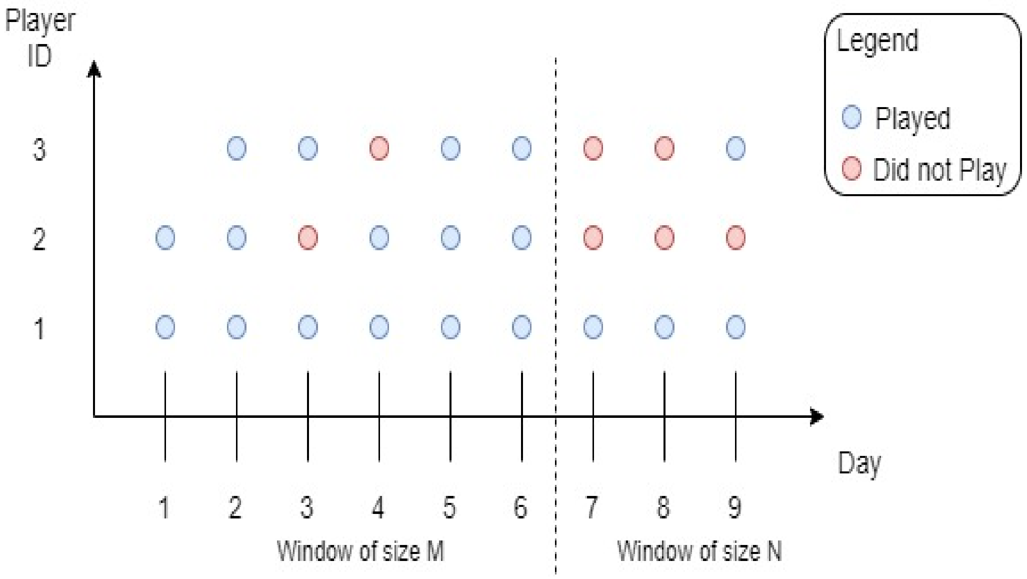

Figure 11 – Sliding Window approach example

the players' presence. These strategies can also encompass individual analysis or not and allow the identification of changes in the definition of churn. Table 1 summarizes such aspects for each approach. Note that none cope with both individualized analysis and the identification of churn definition changes, highlighting the importance of an approach that covers them. Independent of the separation used, this type of analysis was presented to understand the approaches' relations, emphasizing their strengths and weaknesses.

Table 1 – Labelling approaches' characteristics

Labelling Approach	**Labelling Strategy**	**Individualized Analysis**	**Allows Churn Definition Changes**
Fixed Value	Absence Average		X
Naive	Presence	X	
Sliding Window	Presence	X	

3.2 Players' Performance Evaluation

The first attempt to give each player a score was done to chess players, called the Elo System, (ELO, 1978), that uses the BradleyTerry model for pairwise comparison (BRADLEY; TERRY, 1952) and the mean value of the players' performance in a match. Later, the Glicko System was invented to address the uncertainty in the players' current skill (GLICKMAN, 1995). The main problem with both approaches was the inability to use them in a multiplayer context. With this issue in mind, the TrueSkill algorithm was developed (HERBRICH; MINKA; GRAEPEL, 2007). It represents the players' skill as a Gaussian-distributed variable and accepts ties. Numerous extensions were proposed to improve the algorithm, but none incorporate the match statistical results, only the outcome. Since a MOBA game match is a team effort and can be won by a small or large margin, a specific system to evaluate the players' performance or contribution to the team's victory is needed.

In his article, (MAYMIN, 2020) utilizes several techniques and algorithms to generate high-frequency data of ranked League of Legends (LOL) matches, calibrates an in-game win probability model, and introduces dozens of novel metrics, aiming to analyze LOL eSports matches, the competitive scenario. Although he introduces several advanced metrics, there is no calculation of a score for each player.

The work of (HODGE et al., 2019) utilizes the draft, group of champions selected, as information to predict which team will win the match in a competitive environment. With the desire to create an Artificial Intelligent Agent (AIA) that can dynamically change the difficulty based on the players' skill level, evaluating the player was needed. To achieve a score for the player, (SILVA; SILVA; CHAIMOWICZ, 2017) used statistics of the match (the champion's level, death count, and the number of towers destroyed) as an indicative. To summarize, this approach utilizes the character's progression and the completion of the objectives to calculate a player's score. Following the same idea, (CAPLAR; SUZNJEVIC; MATIJASEVIC, 2013) used similar metrics but as averages of a player's whole history to analyze its performance. Even though these are significant indicators, since each character can have a specific team function, they can be misinterpreted. For example, a character chosen to perform as support can have more deaths than a character responsible for dealing damage because the support is supposed to protect the damage dealer.

Wishing to improve the actual rating system in the game and acknowledging the characters' roles, (SUZNJEVIC; MATIJASEVIC; KONFIC, 2015) also used statistics from the match but included the champion used and the roles they can exert. To address the issue that each champion is different and performs another function, they utilized the players' statistics with weights that are used to raise or reduce the importance of that metric to a specific champion. The issue in their work is that the values are limited to 0, 0.5, and 1, and that they are defined empirically.

Following the same idea of using the match outcome statistics and considering each champion individually, (PRAKANNOPPAKUN; SINTHUPINYO, 2016) used the same approach with weights but deploying it as an Artificial Neural Network (ANN). The inputs are separated by champions, where zeros in all values represent a champion not present in the match. If a champion is in the match, its performance statistics (e.g., kills, deaths, assists, amount of damage and healing, etc.) are placed in the correct position of the vector, and the connection weights of the neurons will learn the most important attributes for each character based on the outcome of the match, if the team won or lost.

To accomplish a comparison value, each player is placed in the input vector's correct position, depending on the champion chosen, and the feedforward algorithm is deployed. If the team won, the value resultant should be greater than the value obtained from the feedforward of the players encompassed in the losing team. In the case of an error, the weights are fixed.

Even though this approach is ingenious, the article lacks some information and experiments regarding the method's architecture and how much bigger a value obtained from the winning team should be compared to the losing team. Other doubts that surge is in the ANN architecture, where different parameters could improve the results and the applicability in other game datasets. Lastly, the comparison present in the article calculates the difference between the score obtained and the players' ranking, calculated by the authors, which could not represent the actual rating or the player's real skill. To address these issues, some modifications were made in their method, and experiments were performed to answer the doubts raised in a larger dataset of a different game.

3.3 Churn Prediction

A systematic mapping, more details in Section 4.1, was made to obtain all related works regarding churn prediction in games with social aspects, graph analysis, and social network analysis. These topics were separated to acquire researches in one of this study's focus, churn prediction in games using social aspects like features, and the two techniques, graph and social network analysis, that could be used to calculate this social metric. It is essential to notice that, even though works from other domains and researches that do not consider social aspects were found, they were not the main focus of the systematic mapping. Table 2 describes the works found regarding predicting churn and their most important aspects related to this research's objectives. Since only works that encompass churn prediction in games are considered in this section, from a total of 73 articles, only 16 appear on the presented table. The rest contains techniques or churn prediction methods that could be useful for this work, but, after the definition of this research's scope, were disregarded. This decision occurred due to methods or techniques that could not be replicated or metrics that do not encompass the information required by the proposed method. The applicability could or could not occur because of the game context's specific characteristics, available data, voluntary usage, and changes in the players' behavior.

Even though MOBA is one of the most played game genres, to the best of our knowledge, no research studies MOBA game's social aspects in the churn prediction task, but there is one work that predicts churn in this genre without social interactions. In (CASTRO; TSUZUKI, 2015), the authors applied the login records of three different games, RF Online, a hardcore MMORPG, APB: Reloaded, a hardcore third-person-shooter and

Article	**Other Genres Dataset**	**MOBA Game Dataset**	**Social Aspects**
(BERTENS; GUITART; PERIÁÑEZ, 2017)	X		
(BORBORA et al., 2011)	X		X
(BORBORA; SRIVASTAVA, 2012)	X		
(BORBORA, 2015)	X		X
(CASTRO; TSUZUKI, 2015)		X	
(HADIJI et al., 2014)	X		
(KAWALE; PAL; SRIVASTAVA, 2009)	X		X
(KIM et al., 2017)	X		
(KRISTENSEN; BURELLI, 2019)	X		
(KUMMER; NIEVOLA; PARAISO, 2018)	X		
(MILOŠEVIĆ; ŽIVIĆ; ANDJELKOVIĆ, 2017)	X		
(PERIÁÑEZ et al., 2016)	X		
(RUNGE et al., 2014)	X		
(TAMASSIA et al., 2016)	X		
(TSYMBALOV, 2016)	X		
(YANG et al., 2019)	X		

Table 2 – Articles regarding churn prediction in games

HMM: Heavy Metal Machines, a hardcore MOBA, into a K-Nearest Neighbours algorithm with different features. The features tested were extracted based on recency, frequency, monetary approach, and login records in the time, frequency, and time-frequency plane domain. All features were extracted using the same data, login record, but using different techniques. The objective was to evaluate the same data in various dimensions. The results showed that a temporal analysis could be used in the churn prediction problem, even considering just login data.

Although the systematic mapping focused on finding methods that predict churn considering social aspects, only three works were found. In their work, (KAWALE; PAL; SRIVASTAVA, 2009) proposes a modified diffusion model that propagates positive, non-churner, or negative, churner, influence to its neighbors with a weight depending on the quantity of interaction they had. The difference from the standard diffusion model is the quantity of influence a node propagates to its neighbors. Generally, the influence degrades when propagated. They maintained the same amount to all the neighbors. Their results showed that two factors improved the model performance, the modified diffusion model and social influences. Compared to a simple diffusion model, the proposed method considered only engagement features, defined in their work as the time spent playing.

To compare theory and data-driven approaches, (BORBORA et al., 2011) used ensembles and features regarding achievement and social interactions. For the social features, the theory-driven approach used the rate of group interactions. In the data-driven approach, the number of actions, experience gained, and activities done with churners.

They concluded that the data-driven approach performed better, but experts can interpret the theory-driven approach more descriptively.

In his thesis, (BORBORA, 2015) presented four types of research regarding churn prediction in MMORPGs, user behavior modeling approach for churn prediction, community churn prediction, the impact of achievement and socialization factors on individual churn, and social contagion. The user behavior modeling approach of (BORBORA; SRIVASTAVA, 2012) analyses the players' behaviors before the churn happens, a lifecycle-based approach, defining distinct profiles used in the classification scheme by comparing the distance of a new player's features to the ones that have already been labeled. Because our research's primary focus is to predict individual churn, the community churn will not be explained, but more details can be found in his thesis.

To study the impact of achievement and socialization factors, the players were separated into two groups, loners and socializers, and applied to different models containing achievement and social activities. The social activities utilized include the number of group sessions, their length, and node centrality measures. It was concluded that social features increase the performance of the model to predict churn.

Lastly, social contagion is studied to answer two research questions. The first research question is, when a player churns, what will be his/her impact on the activity of its neighbors, based on its characteristics and relationship with them. For the second research question, the objective was to analyze the same impact but based on the affected player's remaining neighbors' activities. The results showed that homophily-based features are not discriminant in predicting dyadic influence. Still, the churner's level of expertise and centrality are determinants of the impact on its neighbor's activities. Answering the second research question, it was noted that another determinant feature is the number of remaining neighbors and their tie strength.

Aside from the social aspect, 12 works that considered other features, such as engagement or achievement, were found and will be briefly explained in groups of similar studies.

First, (HADIJI et al., 2014) states the methodology to realize the churn prediction task, beginning from the churn period's definition and its features. Following a similar method, (YANG et al., 2019) proposed the definition of the churn period based on data, considering a value that encompasses 95% of the players in the dataset and incremented the features with the use of entropy to capture the variance of engagement over time and compare it with the other players using cross-entropy.

Focusing on high-value players (also known as "whales"), (RUNGE et al., 2014) predicted churn using a Hidden Markov Model (HMM). Also aiming into whales, players that spent the largest quantities of money into the game, (PERIÁÑEZ et al., 2016) applies a different approach, ensemble of survival trees. In their approach, they use survival analysis to predict when a player will churn. Similarly, (BERTENS; GUITART; PERIÁÑEZ, 2017) use the same algorithm but increments the method to predict at which level the player will be when he/she stops playing.

With the focus on the time component of the data, (TAMASSIA et al., 2016) considered the data as a time series and applied it to an HMM, while (KRISTENSEN; BURELLI, 2019) combined sequential and aggregated data, creating different models with temporal acknowledge, using LSTM and Random Forest algorithms.

Aiming at a different group of players, beginners, (KIM et al., 2017) applies the login data into an LSTM algorithm to predict if a player will continue playing for a predetermined period. Alike, (MILOŠEVIĆ; ŽIVIĆ; ANDJELKOVIĆ, 2017) predicts churn for new players but focusing on sending personalized notifications to retain the player based on a clustering made of their way of playing the game.

Proposing a cohort-based meta-metric, (TSYMBALOV, 2016) uses an ensemble of classifiers, each specialized to classify players absent for a specified quantity of days, in the churn prediction task.

Lastly, as already explained in Section 2.4, (KUMMER; NIEVOLA; PARAISO, 2018) applied Commitment to the churn prediction task. The conclusion was that it outperformed the other methods proposed on a data mining competition in the experiments. Different from them, the results obtained were similar in the two datasets. This information implicated that the Commitment could be a generic approach that is independent of the business model.

To summarize, there are numerous works about the churn prediction task, and even though the social aspects were proven to be important in many articles, it is not well studied in the game domain. There were only three works that applied this aspect, none in the MOBA genre.

3.4 Conclusion

This work found two gaps in the churn prediction task in games, one regarding the labeling process and another in the usage of social metrics in MOBA games. Considering the MOBA context, the non-existence of a replicable method capable of returning a value representing a players' performance in a match was discovered. All the gaps encountered are dealt in this research, and each solution and results will be explained in the next chapters.

4 Methodology

This chapter is dedicated to present the methodological approach chosen for this research. It is separated into exploratory research, planning, development, and evaluation, as illustrated by Figure 12.

