July 2019

Phenomena of Social Dynamics in Online Games

Essa Alhazmi

University of South Florida, fattah200@gmail.com

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Computer Sciences Commons

Scholar Commons Citation

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.
Phenomena of Social Dynamics in Online Games

by

Essa Alhazmi

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Adriana Iamnitchi, Ph.D.
John Skvoretz, Ph.D.
Kingsley A. Reeves, Jr., Ph.D.
Yao Liu, Ph.D.
Paul Rosen, Ph.D.
Giovanni Luca Ciampaglia, PhD.

Date of Approval:
June 6, 2019

Keywords: social phenomena, social network analysis, data mining, team formation, migration

Copyright © 2019, Essa Alhazmi
DEDICATION

This dissertation is dedicated to my wife, Maryam, and our daughters, Jude and Elena.

I also dedicate this work to my parents, Zakia and Ali, and all my brothers and sisters.

أَهْنَئُهُمَا الْرِّسَالَةَ إِلَى
رَوْحَتِي مَجْهُورِ فَأَيْتِيَ جَوْدَ وَإِلَيْتَا
وَأَمي رَكْبِيَّ، وَأَليٌّ عَليٌّ
وَجَمِيع أَهْلِيٰ
ACKNOWLEDGMENTS

First, I would like to thank my advisor, Professor Adriana Iamnitchi, for advising and supporting me learn and practice foundations of scientific research. This thesis would not have been possible without her and without freedom and encouragement, she has given me during the time I spent at University of South Florida.

I would like to thank my PhD committee members for their feedback in the major area presentation and the final defense, especially Professor John Skvoretz and Professor Giovanni Luca Ciampaglia. Their time in contributing and discussing my research topics is really appreciated.

I would like to thank my research lab colleagues for their support and help. In particular, I would like to thank Nazim for his advice in academic writing, Sameera for the research ideas and thoughts, Anthony for his experience in video games and data collection, and Abhishek for his technical experience. I really appreciate their time and efforts.

I would like to thank the department staff, especially Gabriela Franco, for all support in my journey. In addition, I would like to appreciate and thank the chair of graduate affairs in the department Professor Yu Sun for his support. He was one of the reasons for me to join the graduate program and pursue the PhD.

Finally, I would like to acknowledge the partial support from NSF (grant: NSF IIS, 1546453) and full support from Jazan University in this journey.
# TABLE OF CONTENTS

LIST OF TABLES iv  
LIST OF FIGURES v  
ABSTRACT viii  

## CHAPTER 1 INTRODUCTION  
1.1 Online Gaming  
1.1.1 Team Formation  
1.1.2 Gamers Mobility  
1.2 Contributions  
1.3 Dissertation Outline  

## CHAPTER 2 BACKGROUND  
2.1 Online Games Studies  
2.2 Teams Formation Studies  
2.3 Migration Studies  

## CHAPTER 3 FRAMEWORK FOR COLLECTING AND PROCESSING GAMING DATA  
3.1 Online Gaming Environments  
3.2 Data Sources  
3.2.1 GameME Web Service  
3.2.2 Steam Platform  
3.3 Web Scraping Techniques  
3.4 The Crawling Process  
3.4.1 Collecting Player Activity  
3.4.2 Collecting Ban Data  
3.5 Design Considerations for the Implementation of the Crawler  
3.5.1 Load Balancing and Resource  
3.5.2 Attribute Patterns in the Web Source  
3.5.3 Data Attributes Generalization  
3.5.4 Data Storing  
3.6 Challenges in Collecting and Processing Data  
3.6.1 Data Collection Challenges  
3.6.1.1 Dynamic Observations  
3.6.1.2 API Limitations  
3.6.1.3 Crawling Failures
3.6.1.4 Storage Space Limitations 30
3.6.2 Data Processing Challenges 30
3.6.2.1 Inconsistent Data Types and Formats 31
3.6.2.2 Data Attributes Languages 32
3.6.2.3 Memory Size Limitations 32
3.6.2.4 Missing Values 33

CHAPTER 4 DATASETS 34
4.1 Co-play Activities 34
4.2 Friendship 36
4.3 Cheaters and Penalized Players 36

CHAPTER 5 TEAM FORMATION IN ONLINE GAMING 38
5.1 Dataset 39
5.1.1 Battlefield-4 39
5.1.2 Data Attributes 41
5.1.3 Data Characterization 42
5.2 Understanding Team Formation 45
5.2.1 Familiarity 45
5.2.2 Homophily 46
5.2.3 Competence 47
5.3 Squad Membership Prediction 49
5.3.1 Features and Categories 49
5.3.2 Model Evaluation and Results 50
5.4 Summary and Discussions 52

CHAPTER 6 SIGN PREDICTION IN ONLINE GAMING 53
6.1 Familiarity Determinants 54
6.1.1 Explicit Determinants 55
6.1.2 Implicit Determinants 55
6.2 Competence Determinants 56
6.2.1 Real Time Performance 56
6.2.2 Overall Performance 56
6.3 Similarity Determinants 57
6.3.1 Explicit Determinants 57
6.3.2 Node Similarity Determinants 57
6.4 Experimental Setup 58
6.5 Performance Evaluation 59
6.6 Model Generalization 60
6.7 Summary 61

CHAPTER 7 PLAYERS MOBILITY NETWORKS 62
7.1 Dataset 63
7.1.1 Game Server 64
7.1.2 Friendship Network 66
LIST OF TABLES