4.1 Exploratory Research

The exploratory research phase included two steps, preliminary research followed by a systematic mapping. The preliminary research was made to find the keywords necessary to define the systematic mapping queries. This initial exploration of methods that predict churn in games with social aspects resulted in the following search queries:

Q1 churn AND (prediction OR predicting) AND (MOBA OR "multiplayer online battle arena") AND social

Q2 social AND (network OR networking) AND (influence OR analysis OR dynamic OR interaction) AND (MOBA OR "Multiplayer Online Battle Arena")

Q3 churn AND (prediction OR predicting) AND (game OR games OR gaming) AND social

Q4 "social network analysis" AND (graph OR graphs) AND "machine learning"

These queries were then used to acquire all English articles, independent of date, found on five scientific databases, ACM, IEEE, ScienceDirect, Scopus, and Springer. As showed in Table 3, 6,882 articles were found using the search strings, and another 42 were found in the preliminary research, totalizing 6,924 articles. Some queries, such as Q4, could

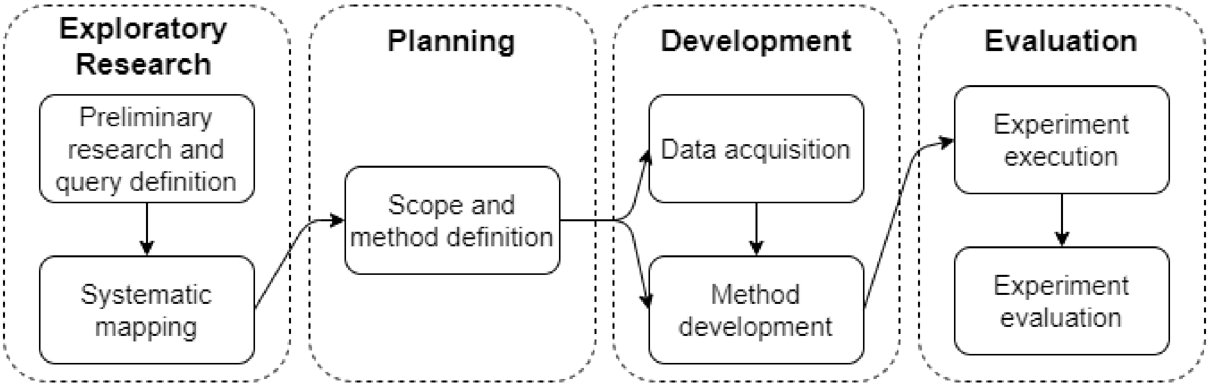

Figure 12 – Research structure

Table 3 – Systematic mapping results

Source	ACM	IEEE	ScienceDirect	Scopus	Springer	Total
Preliminary Research	-	-	-	-	-	42
Q1	12	0	5	0	1	18
Q2	47	3	71	15	154	290
Q3	254	9	554	18	1,354	2,189
Q4	952	34	931	63	2,405	4,385
Total	1,265	46	1,561	96	3,914	**6,924**

Table 4 – Accepted articles

Source	Accepted
Preliminary Research	20
Q1	6
Q2	11
Q3	23
Q4	10
Snowballing	3
Total	**73**

be removed or changed to be more restrictive, but they were maintained since the general context was being pursued.

Each article had its relevancy evaluated depending on the reading of its title and abstract. If it had any information that could be used to accomplish this research's objective, it was considered relevant. In this step, after removing duplicates, 109 articles were separated, and another three were added by snowballing the papers that had similar objectives with this work, summing 112 articles to do the full reading.

After reading all 112 articles and writing summaries for each one, 39 were removed due to the lack of relevant content, considering the research's objective. The remaining 73 relevant articles, separated as illustrated by Table 4, were used to define this research's scope and method.

4.2 Planning

Studying the conclusions stated in the articles selected, gaps, State of the Art (SotA) methods, and tendencies could be found. With the summaries' information in mind, the motivation, objective, and limitations of this works became more evident, so each

Authors	Churn Labeling	Player Evaluation
(BRADLEY; TERRY, 1952)		X
(BUCKINX; POEL, 2005)	X	
(CAPLAR; SUZNJEVIC; MATIJASEVIC, 2013)		X
(CLEMENTE-CÍSCAR; MATÍAS; GINER-BOSCH, 2014)	X	
(ELO, 1978)		X
(FERGUSON et al., 2020)		X
(GLICKMAN, 1995)		X
(HERBRICH; MINKA; GRAEPEL, 2007)		X
(HODGE et al., 2019)		X
(KILIMCI; YÖRÜK; AKYOKUS, 2020)	X	
(MAYMIN, 2020)		X
(ROTHMEIER et al., 2020)	X	
(SILVA; SILVA; CHAIMOWICZ, 2017)		X
(SUZNJEVIC; MATIJASEVIC; KONFIC, 2015)		X
(XIE et al., 2015)	X	

Table 5 – Articles found in the new searches

aspect of this research could be well defined.

Since the systematic mapping focused on predicting churn using social aspects, two more searches were made to acquire references and SotA methods for the other two tasks. Using keywords such as “churn”, “prediction”, and “labeling”, and looking into citations and references of the articles found in the systematic mapping, another five articles were encountered. Regarding the player evaluation, keywords like “player”, “evaluation”, “performance”, and “MOBA”, were used to search methods that could evaluate players’ performance in MOBA games or another context but could be applicable to the MOBA context. Another ten articles were found using the keywords listed and searching into the article’s references and citations. The articles encountered can be seen in Table 5.

Using the information acquired from the searches made in the scientific databases, the definition of the research’s objectives, questions, hypotheses, and scope was concluded, and the method was planned to solve the issues encountered. The method was separated into three parts, the automatic labeler, the calculation of the metrics, and the classifier’s deployment to predict churn. Each part is better explained in Chapter 5.

4.3 Development

Even though the data extraction is not a part of the method, it is necessary to perform the experiments. A script, available at GitHub[1], that utilizes the League Of Legends (LOL) developer’s Application Programming Interface (API) to extract all

[1] <https://github.com/reylle/LOL-Data-Extraction>

Table 6 – Characteristics of the dataset

Map	# Matches
Summoners' Rift	170,148
Howling Abyss	10,320
Total	180,468

information needed was developed to acquire the required data. It controls the number of packages sent regarding the limitation imposed by the API and verifies the response in case of errors, acting accordingly. The behavior of this script is described below.

To start searching for the players' data, the nickname used by a player, called Summoner's Name, is needed due to the way the API was developed. This first name used was acquired randomly using a website[2] that contains a leader-board of players depending on their rank. A page was randomly chosen, and a player from that page was selected. From this initial player name, players that played with him in other matches are stored and used in the next loops of the algorithm until the wanted number of players is recorded.

Considering the API limits, two datasets needed to be extracted, one containing all the matches and their resulting stats from several players, and another containing only the login history of players. The first one was used in the experiments for the players' performance evaluation and the churn prediction training and testing. The second, which contains fewer data and, consequently, can comprise a bigger period, was used in the automatic labeler's experiments. Since the automatic labeler could be suitable to label other game genres, another dataset was prepared to be included in the experiments. More information about all the datasets extraction and data included are detailed below.

The first dataset, encompassing the matches' resulting statistics, was extracted and prepared by the author. First, the extraction was performed using the LOL developers' API. It contains information regarding the players' identification, the matches they played, and the match outcome statistics. After extracting, the data was organized and recorded in a relational database to facilitate access. It contains 180,468 matches, 170,148 in the Summoner's Rift map, the primary and most played map in LOL, and 10,320 in the Howling Abyss map, containing only one lane, the champions are chosen randomly, and the attributes and items have some minor changes. To better illustrate, Table 6 shows these amounts. The map diversity is essential to verify if the method can converge independently of the game mode and map. Present in the 180,468 matches, there is a total of 584,438 unique players.

Among the 111 values returned from the API for individual match outcome statistics,

[2] <https://br.op.gg/>

Table 7 – Performance attributes chosen

Attribute Name	Description
game_duration	Duration of the game in seconds
champ_level	Biggest champion level obtained
kills	Amount of enemy players killed
deaths	Amount of deaths
assists	Quantity of attendance in enemy players deaths
longest_time_spent_living	Biggest amount of seconds that the player remained alive
farm	Sum of the minions killed, in lane and in the jungle
gold_earned	Total of gold earned in the match
gold_per_min	All the gold earned divided by the duration of the game
xp_per_min	All the experience earned divided by the duration of the game
physical_damage_dealt_to_champions	Amount of physical damage dealt to enemy champions
magic_damage_dealt_to_champions	Amount of magic damage dealt to enemy champions
true_damage_dealt_to_champions	Amount of true damage dealt to enemy champions
damage_dealt_to_towers	Quantity of damage dealt to enemy towers
damage_dealt_to_objectives	Quantity of damage dealt to objectives
physical_damage_dealt	Amount of physical damage dealt to any target
magic_damage_dealt	Amount of magic damage dealt to any target
true_damage_dealt	Amount of true damage dealt to any target
physical_damage_taken	Quantity of physical damage taken from any source
magic_damage_taken	Quantity of magic damage taken from any source
true_damage_taken	Quantity of true damage taken from any source
damage_self_mitigated	Amount of damage mitigated by resistances
total_heal	Quantity of life healed
time_ccing_others	Duration in seconds of debilitating effect dealt to opponents
total_time_crowd_control_dealt	Duration in seconds of debilitating effect dealt to any source
vision_score	Amount of vision of the map placed or removed

26 were chosen. The rest were excluded due to deprecated values, functions unavailable at the current version of the game, or values that do not present any information about the player's performance. The chosen attributes and their respective definitions can be seen in Table 7.

When performing the automatic labeler experiments, the data from two games were used, League of Legends (LOL) and World of Warcraft (WOW). LOL is a MOBA game developed by Riot Games, and its goal is to destroy the enemy's Nexus, a structure located near its base. To achieve it, it is necessary to play cooperatively, conquer objectives, and win recurrent team fights. After the end of a match, a player can choose to play again. In this case, a new match will start where all the resources gathered in the previous game are forgotten, meaning that each match is isolated, and the main goal is collective among the players on the same team.

WOW, developed by Blizzard[3] is a Massive Multiplayer Online Role-Playing Game (MMORPG) where the main objective of each player is to get stronger. To conquer higher levels and better equipment, the player can defeat monsters, complete quests, socialize, and accomplish missions with other players. In WOW's world, a player can quit the game

[3] <https://www.blizzard.com>

Table 8 – Characteristics of the datasets

Game	# Players	# Months	Period
WOW	91,064	37	Jan. 2006 - Jan. 2009
LOL	2,400	23	Oct. 2018 - Sep. 2020

anytime he/she wants and return at the same point he/she stopped. These characteristics show that the two games diverge in how the games are played, the goal, cooperativeness, and other game design choices. The differences make them suitable for comparing the different churn labeling approaches because they could indicate that the churn definition changes differently among games.

The datasets containing players' history logs from the game WOW and LOL were used to calculate this churn definition changes. Table 8 presents their characteristics, considering the number of unique players, amount of months, and periods. The only information that the datasets contain are the players' IDs, unique to each player, and series of zeros, ones, and minus ones representing, respectively, days not played, days played, and days antecedent to the first game played. The minus ones are necessary to remove the bias caused by accounts not created before the first date in the dataset, which could be misinterpreted as an absence. The WOW dataset was created by (LEE et al., 2011) and modified to only encompass the information previously described. The LOL dataset was downloaded and organized by the author, utilizing the same producers' API previously mentioned, with randomly selected players. Both history logs datasets are available at <https://www.ppgia.pucpr.br/~paraiso/Projects/GameAnalytics/DataBases/PlayersLogHistory/> and the matches outcomes dataset can be requested from the author (joaofh@ppgia.pucpr.br).

4.4 Evaluation

In the evaluation, two steps were performed, the execution and evaluation of the experiments. The experiments' execution and evaluation are linked to answering the research questions, validating the hypotheses, or improving the method's performance. To facilitate the reading, the research questions and hypotheses are presented again:

- RQ1) What are the issues present on the churn labeling task in games?
- RQ2) Should the churn definition be updated?
- H1) The automatic labeler provides more reliable labels than static definitions;
- H2) The proposed labeling approach better represents the players' churn behavior;

- H3) Commitment can be used in MOBA games;
- H4) Social features improve the churn prediction classifier's performance in a MOBA game.

The first RQ was answered by analyzing related works, revealing the three main issues in churn labeling. To answer the second RQ and validate H1 and H2, an experiment was made comparing the most used labeling approach with a novel one using a new metric and method for comparing them.

Next, three experiments were done to verify the players' performance evaluation method's architecture that obtained the bigger values in the evaluation procedure proposed. It is assumed that with a better method for evaluating the players' performance, the Commitment can better represent the players and, consequently, the model has more accurate representations and best performance.

Finally, an experiment was made to verify H3 and H4 by comparing the metrics proposed with the ones used in State of the Art (SotA) articles in the churn prediction task.

For the execution of the experiments we considered the holdout validation, splitting the data into two disjoint groups (training and test). To compare the results obtained, the F1-Score was selected. The holdout technique was utilized due to the amount of data, and, to reduce bias, repetitive tests were made using random samplings or by sliding time windows. Regarding the F1-Score, its usage aimed at capturing the recall and the precision in an unbalanced dataset. The recall is relevant because it highlights the number of relevant elements, churners, selected, and the precision to encompass the false positives. The false positives numbers should be small due to the churn prevention campaign's investment, meaning that a high number of false positives would lead to a money loss. More details about the experiments are presented in Chapter 6.

5 Proposed Method

This chapter aims to explain each part of the proposed method, responsible for predicting churn. First, a macro view of the whole process, illustrated by Figure 13, is briefly discussed, highlighting the data extraction procedures. Later, the following sections detail each component of the proposed method, displayed in Figure 14. In both figures, the white and grey rectangles are, respectively, encompassed and disregarded in the method. Finally, a running example is presented to clear any doubts regarding each part of the method's inputs and outputs.