Table 4.1 Data description of co-play activities in 20 games observed between May 9, 2016 and July 29, 2016 (81 days). 35
Table 4.2 Data description of co-play activities in two Steam games observed from Feb 16, 2017 for 174 days for Counter-Strike: Global Offensive and 51 days for Team Fortress 2. 36
Table 5.1 Country of origin of the majority of players vs. server location. 44
Table 5.2 A comparison of five classifiers for squad prediction over homophily, competence, and familiarity as categories of features. 50
Table 6.1 Description of the features that are used in the sign predictive models. 59
Table 6.2 A comparison of five classifiers for signs prediction on the most popular server. 60
Table 6.3 Signs prediction results over all servers. 61
Table 7.1 Statistical description of servers’ lifespan and number of matches in CSGO and TF2. 64
Table 7.2 Characteristics of aggregated interaction networks and friendship networks. 65
Table 7.3 Basic statistics of mobility networks in each game. 72
Table 7.4 Features used in the pace (P) and growth (G) prediction tasks. 78
Table 7.5 Classification results for predicting popularity, defined by the number of followers, in the mobility networks for both games using Random Forest. 81
Table 7.6 Prediction performance demonstrated by both LSTM-based neural network and the Random Forest classifier in predicting the movement (adoption) pace. 82
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>A snap-shot of a server browser in TF2 that shows a list of servers and search options.</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>A snap-shot describes a clan of six members in a server recorded by GameMe in Battlefield-4.</td>
<td>17</td>
</tr>
<tr>
<td>3.3</td>
<td>A snap-shot of a game server in GameMe that hosts a match of CSGO game.</td>
<td>20</td>
</tr>
<tr>
<td>3.4</td>
<td>Illustration of crawling steps of active players and their attributes.</td>
<td>22</td>
</tr>
<tr>
<td>3.5</td>
<td>Illustration of crawling steps for retrieving a player's ban details.</td>
<td>26</td>
</tr>
<tr>
<td>5.1</td>
<td>The number of distinct Battlefield-4 players observed over the observation period (81 days).</td>
<td>40</td>
</tr>
<tr>
<td>5.2</td>
<td>CDF of the number of profiles per player.</td>
<td>41</td>
</tr>
<tr>
<td>5.3</td>
<td>CDF of squad size.</td>
<td>43</td>
</tr>
<tr>
<td>5.4</td>
<td>Distribution of the number of matches per player.</td>
<td>43</td>
</tr>
<tr>
<td>5.5</td>
<td>Average number of players per-hour, per-day of the week (starting Sunday) for Battlefield-4 (GMT time).</td>
<td>44</td>
</tr>
<tr>
<td>5.6</td>
<td>Frequency distribution of pairs of players over the games played on the same, different squad, and different team.</td>
<td>45</td>
</tr>
<tr>
<td>5.7</td>
<td>Number of co-team vs. co-squad for pairs of players.</td>
<td>46</td>
</tr>
<tr>
<td>5.8</td>
<td>Probability distribution of squad formation over the players in the same country and randomly assigned country.</td>
<td>46</td>
</tr>
<tr>
<td>5.9</td>
<td>CDF of squad diversity over the country of origin.</td>
<td>47</td>
</tr>
<tr>
<td>5.10</td>
<td>CDF of standard deviation of squad members’ ranks compared with the randomly generated squads, where the squad members (players) have distinct ranks and the squads in the random assignment follow the same size distribution as the real dataset.</td>
<td>48</td>
</tr>
</tbody>
</table>
Figure 5.11 Local predictive models compared to generalized predictive models in squad formation using random forest classifier on different frequencies of games, which includes same side and different side teams, for all pairs of players in the four most popular servers.

Figure 6.1 Illustration of data representation for the sign prediction.

Figure 6.2 Illustration of familiarity determinants in sign prediction.

Figure 6.3 Distribution of servers size.

Figure 7.1 Distribution of the number of games per player in both CSGO and TF2.

Figure 7.2 Distribution of the number of servers per player in both CSGO and TF2.

Figure 7.3 Flow of players' migration across servers in two games within Steam platform observed in circular tracks, where the inner circular track (lane) represents the source servers and the outer (peripheral) track represents the destination servers.

Figure 7.4 Number of pairs of players observed in the same games over the number of servers.

Figure 7.5 Probability distribution of players' movements to a new destination that followed at least one neighbor ($\rho$) vs. the minimum number of games played on the source.

Figure 7.6 Probability distribution of players' movements to a new destination that followed at least one neighbor ($\rho$) vs. delay of moving.

Figure 7.7 A temporal mobility scenario of two players $u$ and $v$ consists of four servers: $P, Q, R$, and $S$, and four different observation periods $\tau_1, \tau_2, \tau_3$, and $\tau_4$, where $\tau_1 \leq \tau_2 < \tau_3 < \tau_4$, and the graph at $\tau_1$ represents the interactions graph between players including $u$ and $v$.

Figure 7.8 Building a mobility network from time stamped edges found in the interaction networks.

Figure 7.9 CCDF of temporal mobility networks sizes denoted by the number of nodes.

Figure 7.10 Average weighted in-degree distribution of mobility networks.

Figure 7.11 Distribution of path sizes vs. the number of paths.

Figure 7.12 Probability distribution of movement delay by considering temporal paths in mobility networks.
Figure 7.13  Distribution of neighborhood adoption ratio of nodes ($P_{v_i}$) in the mobility network. 76
Figure 7.14  Sample of the interaction network with both adopters (red colored nodes) and non-adopters (green colored nodes). 77
Figure 7.15  Ranking of feature importance in predicting the popularity in CSGO (left) and TF2 (right). 80
Figure 7.16  Feature importance by Spearman’s rank correlation coefficient between the predicted outcome and the ground-truth in accurately predicting the adoption pace. 83
ABSTRACT

Online communities exhibit dynamic social phenomena that, if understood, can both influence the design of technical platforms and inform theories about general social dynamics. With increasing popularity, online games provide a rich recording of social dynamics that can contribute to understanding human behavior. This dissertation studies two phenomena of social dynamics at large scale using data traces from online games. The first phenomenon is team formation and the second is players mobility between gaming servers.

This dissertation first presents a framework for collecting data from online gaming through crawling. It includes the data sources and the tools used for data collection and processing. We developed a web crawler to perform longitudinal data collection. We discussed primary design considerations and challenges in data collection that we encountered in this effort.

We examined several hypotheses about team formation using a large, longitudinal dataset from a team-based online gaming environment. Specifically, we tested how positive familiarity, homophily, and competence determine team formation in Battlefield-4, a popular team-based game in which players choose one of two competing teams to play on. Our dataset covers over two months of in-game interactions between over 380,000 players. We showed that familiarity is an important factor in team formation, while homophily is not. Competence affects team formation in more nuanced ways: players with similarly high competence team up repeatedly, but large variation in competence discourages repeated interactions.

In addition, we formulated the team formation behaviors into a sign prediction problem. We classified interactions in online team-based games into different classes. Then, we modeled two predictionsign prediction scenarios: teams versus squads. We extracted time-based features from these determinants to show that our determinants are effective in predicting
signs between gamers, indicating that prior interactions between gamers (familiarity) are more likely to accurately predict signs.

Finally, we presented a data-driven study focused on characterizing and predicting the mobility of players between gaming servers in two popular online games, *Team Fortress 2*, and *Counter Strike: Global Offensive*. Understanding these patterns of mobility between gaming servers is important for addressing challenges related to scaling popular online platforms such as server provisioning, traffic redirection in case of server failure, and game promotion. We built predictive models for the growth and the pace of player mobility between gaming servers. We showed that the most influential factors in predicting the pace and growth of migration are related to the number of in-game interactions. Declared friendship relationships in the online social network, on the other hand, do not affect predicting mobility patterns.

Besides providing a large-scale, empirical based understanding of social phenomena, our work can be used as a basis to a variety of application scenarios including peer, team, group, and server recommendations in gaming platforms.
CHAPTER 1

INTRODUCTION

The exponential increase of structured and unstructured social data provides insights into a range of phenomena of social dynamics. However, working with enormous amounts of data is challenging and requires computational techniques and tools, and “big data” methodologies to advance social sciences. Therefore, the field of computational social sciences as a new field of research includes many areas and topics that were previously beyond the realm of scientific investigation in human and social dynamics [1], including automated information extraction, social network analysis, complexity modeling, and social simulations models. Lazer et al. [2] define computational social sciences as an emerging field that “leverages the capacity to collect and analyze data at a scale that may reveal patterns of individual and group behaviors”. Shah et al. [3] determine computational social sciences field as a method to social inquiry characterized by four properties: 1) large volume of datasets with complex structure, 2) datasets derived from social and digital media sources and other electronic databases, 3) the use of computational tools and “big data” methodologies to generate patterns, and 4) applicability in social theories.

Due to the increased usage of social media for sharing, learning, and interacting, many studies [4, 5, 6, 7, 8] leveraged social media data and encompassed a variety of techniques from different fields such as business bio-sciences, and social science. Researchers have been trying to understand different dynamic social phenomena in the online spaces. By leveraging social network analysis, some studies focus on understanding groups evolution. Backstrom et al. [9] study how properties of the underlying social networks influence group formation and evolution in large social networks, while Zhang et al. [10] study not only how group grows
over time but also how groups decay, and propose a dynamic model for group evolution. Moreover, other studies tried to understand migratory behaviors and patterns in online space. For example, Kumar et al. [11] analyzed the migration of user attention between seven popular social media sites. In addition, Yan et al. [12] investigated reasons of users migratory behaviors of video consumption across different content providers.

The data from online gaming platform are found to be less explored in studying social phenomena compared to other platforms in online space. The surge in gaming population has an important economic and social implication. Despite their distinctiveness from other online social platforms, researchers believe that online gaming environments should be considered as a legitimate source of human interactions [13]. Therefore, rich competitive and social characteristics of gamers are proved to be useful in understanding other human and social behaviors such as social relationships [14], social interactions [15], social presence [16], cheating [17], and toxic behaviors [18].

My research addresses two collective behaviors demonstrated by players in an online gaming environment: team formation and players’ mobility. This dissertation provides an attempt to better understand the behavior of individuals and groups in the competitive environments.

1.1 Online Gaming

Online gaming is not only a multi-billion dollar industry [19] entertaining a large global population but also a popular form of social interaction among millions of individuals.

Online games are offered on different platforms, and sometimes games could have different versions for different platforms. Many games are offered on two platforms: Steam and Origin. Steam, developed by Valve Corporation, and the Origin platform, developed by Electronic Arts Inc., are the common digital distribution platforms for online games. Each platform provides services and software for players to buy, install, and play games.
Many online games are team-based where one of two teams attempts to accomplish some goals. Different works have studied behavioral dynamics of team-based online games for different types of games such as role-playing games like World of Warcraft [20, 21], battle arena games like League of Legends [22, 23], and first-person shooter games like Counter-Strike: Global Offensive [24].