Even though extracting and preparing the data are not included in the method, they were realized as detailed in Section 4.3. Three extractions need to be made. Two are from historical data, executed only once, and another that captures the most recent data.

The first one-time-only extraction, named Login History Data Extraction, includes the login history in the last $n - 1$ days. The n represents the time window in days of the players' behavior that will be analyzed. The minus one is necessary because the extraction is performed in old data, and the most recent data will be included later. The data comprises 0 and 1, indicating if a player played or did not play on each day of the time window. The second one, named Matches' Statistical Outcomes, regards the matches' statistical results, which requires the attributes present in Table 7. It can encompass data from the past week, month or more, and is utilized to induce the players' performance evaluation model. Finally, the Daily Data Usage Extraction includes the most recent game usage data. It has which player played and did not play that day, who played with whom, and the matches' statistical results.

Finalized the extraction process, starts the deployment of the proposed method. If the players' performance evaluation model was not inducted, it utilizes the extractions' data to induct this model. Otherwise, it uses the model inducted to predict the players' performance in the matches played. The next step consists of using the score obtained and the time spent playing to extract each player's Commitment. Meanwhile, the Automatic Labeler process the login history to define churn, label the players and calculate the Churn Definition Change Rate (CDCR). The CDCR can be seen as the amount of change in the labels that a shift in the churn definition caused. More details in the next section. This information and the graph generated using daily data are used to extract the social metrics. Both metrics, social and Commitment, are used to induct the churn classifier, if necessary, and classify the players as churners or non-churners. The churn classifier induction occurs when the current loop is the first one or if the CDCR is greater than a defined threshold. This threshold indicates an acceptable amount of change in the churn labels. When the

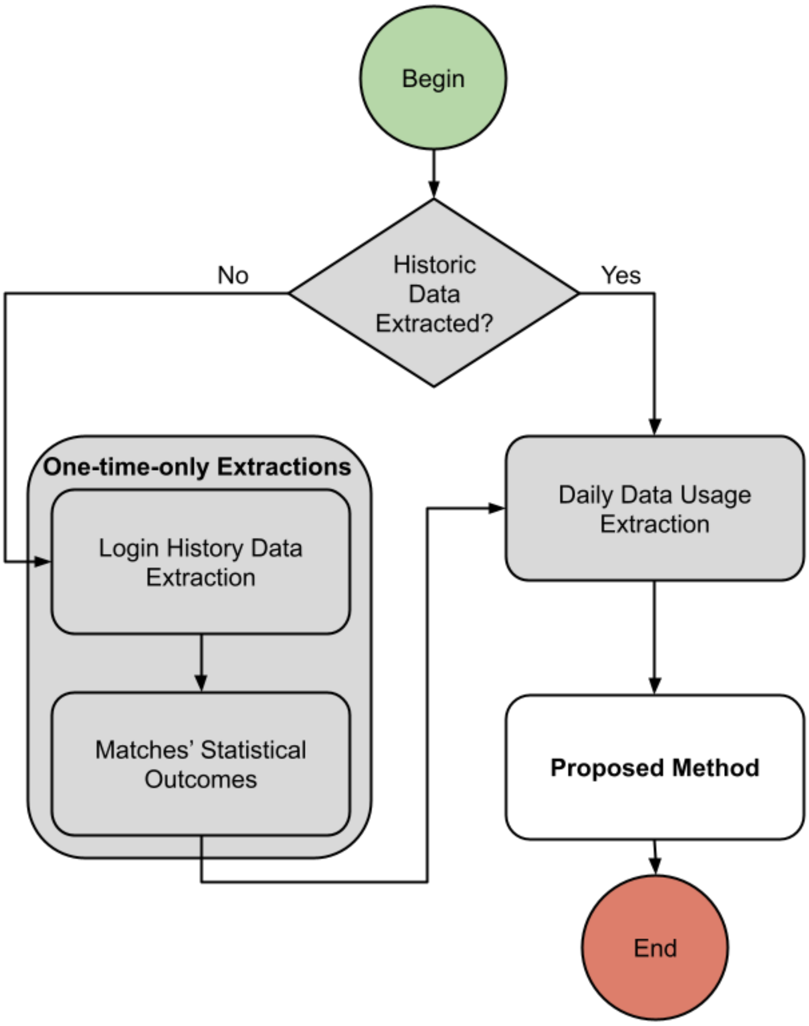
Begin
Historic Data Extracted?
No
Yes
One-time-only Extractions
Login History Data Extraction
Matches' Statistical Outcomes
Daily Data Usage Extraction
Proposed Method
End

Figure 13 – Flowchart illustrating a macro view of whole process

threshold is surpassed, all labels are updated with the new churn definition and used in the induction. Concluded all steps, the process repeats after the next day's data is extracted.

5.1 Automatic Labeler

The first component in the method is the Automatic Labeler. In this step, the goal is to label the players into two classes, churners or non-churners, and verify the impact of a change in the churn definition. This algorithm expects as input a list of zeros and ones representing each player's presence. Zero if the player did not play any game that day, or one if he/she did.

Considering the two major drawbacks of the commonly used labeling approaches, namely, (1) the inability to identify the changes in the churn definition, and (2) the use of the same definition for all players, the Individual Fixed Value (IFV) is proposed. The proposed approach follows the idea of defining a value as a threshold, as done by the FV approach, but focusing on each player separately. This correlation to the FV enables the

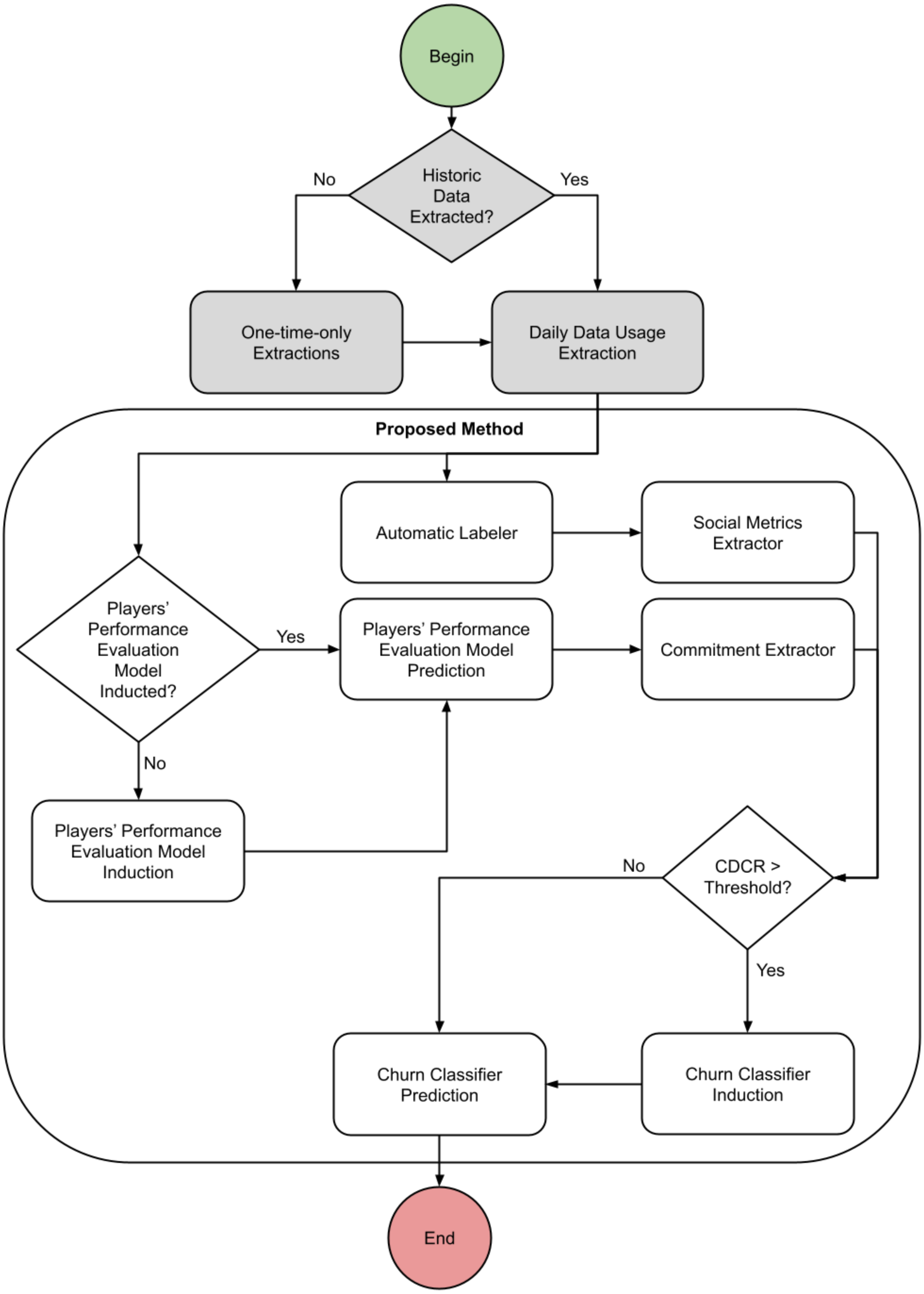
Begin
Historic Data Extracted?
No
Yes
One-time-only Extractions
Daily Data Usage Extraction
Proposed Method
Automatic Labeler
Social Metrics Extractor
Players' Performance Evaluation Model Inducted?
Yes
Players' Performance Evaluation Model Prediction
Commitment Extractor
No
Players' Performance Evaluation Model Induction
No
CDCR > Threshold?
Yes
Churn Classifier Prediction
Churn Classifier Induction
End

Figure 14 – Flowchart detailing the proposed method

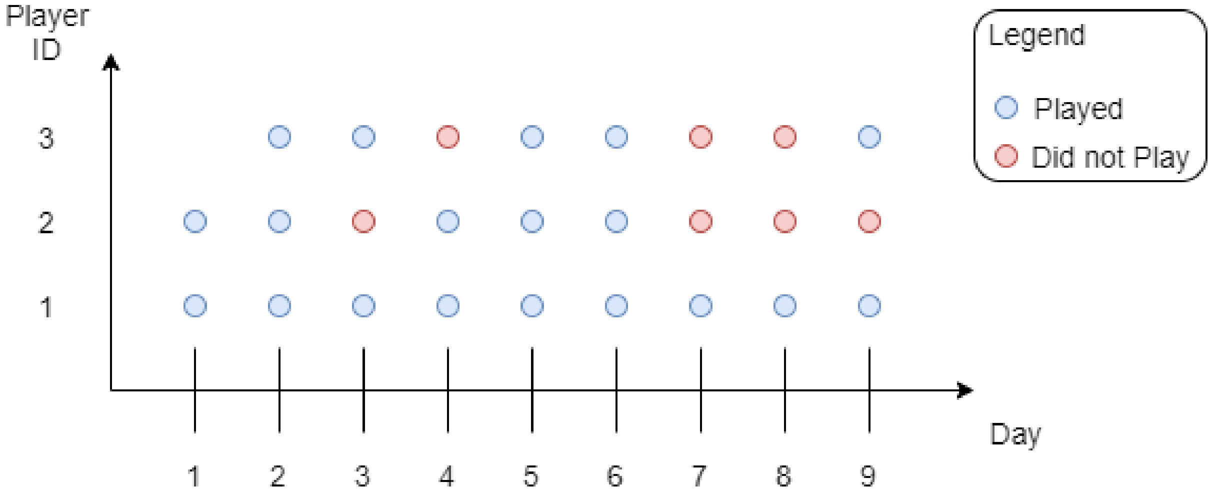

Figure 15 – Fictitious labeling example

possibility to capture changes in the churn definition (solving major problem 1) but, by using an individualized value, it solves the issue of not being able to suit every players' behavior (solving major problem 2). To address churners that churned a long time ago, only absences followed by days played are accounted for, in the same manner as in the FV when a data-driven approach is used.

Following the fictitious example represented by Figure 15, it is possible to apply the proposed IFV approach to label the three players. Using Equation (5.1), where j is the jth player, n_j represents the number of individual absences of this player and i_j is the ith absence with return of this player, each player is linked to a unique IFV. After defining each player's IFV, Equation (5.2) is used to determine if the player is a churner or non-churner and he/she is labeled accordingly. For player 3, two absences, on days 4 and, 7 and 8, followed by a return appeared, resulting in an IFV of 1.5. Since player 3 played on the last day of the dataset, day 9, his/her last absence is zero. By comparing both values, as demonstrated by Equation (5.2), the player is labeled as non-churner. The same calculation can be done to player 2, but since he/she only has one absence, on day 3, the average is the value itself. Unlike player 3, the last absence of player 2 is three, and three is greater than one, the IFV, so he/she is labeled as churner. The last player has an absence with a return of zero since it played every single day. Comparing its last absence of zero with its IFV of zero, we labeled him/her as non-churner. The whole procedure of defining churn and labeling each player according to it can be better comprehended by observing Algorithm 1.

$$IFV_j = \frac{\sum_{i_j=0}^{n_j} AbsenceWithReturn_{i_J}}{n_j} \tag{5.1}$$

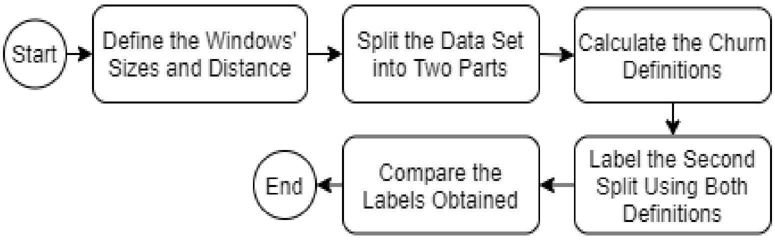
Start
Define the Windows' Sizes and Distance
Split the Data Set into Two Parts
Calculate the Churn Definitions
Label the Second Split Using Both Definitions
Compare the Labels Obtained
End

Figure 16 – Step by step of the evaluation method

$$Label = \begin{cases} \text{Churner}, & \text{if } LastAbsence > IFV \\ \text{Non-Churner}, & \text{otherwise} \end{cases} \tag{5.2}$$

Algorithm 1 Automatic Labeler

```
procedure RUN
    for player in players do
        player[“IFV”] = 0
        for absenceWithReturn in player[“absences”] do
            player[“IFV”] += absenceWithReturn
        player[“IFV”] \= len(player[“absences”])
        if player[“lastAbsence”] > player[“IFV”] then
            player[“label”] = “Churner”
        else
            player[“label”] = “Non-Churner”
    return players
```

Since the churn definition, in our example, the IFV, can change, a method to calculate this change's impact is essential to compare different labeling approaches and adjust the labels and the classifier accordingly. Such a method is proposed and described below.