There are two high-level methods of team assignment in online games: 1) matchmaking and 2) player choice. Matchmaking systems construct teams in such a way as to provide equal chance for each team to win. Matchmaking systems are typically used by popular eSports games like the League of Legends and Dota 2 and in the competitive mode of Counter-Strike. Player choice systems provide players with many more options depending on the particular server (population) on which they play, what team they want to join, and sometimes even down to a smaller “squad”-level unit, as for example, in the Battlefield series. While games with matchmaking are undoubtedly interesting and useful for studying the effects of team formation, they are unsuitable for understanding how humans choose to be on a team. Thus, our focus in this dissertation is on games with player choice systems.

In this dissertation, through social network properties and machine learning techniques, we quantify and model two phenomena of social dynamics in online games to better understand the behavior of individuals and groups in the shared social space in large-scale communities. This chapter provides background on the problems and methodology this dissertation addresses. It breaks into two parts: Teams Formation and Players Mobility in online gaming environments.

1.1.1 Team Formation

The first phenomenon of social dynamics we discuss in this dissertation is team formation. Teams are ubiquitous in society, both in offline and online environments. The main question we address in this part of the dissertation is how gamers form teams in online space. Understanding what people value in their teammates in a competitive environment may potentially lead to understanding what makes the team successful. This understanding can
have practical applications for human resource management in all its facets, from hiring to project management. Existing studies focus on contexts as diverse as online education [25], geographically distributed software development projects [26], and online gaming [27, 28].

In this dissertation, we aim to understand the interplay among three factors that impact team formation in online team-based games: 1) positive familiarity, 2) homophily, and 3) competence. We focus on how these factors shape an individual’s choice of team in two-team competitive first-person shooter games.

To understand this phenomenon of social dynamics, we raise the following research questions:

- Do players choose to team up with others who are similar to themselves (homophily)?
- Do players choose to team up with others who have skills or positions that contribute to the team’s success?
- Do players choose to team up with those with whom they have played in the past?
- Can we build predictive models to identify in-game and in-team interactions between gamers automatically?

1.1.2 Gamers Mobility

The second phenomenon that this dissertation studies in the context of online gaming is mobility. The main interest of our research in this part on the dissertation is to understand the structural properties and patterns in the evolution of mobility networks in online games.

Understanding players’ mobility between gaming servers is important for multiple technical problems, including server provisioning, traffic redirection in case of server failure, and game promotion. In addition, the migratory patterns of players can be leveraged in modeling information dissemination or behavior adoption. For example, a player may introduce a new set of gimmicks or may affect the server culture via positive or toxic social behavior.
In the real world, human mobility is a socially-embedded phenomenon [29], which is affected by both socioeconomic factors and the subjectivity of human behaviors [30]. Two important factors have been observed to contribute to an individual’s migration decision [31]. The first is the extent to which a migrant is connected to communities at home and at their destination. The second is the strength and the support of destination ties in providing access to resources available in the destination environment (e.g., job information). However, the online gaming environment has different characteristics, and it is unclear whether the same arguments apply to player mobility.

This dissertation quantifies the importance of in-game interactions for a player’s decision to migrate from one server to another within the same game. Players move to different servers over time due to various reasons, including technical performance (latency, computation speed), server/game preferences, peer familiarity, or personal endorsements. We specifically focus on social interactions as a factor in players’ mobility patterns. We develop machine learning-based models to predict (1) the popularity of players over time with respect to the number of neighbors following their mobility patterns, and (2) how fast a player moves between servers relative to other players. To understand this phenomenon of social dynamics, we raise the following research questions:

- Do in-game interactions affect players mobility across servers?
- Do gamers’ friendships influence their mobility across servers?
- Can we predict the pace and the growth of mobility networks in online games?

1.2 Contributions

The dissertation makes the following research contributions:

- It provides a practical framework for collecting large-scale longitudinal co-play activities of gamers, including their friends and their bans. This framework can be used specifically to retrieve data for 20 games. The dataset is to be made publicly available after proper
anonymization. Moreover, this dataset is intended to serve the objectives of the NSF proposal\(^1\), as it records dynamic processes embedded in social networks. It is likely that this dataset will enable more research in the space of dynamic processes in online networks.

- It proves empirically that the team formation process is influenced by some common factors in both online and offline environments. Specifically, the results provide evidence consistent with the factors identified by Hinds et al. \([32]\) as important in team formation: familiarity, homophily, and competence. To our knowledge, this study is the first large-scale quantitative study on team formation in the online gaming community based on the combination of familiarity, homophily, and competence.

- It formulates the problem of team formation as a sign prediction problem. This formulation enables the temporal prediction of teams and squads (mini-teams) in the online games we studied. We discovered that dynamic topological features of the gamers’ interaction networks are highly contributing to the performance of sign and, implicitly, to the prediction performance of team and squad signs.

- It models another dynamic process, namely user mobility across servers. We identified the features relevant to the prediction of players’ popularity, including early and late movers in the temporal mobility networks. We showed empirically that the growth and the pace of mobility can be predicted. The results show that co-players influence players’ mobility decisions via the number of interactions and not via declared friendships.

1.3 Dissertation Outline

The remainder of this dissertation is organized into eight chapters, including the Introduction. Chapter 2 presents the background about the phenomena of social dynamics and reviews the related work of this dissertation. Our framework of data collection and

\(^1\)Structural Anonymization Techniques for Large, Labeled, and Dynamic Social Graphs, NSF, IIS, 1546453
processing is presented in Chapter 3. The dataset characteristics are introduced in Chapter 4. In Chapter 5, we perform an empirical analysis of team formation as the first phenomenon of social dynamics\textsuperscript{2}. We formulate the team formation problem as a sign prediction task to build and evaluate different predictive models in Chapter 6. Chapter 7 presents mobility as a second phenomenon of social dynamics and models it as a mobility network\textsuperscript{3}. Finally, we conclude and discuss our findings and future directions in Chapter 8.

\textsuperscript{2}The work in the chapter was first published in [33] and permission is included in Appendix A.
\textsuperscript{3}The work in the chapter was first published in [34] and permission is included in Appendix A.
CHAPTER 2

BACKGROUND

This chapter presents an overview of previous research that this work builds upon.

2.1 Online Games Studies

Online gaming is not only a multi-billion dollar industry entertaining a large global population, but also a popular form of social interactions among millions of individuals [17]. Studies in online games are twofold: technological and behavioral [35]. Our focus in this work is limited to the behavioral study of gamers. McEwan et al. [13] believe that online gaming environment should be considered as a legitimate source of human interactions and may support many types of sociability such as social interactions (shared activity) and permanence (long-term associations).

As online gaming exercises different types of sociability, it becomes a rich source of temporal social interaction data that can be exploited for many computational social science questions [14]. While many researches about social and group phenomena in virtual environments were covered in the review by Sivunen and Hakonen [36], many studies in online games have been introduced in the review by Mora-Cantallops et al. [37]. The authors pointed to many opportunities of unexplored research areas.

Due to their competitive nature, temporal interactions from online gaming environments can provide insight into the dynamics of online friendship networks and empirical temporal graph patterns [38]. Recent studies also analyzed social dynamics and temporal processes in online gaming. For example, in addition to inferring the shape of a massive network from a popular online game, Merritt and Clauset [14] dynamically analyzed the inferred network to
understand the friendship evolution in a gaming environment. In conjunction to player dynamics in massively-multiplayer online gaming (MMOG) environment, Zhuang et al. [39] characterized the temporal variation of players’ participation, session length, down-times, inter-arrival times, availability, player churn rate, the degree of player independence, and visiting time in different locations. They identify several predictors of session length and find inter-arrival times are inversely proportional to the population size.

Social networks emerged from online gaming environments have been addressed in recent studies. Blackburn and Iamnitchi suggested that linking between interactions and declared friendships of gamers in the interaction-backed OSN could provide more meaningful relationships than the declared friendships alone [40]. As mentioned by Balint et al. [41], an adversarial context, denoting the competitive nature of the players within the game, is held responsible for the evolution of social gaming networks emerged in the online gaming environment. Jia et al. [15] introduced a number of graph-based models in multiplayer online games representing gamers relationship (friendship) and a variety of types of interactions such as Same Match, Same Match, Same Side Match, and Opposite Side Match. These proposed models were compared and characterized by different graph metrics.

Due to its competitive nature, unethical behaviors (e.g., cheating and toxicity) are common in an online gaming environment. Blackburn et al. [17] study relationship between cheating status and friendships. They find that cheaters in online gaming are well embedded in the social network (friendship), and they tend to associate with each other more than non-cheaters. To understand and quantify the factors that lead players to adopt cheating behaviors, Zuo et al. [42] empirically verified hypothesis derived from in-lab psychology experiments to explain how cheating spreads via social gaming interactions. In a similar approach, Kwak et al. empirically tested several hypotheses drawn from sociology and psychology theories explaining toxic behaviors such as cyberbullying.
2.2 Teams Formation Studies

Online environments offer great potential for systematically exploring the factors that impact team formation because of the fine granularity recordings of digitally-mediated interactions. The factors that make individuals choose particular teammates were studied in contexts as diverse as education [25], geographically-distributed software development projects [26], and online gaming [27, 28], based on surveys, observations, or digital records of interactions. Three factors commonly studied are positive familiarity, defined as the existence of previous positive experiences, competence (represented as expertise or reputation), and homophily. Familiarity was shown to have an impact both on team formation and on team performance. In online gaming communities, Hudson et al. [16] showed that familiarity and team trust are positively correlated and they improve team performance. Waddell and Peng [43] showed that positive familiarity leads to repeated play, which leads to friendship. Mason and Clauset [44] found that players perform better when they play with friends, and individual performance is independent of team performance. In addition, it has been observed that players tend to be more ambitious in games when they have good cooperation with friends [45]. Good cooperation within the team leads to better performance [46] and is a stronger motivator than competition [47].

In geographically-distributed software teams [26], where the challenge is coordination among team members, the performance was found to depend on two independent factors: competence (defined as familiarity with the task) and familiarity with the other members in the team. When team familiarity is weak, competence was shown to improve performance significantly. These results confirm studies in offline environments that also show that familiarity is a factor in team formation and performance. For example, cooperation among participants in social care institutions in the Netherlands was found to grow with familiarity [48] and led to a higher success rate.

Competence is an intuitive factor in team formation, yet it has been studied more as it relates to team success. In online gaming, Kim et al. [49] studied team congruency vs. individual
proficiency in League of Legends (LoL), and discovered that individual proficiency has a bigger influence to team success. However, because LoL uses match-making algorithms for forming teams, competence could not be evaluated in the context of team formation. Also, the individual choices are influenced by the selection of experts as group members (competence) and the positive past working experience with others (familiarity) [50].

Homophily was shown to be a relevant factor in team formation and success. In a recent study done on Massive Open Online Courses (MOOC), Eftekhar et al. [25] found that age, education level, distance and time zones are factors for students to form successful teams. Moreover, multidisciplinary teams with more diverse skill sets were more successful than the rest. Kamel et al. [51] viewed homophily as a distance function between different user skills and proposed it as a strategy for forming software development teams.