As illustrated by Figure 16, when using the evaluation in a real-life scenario, the first step is to choose the sizes for the past and current windows, which, in this work, are the same to enhance the identification of seasonal players' behaviors. This process involves an experimental step, as presented in Section 6.1 or manual analysis of the players' behaviors. Regarding the distance between the windows, it is advised to utilize them attached where the first day of the current data is the next of the past's last day, which allows a comparison of the earlier definition and the most recent one. Finalized the process of defining the windows' sizes and the distance, the data are separated into two parts, the past, and current windows, with the length determined in the previous step. It is important to notice that in daily evaluation, as done in this work, the windows can overlap, and the split is not needed. One window will be the current data, and the other will substitute the older data with the new one, advancing one day.

After splitting the data, the chosen approach is deployed in both windows, generating two fixed values (individualized per player or not, depending on the adopted labeling approach). The values obtained are used to label the players present in the current window, producing two sets of labels that will be compared against each other in the last step. By comparing the new and the former definitions of churn in the latest data, it is possible to capture which players had their labels changed, a warning that the labels should be revised. The comparison is made using the F1-Score, keeping in view the unbalanced nature of the data.

Since the value that represents a change in the definition is the disagreement between the two sets of labels and not the agreement, we utilize Equation (5.3) to achieve the Churn Definition Change Rate (CDCR), ranging from 0 to 1. The higher the value of the CDCR, the more influence a change in the churn definition have in the labeling process, resulting in less reliable classification performance. When this value reaches a certain threshold that can be defined depending on specific game designs, players base, data granularity, and other characteristics, it is advised to revise the churn definition and retrain the classification model using more accurate labels.

$$ChurnDefinitionChangeRate = 1 - \mathit{F1Score} \tag{5.3}$$

Algorithm 2 illustrates the evaluation method as a pseudocode. Finalized both processes, the automatic labeler returns a label for each player (Algorithm 1 output) regarding if it is considered a churner or non-churner, and the CDCR (Algorithm 2 output).

Algorithm 2 Churn Labeling Evaluation Method

```
procedure RUN
    trainingWS = 7
    testingWS = 7
    windowsDistance = 0
    for x in range(0,len(dataSet)-(trainingWS+windowsDistance+testingWS)+1) do
        currDataSet = dataSet[x:x+trainingWS+windowsDistance+testingWS]
        trainingDataSet = currDataSet[0:trainingWS]
        testingDataSet = currDataSet[trainingWS +windowsDistance:]
        CDTraining = IFV(trainingDataSet)
        CDTesting = IFV(testingDataSet)
        trainingLabels = Label(testingDataSet, CDTraining)
        testingLabels = Label(testingDataSet, CDTesting)
        f1Score = F1Score(trainingLabels, testingLabels)
        CDCR = 1-f1Score
    return CDCR
```

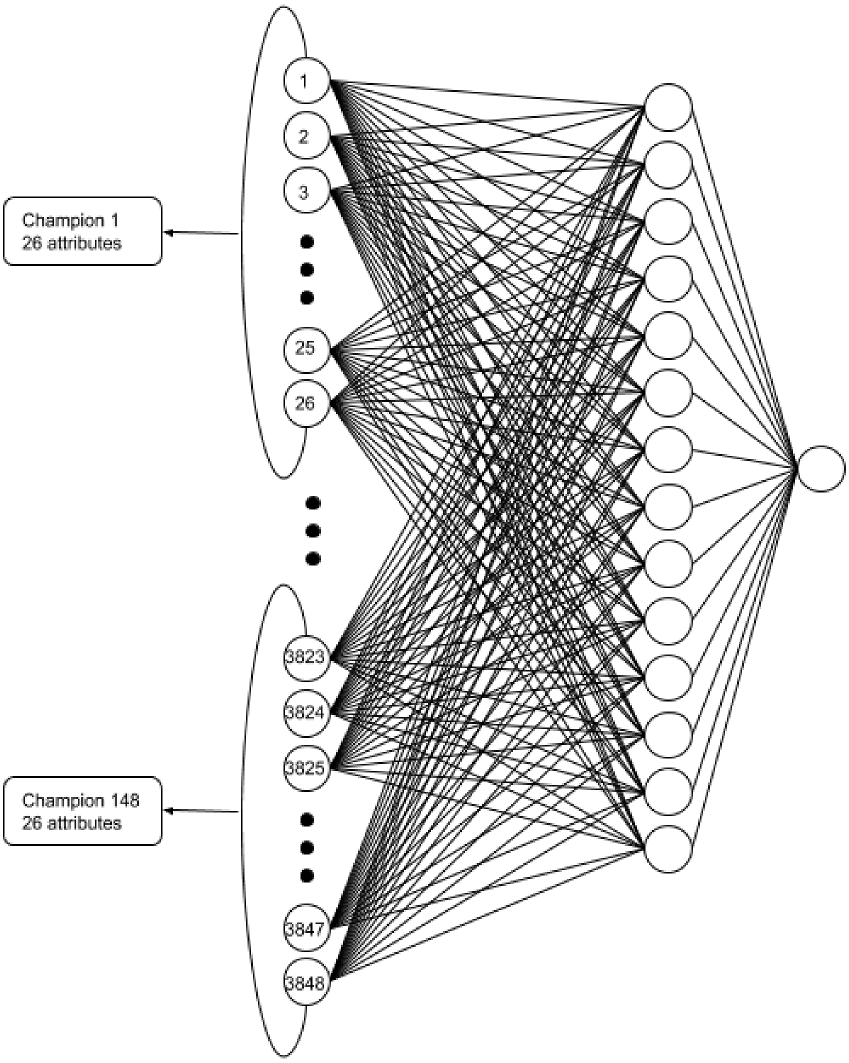

Figure 17 – An example of the MLP used

5.2 Commitment Extractor

The Commitment extractor encompasses four values: time spent playing, min, max, and average players' performance score. The time spent playing can be easily acquired from the daily data usage, but the score is challenging.

Using a similar approach than the one proposed by (PRAKANNOPPAKUN; SINTHUPINYO, 2016), an ANN was used to calculate the weights for each champion's attributes automatically. In this work, we propose using a Multilayer Perceptron (MLP) with 3,848 input neurons, constituted by the 26 attributes of each of the 148 champions. An example is illustrated by Figure 17. The implementation used in this work was developed using the Keras library[1] (CHOLLET et al., 2015) in the PyCharm[2] environment.

The method starts looping through each match, where two inputs are generated, one for each team. The input begins with zeros in all positions and has fixed positions

[1] <https://keras.io>

[2] <https://www.jetbrains.com/pycharm/>

for each character and attribute. For each champion present in a team, its attributes are put in the corrected position. After all the champions in a team have their attributes allocated, the feedforward algorithm can predict a single value at the output. This value is stored, and new input is created based on the steps described earlier with the other team present in the match. In the next step, instead of only predicting a value using the feedforward algorithm, the input is used in the ANN training by deploying the feedforward and backpropagation algorithms. The target is the previous value stored plus or minus an amount. If the team used in the training phase won the match, its value should be higher than the team that lost, so an amount is summed to the stored value. Otherwise, the stored value is subtracted by an amount. Numerous options can be used as this amount. In this work, we tested summing and subtracting the value one, the difference in gold between the teams, and a percentage of the value predicted. A pseucode of the evaluation procedure can be seen in Algorithm 3.

Algorithm 3 Players' Performance Evaluation Method

```
procedure RUN
    amount = 1
    for teamA, teamB in matches do
        teamAOutput = feedforward(nn, teamA['stats'])
        if teamA['win'] == True then
            teamBTarget = teamAOutput - amount
        else
            teamBTarget = teamAOutput + amount
        nn = trainNN(nn, teamB['stats'], teamBTarget)
    return nn
```

With enough matches from every champion and assuming that a winning team has the sum of its players' performances higher than the losing team's performance sum, all characters' attributes have their importance balanced regarding the champion's priorities. After the training phase is over, it is possible to acquire a value representing a player's performance in a match by creating the input for only this player and applying the feedforward algorithm.

After each player's values are acquired, the process proposed by (KUMMER; NIEVOLA; PARAISO, 2018) is used. All attributes are clustered into three groups of engagement: low, average, and high. The clustering algorithm results are used as the players' class in a classification algorithm using the same inputs of the clustering model. Finished the training, the model is stored to be used in an ensemble for the next extractions. For the clustering task, the K-Means algorithm was deployed with three clusters, and for the classification task, an MLP with one hidden layer was utilized. Both algorithms and parameters were chosen following related works and using standard library parameters.

5.3 Social Metrics Extractor

The first step realized of the social extractor is the generation of a graph containing the information of who played with whom and how much interaction they had. The graph contains nodes representing the players, edges indicating if two players played together, and edges' weights the quantity of interaction between them. The interaction is calculated by dividing the number of games played together and the number of total games played by the player being analyzed.

After constructing the graph, all metrics presented, explained, and illustrated in Section 2.5 are calculated for each player and used as an input in the classification task.

5.4 Classifier Induction and Prediction

The last step in the method is to verify the need for retraining the model. This verification is done using the results obtained by the Automatic Labeler. If the current run is the first one, there was insufficient data to compare, meaning that the CDCR is equal to zero and smaller than the threshold. If this was not the first loop, the CDCR is calculated and compared to the threshold. This threshold can change depending on specific game designs, players base, data granularity, and other characteristics, but in this work, it was set to 0.05, allowing the identification of minimal changes. If the CDCR is greater than the threshold, the model is inducted using all the information from current and past days but using the recent churn definition and labels.

After the induction, if necessary, the method will perform the classification step. The algorithm chosen was an MLP due to the State of the Art results found in (ZHENG et al., 2020). It was implemented using the Keras library. All tests utilize two hidden layers and follow a rule of thumb using the number of neurons as 2/3 of the input plus the output (HEATON, 2015). Fine-tuning and tests using other algorithms will be performed in the near future.

The classifier's input can be seen as a vector of six dimensions containing the two extractors' outputs. Bellow there is an example represented by $\vec{v}$.

$$\vec{v} = (0.5, 0.3, 138, 2.5, 0.0062, 0.0072)$$

These outputs represent the commitment category, churn influence, node degree, average neighbors' degree, clustering coefficient, and transitivity. The only value that needed to be converted was the commitment category, which represents a nominal value. Since there are three levels of engagement, the nominal values low, average, and high, are converted to 0, 0.5, and 1, respectively. After the conversion, all values are scaled using the min-max scaler implementation of the scikit-learn[3].

5.5 Running Example

In the first run, the one-time-only extractions are executed. The data from numerous players and matches are extracted, but, to facilitate the understanding, only the data from one player and one of his/her matches played are presented. In the Login History Data Extraction, the information retrieved is a series of ones and zeros representing the days played or not played in a time window. As done in the experiments, the time window chosen was seven days. Since this data is from the past, the number of days considered will be six (the idea of $n - 1$). The *LoginHistoryData* list illustrates this information.

$$LoginHistoryData = [1, 0, 0, 1, 1, 0]$$

The next one-time-only extraction regards the Matches' Statistical Outcomes, which have the champion's ID of the character played, all the attributes listed in Table 7 and if the player won or lost. An example of the data extracted from one match can be seen in Table 9.

Finalized the one-time-only extractions, the Daily Data Usage Extraction is performed. It captures the most recent data regarding the players' presence on that day, if he/she played, the matches' statistical outcomes for the matches played, and who played with whom. The frequency gathered is added to the previous list, which now has seven elements. The matches' statistical outcomes have the same format presented in Table 9 but from the matches played on the last day of the dataset, with the players' IDs present in that match, and without the "champion's ID" and "win" fields.

After extracting all the information needed and considering that this is the first run, the Players' Performance Evaluation Model is inducted using the Matches' Statistical Outcomes. It receives as input the data present in Table 9 and inducts a model capable of returning a value that represents the sum of the champions' performance present in a specific match. Finalized the induction, the matches' statistical outcomes obtained from the Daily Data Usage Extraction are used as input in the model, and the players' performance minimum, maximal and average values are obtained. The remaining metric needed for the Commitment Extractor is the "time spent playing", which is acquired by summing the game's duration that a player attended. An example of the inputs passed to the Commitment Extractor can be seen in Table 10.

[3] <https://scikit-learn.org/stable/index.html>

Table 9 – Example of the data extracted in the Matches' Statistical Outcome process

Attribute Name	Value
champion ID	236
game_duration	3,038
champ_level	18
kills	9
deaths	7
assists	12
longest_time_spent_living	1,352
farm	239
gold_earned	19,938
gold_per_min	361.54
xp_per_min	375.69
physical_damage_dealt_to_champions	21,777
magic_damage_dealt_to_champions	977
true_damage_dealt_to_champions	1,627
damage_dealt_to_towers	4,403
damage_dealt_to_objectives	36,203
physical_damage_dealt	231,095
magic_damage_dealt	6,214
true_damage_dealt	9,351
physical_damage_taken	10,731
magic_damage_taken	16,956
true_damage_taken	809
damage_self_mitigated	13,811
total_heal	6,387
time_ccing_others	0
total_time_crowd_control_dealt	19
vision_score	51
win	true

Table 10 – Example of the inputs passed to the Commitment Extractor

Commitment Feature	Value
Minimum Score	65.55
Maximal Score	88.48
Average Score	78.46
Time Spent Playing	8,559

Table 11 – Example of the Social Metrics Extractor's outputs

Social Metric	Value
Churn Influence	0.2
Degree	48
Average Neighbors' Degree	64
Transitivity	0.0064
Clustering Coefficient	0.0045

Table 12 – Example of the Churn Classifier's inputs

Metric	Value
Commitment Category	0.5
Churn Influence	0.2
Degree	48
Average Neighbors' Degree	64
Transitivity	0.0064
Clustering Coefficient	0.0045

At the Commitment Extractor, these inputs are used to cluster the players into three categories, low, average, and high. Using the clustering algorithm's same inputs, the classification algorithm utilizes the clustering algorithm's output as the labels. The Commitment classification model is stored and will be used in the next runs as an ensemble. The result is a label representing each player's commitment group, low, average or high.