Few studies considered all three categories. By observing a group of students forming 33 small teams over four years, Hinds et al. [32] show how competence, familiarity, and homophily affect team formation. Positive familiarity is shown to be correlated with competence: the better reputation among their peers a student has, the more often he/she will be included in the same team. The study also shows how homophily plays a role in team formation: students show a strong preference for working with colleagues of the same race, but less so for the same gender. At the same time, homophily is the weakest factor.

Ruef et al. [52] studied a sample of 816 organizational founding teams and used structural event analysis to predict the number of entrepreneurial teams. The study shows that homophily and the existence of a previous relationship (friendship or family ties) are important factors to predict team composition, while spatial and geographical proximity are not. Familiarity, competence, and homophily are thus the main categories that were studied as determinants for team formation and performance. However, each context represents each of these categories by a different set of environment-specific variables.
2.3 Migration Studies

In real world, human mobility has been shown to be a socially embedded phenomenon [29], which is affected by both socio-economic factors and the subjectivity of human behavior [30]. Two important factors [31] are seen to contribute towards an individual’s migration decision: first, the extent to which a migrant is related and/or connected to communities at home and at the destination; and second, the strength of destination ties in providing access to supportive and rewarding destination environments (e.g., job information). Quantitative studies of human mobility patterns are relevant to applications such as estimating migratory flows, traffic engineering, urban planning, emergency management, and epidemics control [53].

Migration scholars use social network concepts to comprehend the dynamics [54]. Recent research has emerged with novel insights in the human migratory model. For example, by analyzing the switching behaviors of populations across different social networking sites, Hou et al. [55] identified that low socializing and entertainment, alternative attractiveness, and peer influence are the dominant catalysts that trigger migration decision among humans.

Migration studies are recently employed in online social networks for understanding their users’ migratory behaviors among different online social and content-sharing platforms [55, 12]. Kumar et al. [56] defined two types of migration and analyzed the movement of user attention between seven popular social media sites. Newell et al. [57] analyzed how and why users migrate from Reddit to Reddit-like alternative platforms.

Depending on the incorporation of the social network in the mobility models, Karamshuk et al. [58] categorized mobility models into two classes: (i) real trace-based and (ii) social-aware models. Consequently, a research effort in the literature, through exploring the traces of movements captured on real users, can be found that characterizes both the spatiotemporal (users’ movements across locations) and social (duration of interactions or interval between subsequent interactions) aspects of human mobility. However, a comparative analysis of these works demonstrates different results, even on some basic features of human mobility (e.g., fat-tailed vs. exponential distribution of trip length) [59]. Stressing on the awareness of
understanding human mobility patterns, Jahromi et al. [60] pointed out that human mobility is a prominent form of social aggregation that determines how a specific social network would form and grow. According to Szell et al. [61], this discrepancy results due to the underlying context differences from where the mobility patterns are inferred. Considering the massively multiplayer online game environment as a ‘socioeconomic petri dish’, the authors studied gamers’ mobility in a completely controlled way (i.e., having complete access to practically all actions including movements accumulated over the years) to discover the intricate interplay of spatial constraints, social and economic factors, and patterns of mobility.

In the absence of such complete information, given many unknown factors influencing a population’s mobility patterns, human trajectories are often approximated by diffusion models [62]. Further, Han and Wang [63] suggested that human mobility pattern is an important issue for understanding the human interaction behavior, for studying the spreading of epidemic and information, and for optimizing traffic systems. Likewise, Barbosa et al. [30] stressed that a quantitative study of human mobility ought to answer relevant questions, such as, what factors determine the decision of movement and the choice of the destination, and to what extent is human mobility predictable (features related to mobility prediction).

In online gaming context, studies have concentrated on different aspects of human social behaviors in online environments including individual and team performance [64], expert’s behavior [65], homophily [66] and team formation [33]. The migratory process and mobility pattern (movement of players’ from one server to the other) is an important trend found in players’ behavior. Motivated by these facts mentioned above, and considering the uniqueness of two underlying social networks (i.e., friendship and interaction) that emerge in online games, this dissertation developed a model to understand the migratory patterns of players in the gaming environment.
CHAPTER 3

FRAMEWORK FOR COLLECTING AND PROCESSING GAMING DATA

This chapter presents our process for retrieving information from online gaming through crawling. This includes the data sources and the tools used for data collection and processing. Also, it discusses some basic considerations regarding the data collection process to be considered for future related efforts. Finally, the chapter describes some challenges regarding data collection and processing.

3.1 Online Gaming Environments

Online games are categorized into different genres. Elliott et al. [67] introduce 15 genres of video games. For example, a massively multiplayer online role-playing game is a type of games in which players can collaborate and compete with others in the online shared world, while other types of games have only a single player and the success depends on developing characters with skills. First-person shooter (FPS) games are another type of game in which players can “kill” others. In real-time strategy games, players join a strategic combat-oriented game with no wait between moves. In contrast, players in turn-based strategy games wait for others to act. Many games are team-based. As we stated before, our focus is on the player choice system of team-based games which provide players with many options depending on the particular server (population) on which they play, and what team they want to join.

To select their servers and their teams, gamers have different ways to do so. In Team Fortress 2 (TF2) specifically, players use “server browser” to search from a list of available servers and choose a map based on multiple options. Figure 3.1 shows an example of a server browser and search options such as a list of available servers, specific servers that were added
as favorites, a list of the recent servers a player has visited, a list of which servers a player’s friends are currently on, and others. When a player joins a server, he/she has four options to select a team: 1) Random/Auto Assign, 2) Spectator, 3) BLU, or 4) RED. In Battlefield-4, players can invite others to join a squad in a team. Thus, online games interfaces provide options for gamers to select servers as well as what team or squad they want to join.

There are many shared attributes among team-based games that we are collecting. In the following, we describe them in order to explain our process of data collection.

- Player Connection and Game Server:

Player choice systems are usually built around a “server browser.” As the name suggests, server browsers allow players to discover available servers by providing information like the map that is being played, players that are actively playing on the server, and perhaps the scores of these players. Unlike matchmaking systems, players are free to connect to whichever server they want, thus the server browser presents the first point at which players are making a choice regarding team formation. Whenever a player
selects a server and connect, two attributes specify where a player plays and what time he joins. Connection time is represented using a Unix timestamp, and the IP address and port number define the game server.

- **Maps and Rounds:**
  Most multiplayer games take place on one or more maps. A map is a simulated environment in which the game takes place. Multiple maps are typically available for each game. Maps vary not only in their layout and visual design, but also in the goals that players must meet. For example, one map might be a simple “death match”, where the winning team is the one that kills more opposing players, while another map might be “point control”, where the winning team is able to capture and control certain areas on the map. Maps are usually rotated after some win condition or a time limit is reached. Because some maps are asymmetric (i.e., one team attacks and the other defends), many maps are broken down further into rounds.

  We use match and map synonymously in this dissertation, i.e., a match takes place on a single map, has a beginning and an end, and is usually broken up further into rounds in some games. The attributes that are relevant to the map and rounds include the map name, map start time (Unix timestamp), map country, and the total number of rounds for the player in the map since connecting.

- **Clans, Teams, and Squads:**
  Gamers usually engage and establish social and competitive groups. Clans, squads, and teams are the most common types of groups formed by players. Players join or belong to clans independently of the active games, while squads and teams must be formed during game play.

  Clans are an informal group of players created under their initiative, their size may have just a few players, or they could have hundreds [68]. Clans could reflect different social and competitive aspects, such as identities, a form of internal organization, nationality,
and professional characteristics [69, 70, 71]. Some players maybe use clans to form their off-line networks to reduce the risk of cooperating with strangers in games [70]. A clan tag usually appears either as a prefix or suffix strings added to a player’s screen name in closed square brackets (see Figure 3.2 for an example).

Teams are competitive groups where players often compete in online tournaments, maps, or sessions for benefits. They are also much smaller than clans with specific size limits, depending on the game. Players join maps and choose their teams based on their roles or classes. Players can change their teams and sometimes become spectators while watching the game.

In some games, a squad is a smaller group of players that is on the same team. Squads have members of different roles in a game. For example, Battlefield-4 contains five players: one leader and four other members. Some games employ “classes”, where certain abilities and weapons are restricted to each class. For example, engineer, assault,
medic, support, and recon are classes in Battlefield games. Each one of these classes dictates a specific role in the game with a specific type of weapons such as sniping and healing. Thus, it is often wise to build a squad from a diverse set of classes to ensure that any challenges can be met. For example, it is a good idea for a squad to include not only assault classes, but also a medic class to heal and revive teammates, as well as an engineer class to repair any vehicles the squad might come across.

- **Skills and Ranks:**
  Skills are calculated using different variables and depend on the game. For example, gamers in Battlefield-4 discussed in multiple forums\(^1\)\(^2\) how to calculate skills. They found skills are calculated by a combination of scores per minute (SPM), kills per match (KPM), and kill to death ratio (KDR). Ranks reflect the competency level and position of a player among other players within a game server. In general, games use different rating systems [72] such as the Ingo, Glicko, and Elo rating systems. The skills and ranks change based on players’ activities and performance. Thus, the skill change is an attribute that reflects the change in skills since the player joined a new map compared to the last map he joined.

- **Wins and Losses:**
  While ranks and skills reflect the player performance among others, win and loss points reflect the performance of that player’s team since joining the map.

- **Server Ban:**
  A server ban is a ban that is given by a server administrator who, in most cases, is the server owner. The server admin or owner can also choose the length of the ban.

- **Other general attributes:**
  Some attributes that are reflecting players’ performance are numeric, e.g., total kills,

---

\(^1\)http://battlelog.battlefield.com/bf4/forum/threadview/2955064772055477261/
\(^2\)http://battlelog.battlefield.com/bf4/forum/threadview/2955065248394286030/
deaths, suicides, headshots, and shots, while others are binary, e.g., survived, Assists, Assisted, team kills, and team killed. There are a few string attributes, e.g., favorite weapon and map name.

In the player profile in a game server, there are many attributes represented in a cumulative form of in-game attributes, such as total wins, total losses, playing duration, total maps, and weapons in all matches in the game server. Awards, roles, aliases, actions, and targets are other attributes represented in the player profile in the form of a list that includes some details like name and time.

Many of these attributes describing gaming environments were collected in our framework.

3.2 Data Sources

Two data sources are being used to collect data analyzed in this work: the GameME web service (gameme.com) and the Steam platform (https://steampowered.com). Both sources contain attributes that describe gaming environments from different aspects.

3.2.1 GameME Web Service

GameME is a third-party service that monitors real-time playing activities for approximately 20 team-based online games on many gaming servers globally. It provides real-time statistics on players’ scores for the monitored games. Figure 3.3 shows a screen-shot of a game server that hosts a match of two teams in Counter-Strike:Global Offensive (CSGO) game. The figure shows players that are playing and their attributes in the game. It also shows the spectators.