While the previous steps are performed, the Automatic Labeler utilizes the list containing the players' frequencies to define their Individual Fixed Values (IFV) and label them as churners or non-churners. The output of the Automatic Labeler is each player's IFV and label, and the CDCR. Since this is the first run, the CDCR is set to zero. The next procedure is the Social Metrics Extractor. Using the daily information of who played with whom, a graph containing players as nodes, edges as interactions between them, and weights, the amount of interactions, is constructed. This graph is passed as input to the Social Metrics Extractor, and an example of the outputs can be seen in Table 11.

After the metrics are extracted, the CDCR is compared to the threshold. The threshold defined was 0.5 and the CDCR, considering this as the first run, is equal to zero. Comparing both, the method needs to induct the Churn Classifier. The inputs are the combination of the Commitment and the Social Metrics Extractors outputs and can be seen in Table 12.

After the model's induction, the Churn Classifier is used to label the players based on the previous inputs. It outputs, for each player, a label representing if the player will or will not churn. The most recent data is extracted after one day, now excluding the one-time-only extractions, and the whole process repeats. The differences from the first run are that the one-time-only extractions are disregarded, the Players' Performance Evaluation Model is already inducted, there are more Commitment classifiers, and the CDCR can be calculated comparing the previous data and the most recent one. If the CDCR is greater than the threshold, all the previous data is utilized to induct the churn classifier model using the most recent churn definition and labels.

Even though the method is complex and has many components, this work presents running examples, codes, and images to help other authors reproduce it. Excluding the algorithms' parameters explicitly discussed, all the others were set as the default of the libraries used.

6 Experiments

To answer RQ2: “Should the churn definition be updated?” and verify all hypotheses, six experiments were done. In this chapter, each experiment will be detailed about its objective and execution. To better comprehend the experiments, they were divided into three categories:

- The experiment regarding churn labeling;
- Four experiments aiming to improve the method responsible for evaluating the players’ performance;
- The comparison between the metrics used for predicting churn.

6.1 Churn Labeling Experiment

In this experiment, a comparison between the most used and the proposed labeling approach was performed. To compare the approaches, the evaluation method presented in Section 5.1 and illustrated by Figure 16 was used. Performing the evaluation requires awareness of two details. The first one refers to the windows’ sizes of the past and current data parts used in the analysis of each approach. To include various behaviors, windows encompassing 7, 14, 21, 30, 60, 90, 180, and 270 days were chosen for the experiment. The max value was set to 270 days because the smallest dataset has approximately 700 days, which is almost the size of the sum of the two windows.

The second alludes to the windows that could have better or worse performance regarding identifying changes on the churn definition. Desiring to minimize this effect, each window in the experiment scrolls through the whole dataset day by day, and the scores obtained are averaged.

Since the only approaches that can be used to calculate a change in the churn definition are the Fixed Value (FV) and Individual Fixed Value (IFV), the Naive and the Sliding Window approaches were excluded. Entering in more detail, for the FV approach, the value utilized in the labeling process was chosen following (RUNGE et al., 2014), averaging the absences of all players in the window used in the evaluation. An important note is the definition of absence. Since long-term churners can influence this average, we consider an absence the number of consecutive days not played followed by at least a day which the player played. The IFV follows the same process, but the average is calculated separately for each player.

The Churn Definition Change Rate (CDCR) computation considers one past and one current window at a time, generating a value indicating the disagreements between the old and new definitions of churn. As stated earlier, since the sizes of the windows and the distance used can influence the result, several CDCRs were performed for experimental purposes, and their average was used as the final result. After calculating the CDCR, the past and current windows scroll through the dataset, advancing one day at a time and generating new sets of past and current to be computed. This process continues until the last day. When the end of the dataset is reached, all CDCRs obtained are averaged and stored, referencing the adopted window size. Next, the same whole process is performed considering a different size for the windows, starting from the first date.

6.2 Players' Performance Evaluation Experiment

For the evaluation of the players' performance evaluation experiments, a procedure is proposed. Since the players' history and rank are not desired in this evaluation, the data does not have a label, and automatically balanced weights are not present in unsupervised learning, the match outcome was used. Finished the training phase, the testing data is run in the model following the same idea of separating the teams and constructing two inputs. Each input is used in the feedforward algorithm to acquire two outputs. If the team that won has a bigger output value than the losing team, the instance is considered correct. Otherwise, it is an error. Since the number of teams that win and lose is the same, the dataset can be considered balanced. Bearing in mind a balanced dataset, accuracy was used to measure the model's performance in the testing phase.

Using the League of Legends (LOL) dataset containing the attributes presented in Table 7, three experiments were done. The first one had the objective of finding an appropriate architecture for the Artificial Neural Network (ANN). To achieve it, combinations of its parameters (e.g., optimizer, activation function, and loss function) and the possible preprocessing algorithms were made. In this experiment, the data were reduced to 5% of its size, separated using the holdout technique, 67% for training and 33% for testing, and three different seeds were utilized for the random sampling. A list of the parameters tested can be seen in Table 13.

Due to the enormous quantity of permutations, a grid search was not viable and the grid step wise approach was used, where each parameter was tested separately with two random values of the untested parameters. As a running example, the normalizer could be combined with two random optimizers (e.g., Adam and Nadam), activation functions, and loss functions. For the number of hidden layers and neurons in each layer, empirical values were chosen to reduce the analysis's bias. The values determined were, respectively, 2, 2,565, and 1,710. Two hidden layers were considered ideal for this task because the

Table 13 – Parameters used in the experiment

Normalization Algorithm	Optimizer	Activation Function	Loss Function
Normalizer	Ftrl	selu	Huber
Standardizer	SGD	tanh	logcosh
Robust Scaler	Adam	linear	Cosine Similarity
Max-Abs Scaler	Nadam	sigmoid	Mean Squared Error
Min-Max Scaler	Adamax	softplus	Mean Absolute Error
	Adagrad	softsign	Mean Squared Logarithmic Error
	Adadelta	leakyrelu	Mean Absolute Percentage Error
	RMSprop	exponential	

solution could not be linear and continuous, and the features were handmade, meaning that automatic feature engineering is not necessary. Regarding the number of neurons, following (BOGER; GUTERMAN, 1997), 2/3 of the inputs were used for the first hidden layer, and 2/3 of the first hidden layer were used in the second one. Another measure utilized to reduce the computational cost was to utilize the same activation functions for the hidden layers.

After defining the parameters, the method is deployed as described in 5.2, and the accuracy obtained from the three testing samples are averaged. The next step is to define a new combination of parameters using the next preprocessing technique and deploy the method again. When one parameter had all its components tested, the one that had the bigger accuracy is selected, fixed, and not included in the next permutations. The next parameters follow the same process explained above until all have been tested and fixed.

The second experiment aim at checking the accuracy of the model when different maps are put together. Using the architecture found in the last experiment, the data is separated regarding the map a match was played. Four groups of data were constructed, only matches from Summoner's Rift (SR), only matches from Howling Abyss (HA), matches from both maps in both training and testing datasets, and matches from SR in the training and HA in the testing dataset. Ten runs were performed to minimize the bias created by the non-deterministic algorithms present in the Keras library, and their accuracy was averaged.

Lastly, an experiment was performed to find the most appropriate value for training the model. When the training phase starts, the target value used to fit the model is the value predicted of the other team in the match plus (if the current team won) or minus (otherwise) an amount. The amount could be a constant or a variable that depends on the results of the match. In this work, five amounts were chosen to be tested, the constant one, two ratios of the predicted value, and two ratios of the gold difference between the teams. The percentages were chosen arbitrarily based on the maximal and minimal values of the amounts used, and the modulus was applied to exclude calculation errors with negative numbers. While the constant was chosen to indicate slight changes among the teams, considering fair matches, the prediction ratio is an attempt to utilize a percentage of the opposing team's score to indicate that a team was $n\%$ better or worse than the performance of the other team. Lastly, in LOL, analysts consider that the bigger the amount of gold a team earned, the more significant this team's advantage was against their opponent, meaning that it could be a good indicator of the performance difference.

6.3 Churn Prediction Experiment

The last experiment regards the churn prediction task. The method presented in Chapter 5 and illustrated by Figure 14 was utilized in all the tests, changing only the metrics used as inputs of the classifier. Four sets of metrics were used in this experiment, a baseline metric, the social metric, the Commitment, and the social combined to the Commitment.

Numerous works such as (MILOŠEVIĆ; ŽIVIĆ; ANDJELKOVIĆ, 2017; RUNGE et al., 2014; HADIJI et al., 2014), and (YANG et al., 2019) utilize the time spent playing for predicting churn. Considering (ZHENG et al., 2020), the State of the Art algorithms generally use login information in various classification algorithms. With this in mind, the baseline metric is composed of the players' login information, their time spent playing, and the classification algorithm is a Multilayer Perceptron Neural Network, the algorithm that got the best results.

First, a window containing seven days of data was used to label the players and calculate the metrics. Later, another window containing seven days of data counting after the last day of the first window was constructed. This second window was used to label the players and is considered the true labels because they contain the most recent data. The labels of the first window were compared to the second window, and the F1-Score was calculated. Finished the first analysis, the first window advances one day, and the process repeats. When the second window reached the last day, all the F1-Scores obtained were averaged. Due to the non-deterministic behavior of the Keras library's implementation, all metrics were tested three times, and their F1-Score and Loss Function values were averaged.

7 Results and Discussions

The results and discussions are separated regarding the experiments, first the churn labeling experiment, detailed in Section 6.1, followed by the players' performance evaluation, Section 6.2, and, lastly, the churn prediction experiment which was presented in Section 6.3.

7.1 Churn Labeling Results

After computing all the Churn Definition Change Rates (CDCR) for all window sizes, the results were separated regarding each dataset. Table 14 represents, respectively, the averages of the CDCRs obtained for each sliding window on the League Of Legends (LOL) and the World Of Warcraft (WOW) datasets in a percentage perspective. By summarizing the results, it is possible to perform three comparisons regarding the average, one considering the dataset, another the approach, and the windows' sizes used. Bearing in mind the two datasets, in most cases, the LOL dataset has higher values of CDCR than the WOW dataset. When comparing the two approaches, the FV approach's CDCR value is smaller than the IFV in both games. Lastly, if focusing on the windows' size, while in the LOL dataset, it appears to have no pattern to the obtained average, in the WOW dataset, the higher the windows' sizes, the higher the CDCRs average.

Looking from the LOL perspective, it is interesting to notice that the FV's general overview presents little variance of the players' behavior, ranging from 0% to 2.84%. However, the IFV's specific perspective shows a more significant variance, from 11.71% to 23.32%. Considering that in the churn management each retained player is worth (KARNSTEDT et al., 2010), the adoption of the IFV approach presents benefits in comparison to the FV, as it can better indicate when the churn definition of players change, considering an individualized perspective. This entails in more reliable labels to be used by the churn prediction classifier. Note that if a game producer adopted a threshold of 5% to relabel and retrain the classifier, it would never happen in LOL if the FV was adopted, where it should, as the players' behaviors are changing in the individualized perspective and the classifier keeps working with an out-of-date notion of churn.

Considering the results obtained from the WOW dataset, the correlation between the size of the windows and the CDCRs remained unclear. Two suppositions emerged, one regarding the distance between the windows and another that explains this phenomenon using the players' behavior. Bearing in mind the suppositions, one more experiment was performed in the WOW dataset following the same procedures of the previous experiment but moving only the second window while the first remained static. Maintaining the past window static makes it possible to verify if the CDCR values increase within the distance, proving the first supposition. The results obtained from the FV and the IFV can be seen, respectively, in Figures 18 and 19.

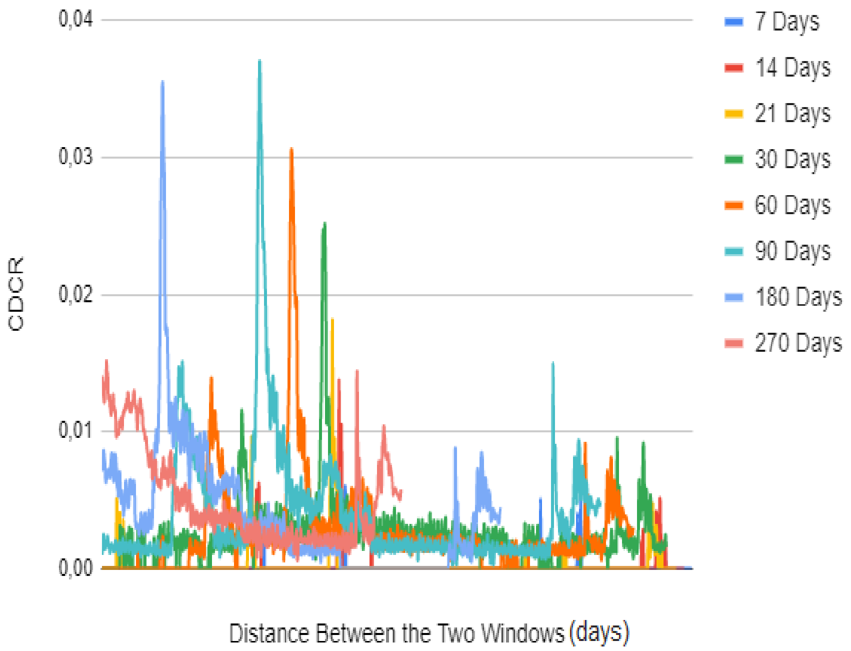

Figure 18 – The CDCR values from moving the second window and using the FV approach on the WOW dataset

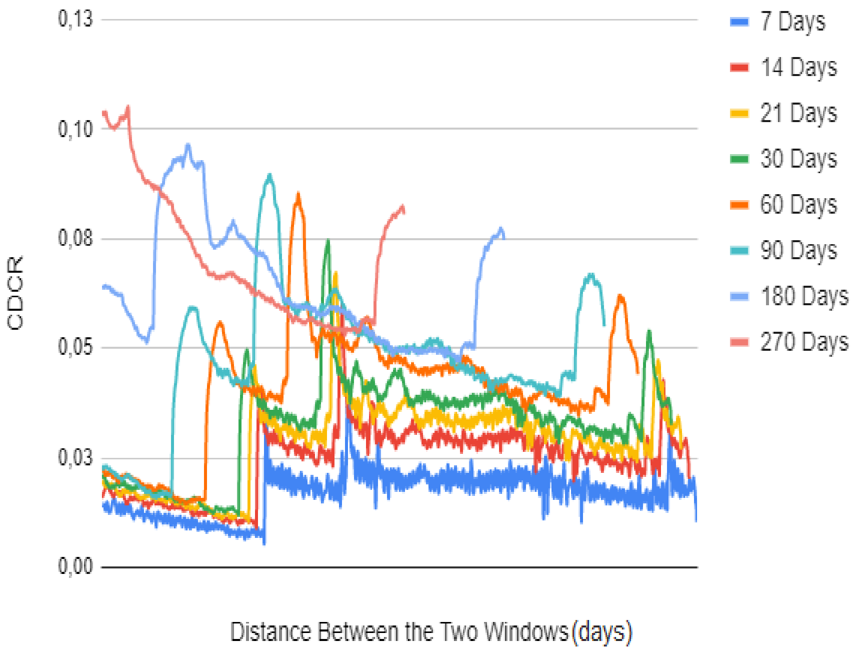

Figure 19 – The CDCR values from moving the second window and using the IFV approach on the WOW dataset

Even though it is hard to observe each line separately, it is possible to notice that none had its values of CDCR increasing together with the distance between the

Table 14 – CDCR for each game

Game	Approach	Window Size (days)	CDCR Average (%)
LOL	Fixed Value	7	1.54
LOL	Individual Fixed Value	7	20.51
WOW	Fixed Value	7	0.01
WOW	Individual Fixed Value	7	1.12
LOL	Fixed Value	14	0
LOL	Individual Fixed Value	14	23.32
WOW	Fixed Value	14	0.03
WOW	Individual Fixed Value	14	1.41
LOL	Fixed Value	21	0.43
LOL	Individual Fixed Value	21	22.73
WOW	Fixed Value	21	0.05
WOW	Individual Fixed Value	21	1.57
LOL	Fixed Value	30	2.84
LOL	Individual Fixed Value	30	21.65
WOW	Fixed Value	30	0.10
WOW	Individual Fixed Value	30	1.79
LOL	Fixed Value	60	0.11
LOL	Individual Fixed Value	60	19.12
WOW	Fixed Value	60	0.18
WOW	Individual Fixed Value	60	2.35
LOL	Fixed Value	90	1.58
LOL	Individual Fixed Value	90	18.23
WOW	Fixed Value	90	0.25
WOW	Individual Fixed Value	90	2.88
LOL	Fixed Value	180	2.60
LOL	Individual Fixed Value	180	17.37
WOW	Fixed Value	180	0.36
WOW	Individual Fixed Value	180	4.53
LOL	Fixed Value	270	1.74
LOL	Individual Fixed Value	270	11.71
WOW	Fixed Value	270	0.35
WOW	Individual Fixed Value	270	5.32

two windows. The results obtained and considering that this behavior was not present in the LOL dataset, the most probable cause of this phenomenon is the changes in the players' behaviors. Other data that validate this hypothesis is the higher standard deviation encountered in bigger windows than in smaller ones. This experiment proves that, differently from the LOL dataset, the WOW players' behaviors change gradually in a specific direction, which causes the CDCR values to be more distanced from the average the more data are analyzed. This fact exemplifies the process of players passing through different motivational stages (ZHU; LI; ZHAO, 2010; COOK, 2007).

Taking into account the results obtained from the datasets comparison, it is possible to assume that the players' behavior contained in the WOW dataset are dissimilar to the LOL, because of the difference in the CDCRs, and that each game should be analyzed separately when deciding the best techniques or parameters for the labeling process.

Focusing on the averages obtained by each approach, independent of the dataset, the values of CDCR using the FV approach are smaller, in every window size, than the IFV, demonstrating that with the arrival of new data, the individual definition of churn changes more and has more impact in the labels than the general definition. Given that, it is possible to assume that a fixed value representing a whole player base's churn definition cannot represent every player's definition.

By deploying a new experiment where only the second window is moved across the dataset, another doubt was answered concerning the correlation between the churn definitions' distance and their impact on the resulting labels. The results showed that the values of CDCR do not change proportionally to the distance between the churn definitions, meaning that the behavior presented in the comparison between approaches in the WOW dataset was probably caused due to a gradual change in the players' behavior.

Finally, by observing the values of CDCR, primarily on the LOL dataset, it is safe to conclude that new data's arrival leads to new definitions of churn and, consequently, considerable influence in the labeling process, causing an unreliable performance of the classifier. The results obtained from the proposed evaluation method showed that more attention is needed when defining churn and prove that the method can be used to evaluate situations of risk in the classification task, helping producers determine when it is necessary to re-evaluate the definition and retrain the classifier.

7.2 Players' Performance Evaluation Results

For the first players' performance evaluation experiment, the normalization algorithm that got the bigger accuracy was the min-max scaler, followed by the max-abs scaler and the normalizer. When considering the optimizers, the ranking was respectively adam, adamax, and nadam. The loss functions were ranked from first to third, mean absolute error, mean squared error, and logcosh. Finally, for the activation functions a permutation was needed to verify all the possibilities, and the combination that got the bigger accuracy was selu, selu, and linear. All the results acquired can be seen in Tables 15, 16, 17, 18, and 19. The results obtained regards 5% of the dataset, this measure was adopted due to time constraints, and, even though it can not represent the best parameters, it is appropriate to find a baseline for the data utilized. Since these results could change depending on the dataset used, the following architecture can be used by other authors as a baseline, and a fine-tuning could be done to improve their models' performance. Summarizing, the

Table 15 – Normalization algorithms comparison

Algorithm	Accuracy (%) Seed = 1	Accuracy (%) Seed = 4	Accuracy (%) Seed = 6	Average Accuracy (%)
Normalizer	57.75	55.26	50.77	54.99
Standardizer	43.43	38.34	37.67	39.81
Robust Scaler	28.29	38.59	28.50	31.79
Max-Abs Scaler	65.08	56.97	57.75	59.93
Min-Max Scaler	**67.48**	**65.49**	**65.19**	**66.05**

Table 16 – Optimizers comparison

Optimizer	Accuracy (%) Seed = 1	Accuracy (%) Seed = 4	Accuracy (%) Seed = 6	Average Accuracy (%)
Ftrl	15.70	19.09	18.50	17.77
SGD	56.27	58.59	56.58	57.15
Adam	68.48	**71.50**	**70.65**	**70.21**
Nadam	72.13	68.27	64.52	68.31
Adamax	**68.85**	70.54	67.42	68.94
Adagrad	54.89	56.48	55.52	55.63
Adadelta	52.99	51.61	49.07	51.23
RMSprop	62.50	59.44	62.40	61.45

Table 17 – Loss Functions comparison

Loss Function	Accuracy (%) Seed = 1	Accuracy (%) Seed = 4	Accuracy (%) Seed = 6	Average Accuracy (%)
Huber	70.12	68.69	68.16	68.99
Logcosh	70.81	70.44	71.54	70.93
Cosine Similarity	50.34	52.09	52.46	51.63
Mean Squared Error	71.28	71.60	71.87	71.58
Mean Absolute Error	70.86	71.92	72.45	71.74
Mean Squared Logarithmic Error	76.94	52.46	48.12	59.18
Mean Absolute Percentage Error	69.06	67.74	69.54	68.78

final architecture was a Multilayer Perceptron Neural Network, chosen due to preliminary experiments, with two hidden layers, and the number of neurons present in the input, hidden layers, and output are respectively 3848, 2565, 1710, and 1. The activation functions were, in order, selu, selu, and linear, and the optimizer was adam. For the loss function, it was used the mean absolute error, and the normalization algorithm used in the dataset was the min-max scaler.

Table 18 – Activation functions comparison part 1

Activation	Activation	Accuracy (%)	Accuracy (%)	Accuracy (%)	Average
Function 1 & 2	Function 3	Seed = 1	Seed = 4	Seed = 6	Accuracy (%)
selu	selu	44.03	77.12	64.81	61.99
selu	tanh	66.34	72.35	67.67	68.78
selu	**linear**	76.83	75.79	80.18	**77.60**
selu	sigmoid	74.56	72.69	81.70	76.32
selu	softplus	69.29	69.98	71.71	70.33
selu	softsign	61.07	53.39	68.21	60.89
selu	leakyrelu	68.06	52.06	64.86	61.66
selu	exponential	71.51	51.76	69.24	64.17
tanh	selu	68.80	70.43	68.16	69.13
tanh	tanh	73.58	72.49	79.04	75.04
tanh	linear	64.32	70.08	56.34	63.58
tanh	sigmoid	74.46	76.04	67.37	72.63
tanh	softplus	82.00	75.50	63.53	73.68
tanh	softsign	69.79	72.64	68.31	70.25
tanh	leakyrelu	74.12	75.06	61.81	70.33
tanh	exponential	56.54	64.71	74.27	65.17
linear	selu	69.44	69.10	61.66	66.73
linear	tanh	66.19	69.79	82.94	72.97
linear	linear	41.07	67.03	49.84	52.65
linear	sigmoid	64.76	62.10	63.88	63.58
linear	softplus	78.21	79.73	28.81	62.25
linear	softsign	61.31	30.58	70.38	54.09
linear	leakyrelu	50.14	52.55	73.38	58.69
linear	exponential	69.93	66.93	68.41	68.42
sigmoid	selu	63.19	67.18	68.65	66.34
sigmoid	tanh	64.12	70.18	64.12	66.14
sigmoid	linear	72.74	71.02	65.21	69.65
sigmoid	sigmoid	71.85	41.07	70.33	61.09
sigmoid	softplus	65.30	70.13	61.27	65.57
sigmoid	softsign	77.81	57.23	71.41	68.82
sigmoid	leakyrelu	76.09	66.19	74.96	72.41
sigmoid	exponential	73.63	71.90	57.33	67.62
softplus	selu	79.68	66.78	66.14	70.87
softplus	tanh	56.44	68.95	78.36	67.91
softplus	linear	75.65	70.62	60.23	68.83
softplus	sigmoid	72.94	64.37	65.80	67.70
softplus	softplus	65.11	69.69	72.94	69.24
softplus	softsign	78.70	68.11	72.05	72.95
softplus	leakyrelu	70.67	70.33	68.90	69.97
softplus	exponential	72.99	24.08	68.46	55.18

Table 19 – Activation functions comparison part 2

Activation Function 1 & 2	Activation Function 3	Accuracy (%) Seed = 1	Accuracy (%) Seed = 4	Accuracy (%) Seed = 6	Average Accuracy (%)
softsign	selu	60.03	67.52	73.13	66.90
softsign	tanh	48.81	70.67	73.48	64.32
softsign	linear	77.76	68.70	82.30	76.25
softsign	sigmoid	51.91	76.88	65.80	64.86
softsign	softplus	63.48	70.13	41.07	58.23
softsign	softsign	78.01	74.66	71.66	74.78
softsign	leakyrelu	65.75	60.53	64.91	63.73
softsign	exponential	66.73	77.47	67.32	70.51
leakyrelu	selu	64.61	65.99	72.05	67.55
leakyrelu	tanh	77.86	66.19	71.46	71.84
leakyrelu	linear	60.33	77.27	39.94	59.18
leakyrelu	sigmoid	70.87	69.88	71.26	70.67
leakyrelu	softplus	67.32	71.16	66.58	68.36
leakyrelu	softsign	79.98	43.39	57.08	60.15
leakyrelu	leakyrelu	65.06	63.83	63.38	64.09
leakyrelu	exponential	73.04	58.36	52.30	61.23
exponential	selu	75.30	74.66	64.86	71.61
exponential	tanh	75.01	67.47	71.31	71.26
exponential	linear	52.01	73.48	48.46	57.98
exponential	sigmoid	61.56	72.05	20.64	51.42
exponential	softplus	73.13	76.09	80.37	76.53
exponential	softsign	57.28	58.06	61.02	58.79
exponential	leakyrelu	78.95	78.55	60.23	72.58
exponential	exponential	40.73	73.28	57.13	57.05

In the next experiments, the previous ones' architecture was used together with the whole dataset's data. The results from the second experiment can be seen in Table 20. The accuracy of only using the SR map ranged from 79.98% to 96.94%, having its average as 91.66%. Using the HA map for training and testing showed a significant decrease in the accuracy, possibly due to the small number of matches of this type of map in the dataset. The accuracy of 75.92% present when training with the SR map and testing with the HA shows that they possess similarities between them. Without seeing any information from the HA map, this test achieved higher accuracy than when training only with the HA map. Finally, the accuracy obtained when using both maps, besides being smaller than the one obtained from training only with the SR map, encompasses two different game modes, which can be considered a better alternative to a more general context. Wishing to cover various game modes, training using both maps was chosen for the method and the next experiments.

Finally, the last experiment regarding the amount that could be used to increment or reduce from the winning or losing team score was tested. The results are shown in

Table 20 – Maps comparison

Map Present in Training	Map Present in Testing	Average Accuracy (%)
SR	SR	**91.66**
HA	HA	68.61
SR	HA	75.92
SR+HA	SR+HA	89.89

Table 21 – Results of the different score differences used in the method

Score Difference (predicted_value +-)	Average Accuracy (%)
1	89.89
abs(predicted_value/100)	**93.83**
abs(predicted_value/50)	92.07
abs(gold_difference/20,000)	0
abs(gold_difference/10,000)	0

Table 21. The accuracy obtained from the constant one is the same as the previous results because it was used as the previous experiments' baseline. It was chosen assuming the matches' fairness, and it would indicate a slight difference in performances between the teams. For the next amounts, the ratio was determined by observing their maximal and minimal values, which resulted in their normalization. It is possible to notice that even though the gold difference could be considered a good indicator for performance difference, its accuracy was zero. It is assumed that this occurred due to the uniqueness of each match. An example where this indicator is flawed is in the comparison of a longer and a shorter match. If an assumption that a short match indicates that a team was much better than the other was made and considering that less gold is collected with less match time, the difference of gold would not represent the difference of performance. Lastly, both amounts that got the bigger accuracy were the predicted value ratios, meaning that the teams' performance is interconnected. Considering that there are objectives shared among the two teams and that each lane is a direct confrontation, each player's performance from a team impacts the other team's performance.

7.3 Churn Prediction Results

The last experiment regards the comparison of the metrics in the churn prediction task. The results are illustrated by Figure 20. In this last experiment, it is possible to notice that the baseline and the social features start with low F1-Scores and the Commitment or the combination of Commitment and social have a higher F1-Score at the first window of analysis. It is also possible to observe that to reach its highest value of F1-Score, the metrics that demand fewer data are respectively the social combined with the Commitment, the social, the Commitment, and the baseline. These behaviors indicate that joining the social metrics with the Commitment cause their best characteristics, a better start, and a fast adjustment to the players' behavior, to remain and be amplified. This improvement can be proven due to the highest F1-Score obtained in the first window by the union of the two metrics and their fast rise to its maximum score.

Averaging the results from all windows can indicate the better metrics, among the ones tested, to be used in the churn prediction task. The average for the baseline, Commitment, social, and social plus Commitment are respectively 0.7385, 0.8021, 0.7941, and 0.8682. These results show that the social metrics improve the classifiers' performance compared to the baseline, which encompasses the most used metric in State of the Art (SotA) articles (YANG et al., 2020). The Commitment obtained similar results to the social metric, and their union resulted in the highest F1-Score. These findings prove H3 and H4 hypotheses and demonstrate that combining the Commitment with the proposed social metric is a valuable discovery that improves the churn prediction classifier's performance compared to related works.

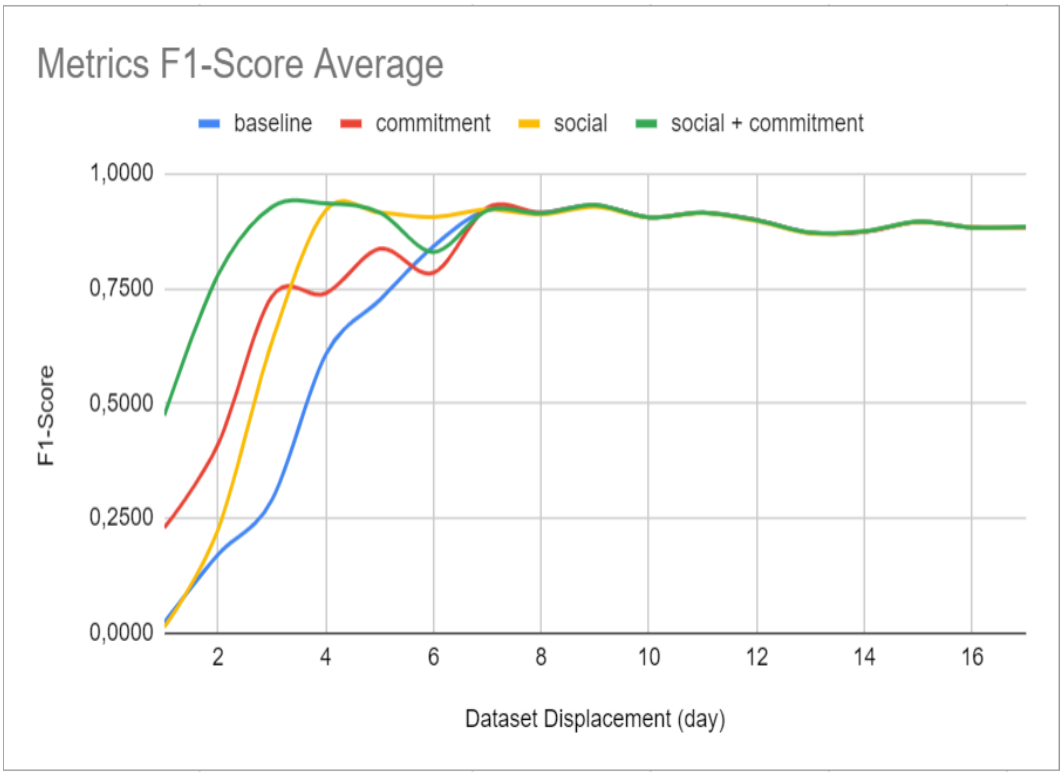

Figure 20 – The F1-Score average of the three runs for each metric in the churn prediction task

8 Conclusion and Future Works

This chapter will summarize all the work done, present the limitations, and propose future works. First, an initial search for gaps in the churn prediction task in games was done. In this preliminary research, it was found that there is not much research about social features in games for the churn prediction task, even though its importance is proved. With this information in mind, a systematic mapping was made to acquire references and further explore these topics. After the systematic mapping, it was identified that aside from the social aspects, the churn labeling process lacked a study to encounter and solve its challenges.

Joining all the information acquired, this work focused on the players' labeling process, which is crucial for a reliable churn prediction, evaluating the players' performance, and churn prediction. Regarding churn labeling, as far as our knowledge goes, this study is the first to analyze this process from a quantitative perspective. We investigated related works to find what could be improved and found three issues, answering RQ1: "What are the issues present on the churn labeling task in games?". When defining churn, the main problems are related to the nonexistence of two things, a method to evaluate the best definition and the true labels used to compare with the response generated by each approach.

Another issue appeared when reviewing the approaches used. Two of them use the idea of presence, which can bring many drawbacks, like defining an acceptable window and the window split, that could implicate bias to the labeling. Excluding both approaches that use presence, the most common technique uses the idea of calculating a fixed value that is used to label the players and can be seen as the average behavior of the players, based on their absences. It does not suffer from the same drawbacks when presence is considered, but doubt regards using the same value for different players was raised. Bearing in mind that players' can have different behaviors among themselves, and a unique value could not be sufficient to represent them all, a novel approach that calculates individual values was proposed.

Still analyzing the methods used in the related works, it was identified that the churn definition is decided as the first step and never changes, which could be a problem, raising the RQ2: "Should the churn definition be updated?". Wishing to have a way to compare the labels produced by different definitions of churn, we proposed a method capable of quantitatively calculating the influence a change of churn definition has on the resulting labels. Using the proposed evaluation method, it was possible to achieve three conclusions. The datasets resulted in different results, implicating that a game can have players with different behaviors, and each approach, parameter, and decision when predicting churn should be specific to the game in question. By comparing the most common approach, Fixed Value (FV), with the authors' proposed Individual Fixed Value (IFV), it is possible to assume that it accomplished its objective by calculating individualized values for each player because the results showed that it captured more variance in the churn definition. It indicates that the FV approach cannot represent all the players' behaviors using a collective value, and the IFV does. Lastly, it was possible to observe that the values of CDCR can be used together with a user-defined threshold as an advice mechanism that alerts when a re-evaluation of the churn definition and the classifier's retraining should happen.

The second objective was to define a value that represents a player's performance in a match of the game League Of Legends (LOL), a Multiplayer Online Battle Arena (MOBA) game. The performance evaluation has various applications, like assistance to analysts in the competitive environment, churn prediction, Dynamic Difficult Adjustment of Artificial Intelligence Agents, etc. Even though numerous works had similar goals, only one could be used for this specific task, the one proposed by (PRAKANNOPPAKUN; SINTHUPINYO, 2016). Still, it lacks experiments and information regarding the method and the evaluation. To improve this method and solve the doubts encountered when developing it, the author of this dissertation: extracted and constructed a larger dataset of another game than the one used by Prakannoppakun and Sinthupinyo; explained in detail the method used; proposed another way to evaluate the performance of the method, and performed three experiments.

After the complete explanation of the method, three experiments were described. To assess each experiment, an evaluation based on the correct classification of the winner team was used. Since the winning team should have the bigger sum of the players' performance score, this evaluation can be seen as the correct estimate of the players' performance.

The first experiment was performed with the intent of fine-tuning the Artificial Neural Network to the dataset. The architecture found is discussed in detail and can be used by other authors as a baseline in their works. Next, an experiment was made to verify the model's performance when different game modes and maps are added to the training and testing datasets. This second experiment showed that the method could adapt to additional maps and game modes, which is an excellent feature if this method is deployed in other MOBA games or in a more general context. Finally, the last experiment aimed to find an appropriate value for the difference in performance between the two teams. The results revealed that the team's gold difference is not a good indicator, but the ratio of the other team's score showed excellent results.

The last objective regarded the churn prediction and the validation of the chosen metrics. The last experiment results proved that the Commitment can be used in MOBA games, and the social features improve the churn prediction classifiers' performance in a MOBA game. It also showed that the combination of the two concepts improved the classifiers' performance furthermore, as described by (PARK et al., 2017), surpassing State of the Art metrics and achieving the goal of developing a method capable of predicting churn.

This work encompasses the understanding and the solution to the problems encountered in the churn labeling task. Researchers from academia or industry can use it to improve their churn prediction systems, with more reliable labels and by defining churn individually. It can also be used to evaluate how much a churn definition change impacts the resulting labels, deciding when it is time to redefine it and retrain the classifier. Future works can evaluate and propose methods to define the most appropriate threshold for each case, improving the model's adaptability.

The approaches that use presence (Naive and Sliding Window) are not used as often as the ones that use a fixed value, but methods that can evaluate and recommend the best splitting, distance, and window size could be proposed, improving their usage and leading to new studies and methods capable of evaluating them. Aside from more deep studies of these two approaches, a comparison between the standard classification methods and another with the improvements proposed by this dissertation can bring important insights into the classifiers' convergence and performance. Lastly, the proposed labeling approach and evaluation method can be used in different domains to prove or disprove its applicability and importance in other areas that deal with the churn prediction challenge.

By improving the method proposed by (PRAKANNOPPAKUN; SINTHUPINYO, 2016), the goal of developing a method that outputs a value representing a players' performance in a MOBA game match quantitatively was achieved. In the process, numerous insights regarding the architecture of the method were found. Using the detailed explanation of the method, its characteristics, improvements, architecture, and evaluation process, other authors can calculate a player's performance in a match and use the value acquired in various applications. Even though an exhaustive search was made to find the most appropriate attributes and values for the method, other ideas could be found and tested, especially to the score difference. Future works can also utilize the advanced metrics proposed by (MAYMIN, 2020) in our method and compare them with the standard metrics. Aside from the parameters, other inputs could be used, as well as the test of automatic feature engineering using deep learning.

Considering the churn prediction using social features, studies could evaluate the hypothesis that social metrics' addition causes a more significant impact on the churn classifiers' performance on MOBA games than in other genres.

Although the performance of the metrics used surpassed the ones of the State of the Art articles in the systematic mapping, another factor that showed significant improvements is the consideration of the metrics as time-series, revealing the players' behavior through time. This consideration can be inserted into the proposed method in future works, as well as the addition of more social metrics. Lastly, other improvements could be made by the deployment of the survival analysis technique, predicting not only if a player will churn but when it is most probable that he/she will, and the usage of psychological features to label the players, changing the churn prediction approach from reactive to proactive.

Even though this work proposed a complete method for predicting churn, it can not be considered a definitive solution to churn prediction in games. The labeling procedure and, consequently, the churn prediction utilizes a reactive approach, which could implicate less successful retaining campaigns. The social metrics do not encompass all the social features used in related works, meaning they could be improved. The output of the method only returns if a player will or not stop playing, not when it will happen, which could be implemented using survival analysis. This change could provide valuable information for retaining campaigns. Lastly, other aspects could increase the model's performance, like psychological, audio, and visual aspects. These improvements and the possible future works listed can lead to a complete assessment of the players' actions, needs, and desires, which could improve the churn prediction task's assertiveness in games.

Bibliography

BACKIEL, Aimée; VERBINNEN, Yannick; BAESENS, Bart; CLAESKENS, Gerda. Combining local and social network classifiers to improve churn prediction. In: IEEE. *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).* [S.l.], 2015. p. 651–658.

BARAS, Dorit; RONEN, Amir; YOM-TOV, Elad. The effect of social affinity and predictive horizon on churn prediction using diffusion modeling. *Social Network Analysis and Mining*, Springer, v. 4, n. 1, p. 232, 2014.

BERTENS, Paul; GUITART, Anna; PERIÁÑEZ, África. Games and big data: A scalable multi-dimensional churn prediction model. In: IEEE. *2017 IEEE Conference on Computational Intelligence and Games (CIG).* [S.l.], 2017. p. 33–36.

BOGER, Zvi; GUTERMAN, Hugo. Knowledge extraction from artificial neural network models. In: IEEE. *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation.* [S.l.], 1997. v. 4, p. 3030–3035.

BORBORA, Zoheb. Computational analysis of churn in multiplayer online games. 2015.

BORBORA, Zoheb; SRIVASTAVA, Jaideep; HSU, Kuo-Wei; WILLIAMS, Dmitri. Churn prediction in mmorpgs using player motivation theories and an ensemble approach. In: IEEE. *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing.* [S.l.], 2011. p. 157–164.

BORBORA, Zoheb H; SRIVASTAVA, Jaideep. User behavior modelling approach for churn prediction in online games. In: IEEE. *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing.* [S.l.], 2012. p. 51–60.

BRADLEY, Ralph Allan; TERRY, Milton E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, JSTOR, v. 39, n. 3/4, p. 324–345, 1952.

BUCKINX, Wouter; POEL, Dirk Van den. Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual fmcg retail setting. *European journal of operational research*, Elsevier, v. 164, n. 1, p. 252–268, 2005.

CAPLAR, Neven; SUZNJEVIC, Mirko; MATIJASEVIC, Maja. Analysis of player's in-game performance vs rating: Case study of heroes of newerth. *arXiv preprint arXiv:1305.5189*, 2013.

CASTRO, Emiliano G; TSUZUKI, Marcos SG. Churn prediction in online games using players' login records: A frequency analysis approach. *IEEE Transactions on Computational Intelligence and AI in Games*, IEEE, v. 7, n. 3, p. 255–265, 2015.

CHEN, Vivian Hsueh-Hua; DUH, Henry Been-Lirn; PHUAH, Priscilla Siew Koon; LAM, Diana Zi Yan. Enjoyment or engagement? role of social interaction in playing massively mulitplayer online role-playing games (mmorpgs). In: SPRINGER. *International Conference on Entertainment Computing.* [S.l.], 2006. p. 262–267.

CHOLLET, Francois et al. *Keras*. GitHub, 2015. Disponível em: <https://github.com/fchollet/keras>.

CLARK, Oscar. *Games as a service: How free to play design can make better games*. London: Focal Press, 2014.

CLEMENTE-CÍSCAR, Mónica; MATÍAS, Susana San; GINER-BOSCH, Vicent. A methodology based on profitability criteria for defining the partial defection of customers in non-contractual settings. *European Journal of Operational Research*, Elsevier, v. 239, n. 1, p. 276–285, 2014.

COOK, Daniel. *The Circle of Life: An Analysis of the Game Product Lifecycle*. 2007. Disponível em: <https://www.gamasutra.com/view/feature/129880/the_circle_of_life_an_analysis_of_.php>.

DASGUPTA, Koustuv; SINGH, Rahul; VISWANATHAN, Balaji; CHAKRABORTY, Dipanjan; MUKHERJEA, Sougata; NANAVATI, Amit A; JOSHI, Anupam. Social ties and their relevance to churn in mobile telecom networks. In: *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*. [S.l.: s.n.], 2008. p. 668–677.

DRACHEN, Anders; LUNDQUIST, Eric Thurston; KUNG, Yungjen; RAO, Pranav Simha; KLABJAN, Diego; SIFA, Rafet; RUNGE, Julian. *Rapid prediction of player retention in free-to-play mobile games*. 2016.

DROFTINA, Uroš; ŠTULAR, Mitja; KOŠIR, Andrej. A diffusion model for churn prediction based on sociometric theory. *Advances in Data Analysis and Classification*, Springer, v. 9, n. 3, p. 341–365, 2015.

EL-NASR, Magy Seif; DRACHEN, Anders; CANOSSA, Alessandro. *Game analytics*. [S.l.]: Springer, 2016.

ELO, Arpad E. *The rating of chessplayers, past and present*. [S.l.]: Arco Pub., 1978.

FERGUSON, Mark; DEVLIN, Sam; KUDENKO, Daniel; WALKER, James Alfred. Player style clustering without game variables. In: *International Conference on the Foundations of Digital Games*. [S.l.: s.n.], 2020. p. 1–4.

GAJADHAR, Brian; KORT, Yvonne De; IJSSELSTEIJN, Wijnand. Influence of social setting on player experience of digital games. In: *CHI'08 extended abstracts on Human factors in computing systems*. [S.l.: s.n.], 2008. p. 3099–3104.

GLICKMAN, Mark E. The glicko system. *Boston University*, v. 16, 1995.

HADIJI, Fabian; SIFA, Rafet; DRACHEN, Anders; THURAU, Christian; KERSTING, Kristian; BAUCKHAGE, Christian. Predicting player churn in the wild. In: IEEE. *2014 IEEE Conference on Computational Intelligence and Games*. [S.l.], 2014. p. 1–8.

HEATON, Jeff. *AIFH, volume 3: deep learning and neural networks*. [S.l.]: Heaton Research, 2015.

HERBRICH, Ralf; MINKA, Tom; GRAEPEL, Thore. Trueskill$^{\text{TM}}$: a bayesian skill rating system. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2007. p. 569–576.

HODGE, Victoria J; DEVLIN, Sam Michael; SEPHTON, Nicholas John; BLOCK, Florian Oliver; COWLING, Peter Ivan; DRACHEN, Anders. Win prediction in multi-player esports: Live professional match prediction. *IEEE Transactions on Games*, York, 2019.

KARNSTEDT, Marcel; HENNESSY, Tara; CHAN, Jeffrey; HAYES, Conor. Churn in social networks: A discussion boards case study. In: IEEE. *2010 IEEE Second International Conference on Social Computing*. [S.l.], 2010. p. 233–240.

KAWALE, Jaya; PAL, Aditya; SRIVASTAVA, Jaideep. Churn prediction in mmorpgs: A social influence based approach. In: IEEE. *2009 International Conference on Computational Science and Engineering*. [S.l.], 2009. v. 4, p. 423–428.

KILIMCI, Zeynep Hilal; YÖRÜK, Hasan; AKYOKUS, Selim. Sentiment analysis based churn prediction in mobile games using word embedding models and deep learning algorithms. In: *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. Novi Sad: IEEE, 2020. p. 1–7.

KIM, Seungwook; CHOI, Daeyoung; LEE, Eunjung; RHEE, Wonjong. Churn prediction of mobile and online casual games using play log data. *PloS one*, Public Library of Science, v. 12, n. 7, 2017.

KRISTENSEN, Jeppe Theiss; BURELLI, Paolo. Combining sequential and aggregated data for churn prediction in casual freemium games. In: IEEE. *2019 IEEE Conference on Games (CoG)*. [S.l.], 2019. p. 1–8.

KUMMER, Luiz; NIEVOLA, Julio; PARAISO, Emerson. A key risk indicator for the game usage lifecycle. In: *The Thirtieth International Flairs Conference*. [S.l.: s.n.], 2017.

KUMMER, Luiz Bernardo Martins; NIEVOLA, Julio Cesar; PARAISO, Emerson Cabrera. Digital game usage lifecycle: a systematic literature review. In: *Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. Curitiba: SBC, 2017. p. 1163–1172.

KUMMER, Luiz Bernardo Martins; NIEVOLA, Júlio César; PARAISO, Emerson Cabrera. Applying commitment to churn and remaining players lifetime prediction. In: IEEE. *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. [S.l.], 2018. p. 1–8.

LEE, Yeng-Ting; CHEN, Kuan-Ta; CHENG, Yun-Maw; LEI, Chin-Laung. World of warcraft avatar history dataset. In: *Proceedings of the second annual ACM conference on Multimedia systems*. San Jose: ACM, 2011. p. 123–128.

LIU, Duen-Ren; LIAO, Hsiu-Yu; CHEN, Kuan-Yu; CHIU, Yi-Ling. Churn prediction and social neighbour influences for different types of user groups in virtual worlds. *Expert Systems*, Wiley Online Library, v. 36, n. 3, p. e12384, 2019.

MASAND, Brij; DATTA, Piew; MANI, Deepak R; LI, Bin. Champ: A prototype for automated cellular churn prediction. *Data Mining and Knowledge Discovery*, Springer, v. 3, n. 2, p. 219–225, 1999.

MAYMIN, Philip Z. Smart kills and worthless deaths: esports analytics for league of legends. *Journal of Quantitative Analysis in Sports*, De Gruyter, v. 1, n. ahead-of-print, 2020.

MILOŠEVIĆ, Miloš; ŽIVIĆ, Nenad; ANDJELKOVIĆ, Igor. Early churn prediction with personalized targeting in mobile social games. *Expert Systems with Applications*, Elsevier, v. 83, p. 326–332, 2017.

MORA-CANTALLOPS, Marçal; SICILIA, Miguel-Ángel. Moba games: A literature review. *Entertainment computing*, Elsevier, v. 26, p. 128–138, 2018.

MOZER, Michael C; WOLNIEWICZ, Richard H; GRIMES, David B; JOHNSON, Eric; KAUSHANSKY, Howard. Churn reduction in the wireless industry. In: *Advances in Neural Information Processing Systems*. Cambridge: ACM, 2000. p. 935–941.

ÓSKARSDÓTTIR, María; BRAVO, Cristián; VERBEKE, Wouter; SARRAUTE, Carlos; BAESENS, Bart; VANTHIENEN, Jan. A comparative study of social network classifiers for predicting churn in the telecommunication industry. In: IEEE. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. [S.l.], 2016. p. 1151–1158.

OU, Mingdong; CUI, Peng; PEI, Jian; ZHANG, Ziwei; ZHU, Wenwu. Asymmetric transitivity preserving graph embedding. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 1105–1114.

PARK, Kunwoo; CHA, Meeyoung; KWAK, Haewoon; CHEN, Kuan-Ta. Achievement and friends: key factors of player retention vary across player levels in online multiplayer games. In: *Proceedings of the 26th International Conference on World Wide Web Companion*. [S.l.: s.n.], 2017. p. 445–453.

PERIÁÑEZ, África; SAAS, Alain; GUITART, Anna; MAGNE, Colin. Churn prediction in mobile social games: Towards a complete assessment using survival ensembles. In: IEEE. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. [S.l.], 2016. p. 564–573.

PHADKE, Chitra; UZUNALIOGLU, Huseyin; MENDIRATTA, Veena B; KUSHNIR, Dan; DORAN, Derek. Prediction of subscriber churn using social network analysis. *Bell Labs Technical Journal*, Nokia Bell Labs, v. 17, n. 4, p. 63–76, 2013.

PIRKER, Johanna; RATTINGER, André; DRACHEN, Anders; SIFA, Rafet. Analyzing player networks in destiny. *Entertainment Computing*, Elsevier, v. 25, p. 71–83, 2018.

PRAKANNOPPAKUN, Noppon; SINTHUPINYO, Sukree. Skill rating method in multiplayer online battle arena. In: IEEE. *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. [S.l.], 2016. p. 1–6.

ROTHENBUEHLER, Pierangelo; RUNGE, Julian; GARCIN, Florent; FALTINGS, Boi. Hidden markov models for churn prediction. In: *2015 SAI Intelligent Systems Conference (IntelliSys)*. London: IEEE, 2015. p. 723–730.

ROTHMEIER, Karsten; PFLANZL, Nicolas; HÜLLMANN, Joschka; PREUSS, Mike. *Prediction of Player Churn and Disengagement Based on User Activity Data of a Freemium Online Strategy Game*. [S.l.]: IEEE, 2020.

RUNGE, Julian; GAO, Peng; GARCIN, Florent; FALTINGS, Boi. Churn prediction for high-value players in casual social games. In: IEEE. *2014 IEEE conference on Computational Intelligence and Games*. [S.l.], 2014. p. 1–8.

SARAVANAN, M; RAAJAA, GS Vijay. A graph-based churn prediction model for mobile telecom networks. In: SPRINGER. *International Conference on Advanced Data Mining and Applications*. [S.l.], 2012. p. 367–382.

SILVA, Mirna Paula; SILVA, Victor do Nascimento; CHAIMOWICZ, Luiz. Dynamic difficulty adjustment on moba games. *Entertainment Computing*, Elsevier, v. 18, p. 103–123, 2017.

SUZNJEVIC, Mirko; MATIJASEVIC, Maja; KONFIC, Jelena. Application context based algorithm for player skill evaluation in moba games. In: IEEE. *2015 International Workshop on Network and Systems Support for Games (NetGames)*. [S.l.], 2015. p. 1–6.

TAMADDONI, Ali; STAKHOVYCH, Stanislav; EWING, Michael. Comparing churn prediction techniques and assessing their performance: a contingent perspective. *Journal of service research*, SAGE Publications Sage CA: Los Angeles, CA, v. 19, n. 2, p. 123–141, 2016.

TAMASSIA, Marco; RAFFE, William; SIFA, Rafet; DRACHEN, Anders; ZAMBETTA, Fabio; HITCHENS, Michael. Predicting player churn in destiny: A hidden markov models approach to predicting player departure in a major online game. In: IEEE. *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. [S.l.], 2016. p. 1–8.

TSYMBALOV, Evgenii. Churn prediction for game industry based on cohort classification ensemble. 2016.

XIE, Hanting; DEVLIN, Sam; KUDENKO, Daniel; COWLING, Peter. Predicting player disengagement and first purchase with event-frequency based data representation. In: *2015 IEEE Conference on Computational Intelligence and Games (CIG)*. Tainan: IEEE, 2015. p. 230–237.

YANG, Wanshan; HUANG, Ting; ZENG, Junlin; YANG, Gemeng; CAI, Jintian; CHEN, Lijun; MISHRA, Shivakant; LIU, Youjian Eugene. Mining player in-game time spending regularity for churn prediction in free online games. In: IEEE. *2019 IEEE Conference on Games (CoG)*. [S.l.], 2019. p. 1–8.

YANG, Wanshan; HUANG, Ting; ZENG, Junlin; CHEN, Lijun; MISHRA, Shivakant; LIU, Youjian. Utilizing players' playtime records for churn prediction: Mining playtime regularity. *IEEE Transactions on Games*, IEEE, 2020.

YUAN, Sha; BAI, Shuotian; SONG, Mengmeng; ZHOU, Zhenyu. Customer churn prediction in the online new media platform: a case study on juzi entertainment. In: IEEE. *2017 International Conference on Platform Technology and Service (PlatCon)*. [S.l.], 2017. p. 1–5.

ZHENG, Angyu; CHEN, Liang; XIE, Fenfang; TAO, Jianrong; FAN, Changjie; ZHENG, Zibin. Keep you from leaving: Churn prediction in online games. In: SPRINGER. *International Conference on Database Systems for Advanced Applications*. [S.l.], 2020. p. 263–279.

ZHU, Ling; LI, Yifan; ZHAO, Guanshi. Exploring the online-game life cycle stages. In: IEEE. *E-Business and E-Government (ICEE), 2010 International Conference on*. [S.l.], 2010. p. 2436–2438.