GameME provides APIs to access two different views of the player population: a global view that ranks players globally overall games monitored, and a local, server-specific view that ranks players on the local game server and provides more detailed information on player profiles. The local view, accessible through a GameMe client API, provides detailed
Figure 3.3: A snapshot of a game server in GameMe that hosts a match of CSGO game.

information regarding the current match played on each game server. This information includes attributes such as the name of the map played, the start time of the match, players in each team and their squad membership, skill change in a match, and current rank on the server. The client API also provides access to on-server player profiles that include information such as historical information (e.g., achievements, rewards, and statistics on team membership), weapons owned and their usage, clan membership, the reported location (country), and aliases. Also, GameME records players’ ban details by server admins for specific game servers.

3.2.2 Steam Platform

Steam is one of the largest digital distribution platforms for online games developed by Valve Corporation. Steam provides multiple services. It provides a system for players to
buy, install, and play games of different environments such as a team-based, cooperative, multiplayer, and single-player gaming environments. Players in the Steam platform can be members of the Steam Community and form social circles using Community profiles within the platform. The Community option allows users to join game groups, form clans and friendships, and chat in-game, and each player with a Steam community profile can set up his/her profile as public or private. Also, Steam provides API services to extract from Steam Community a players’ list of friends and games owned by them, including network and game statistics for the most recent 48 hours. Different player-specific information, including activities over the previous two weeks, and games owned by them can be extracted through the APIs provided by Valve Corporation.

3.3 Web Scraping Techniques

Web scraping is a method of data extraction from a web source, such as HTML. In general, a web scraping process includes essential steps. First, web scraping requires specifying a URL or list of URLs that include the targeted data. Most high-level languages have built-in or standard library support for reading URL sources. Second, it is important for the coders to identify patterns in targeted data fields and their contents in the sources. After finding data field patterns, a regular expression is a useful tool for selecting data fields among texts in the sources.

In our case, HTML and XML are two types of sources from which we parsed data fields and contents. Different methods for parsing sources have been introduced in the literature. We used a Python regular expression module, re, to extract the contents of the data we targeted. The module provides regular expression matching operations similar to those found in Perl. For further details about the re module, the Python web site provides a full documentation of the regular expression operations.

https://developer.valvesoftware.com/wiki/Steam_Web_API
https://docs.python.org/3/library/re.html
The data of interest for collection are arranged in different ways in the sources. Some information is embedded in a single tag, and others are embedded in nested tags. Also, some contents are represented in a list and embedded in repeated tags. We built different methods for parsing each type of contents using the regular expression technique. These methods use only two objects of the regular expression: `search` and `findAll`. Different symbols were used in a regular expression to extract different attributes in the content from the source. Some of them start with digits and others contain special characters.

### 3.4 The Crawling Process

In the previous section, we presented the techniques of extracting web content by crawling. In our framework for collecting gaming data, two crawling processes have been used. One task is to collect players’ activities in matches with their attributes including their friendship from Steam, while the other task is to collect players’ ban records.

---

5HTML code that defines every structure on the HTML page, including texts, images, and links.
3.4.1 Collecting Player Activity

In this crawling process, we collected data relevant to a specific game in real-time and at two levels. The first one targets the user (player) level, while the other one targets the community (server) level. At the user level, the crawler collects players’ information, including players’ activities in matches and players’ profile information in the servers. In addition, the crawler collects friendship information for players in STEAM games. At the community level, the crawling process aims to gather information reflecting multiplayer activities in servers and the server life span, including total teams and players.

In our framework, 20 games (characterized in Chapter 4 are distributed according to their volumes and popularity for load balancing on slave-nodes (machines) to avoid traffic in the network and lag during the crawling process. Figure 3.4 shows seven consequent steps that the crawler must fulfill. The details of the tasks are explained as follows:

- Scheduler Setup:
  While the crawler spins over GameME contents to collect data, the scheduler controls the crawl time based on the previous crawling time span. Different games have different peak activity and off-peak times. For popular games, the crawler took up to 50 minutes to record information about all active game servers at the peak time, and under a minute for less popular games. This will show an unbalanced time of observations of gaming servers. In other words, the time gaps between consecutive observations will not be identical, and it will affect our analyses. Hence, these variations in the periodicity of crawling require the scheduler to avoid bias in data collection. For example, the scheduler will add a time delay for the crawler to start a new round if the time span is short.

- Extracting Game Pages:
  GameME shows a list of 20 games. We extracted the current number of pages that record running game servers using the GameME URL and specifying the name of the targeted game. The total number of pages was embedded in the source of GameME. These pages
record the list of running matches. We ran parallel jobs for these pages by scraping the source to extract the number of available pages. It is important for programmers to identify the code segments that require parallel processing in a crawling task.

- Extracting List of Matches:
  From the previous task, we stated that every page records a list of matches/game servers. The IP address and port number represent every match. Our target in this step is to obtain the IP address and port number to use them in the client API and retrieve game server details. The client API requires a client address that records some game servers and player profiles. To scrape the IP address with the port number and the client, we identified the pattern for all matches in the HTML source and used a regular expression to extract them.

- Using GameME game-server API:
  The GameME game server API requires three inputs to retrieve details regarding active matches: the IP address, port number, and GameME client. These inputs were extracted in the previous task. These were input in the GameME game server API query. The results from this API query are presented in an XML page that shows information on players who are joining the game and their attributes in the game, including IDs, game name, teams, skills, connection time, and others reflecting in-team behavior. This information about a player in a game is collected by scrap ing the XML content using a regular expression.

- Using GameME Player API:
  The GameME player API requires three inputs to get the details about the players’ profile in the game server: the player’s Steam ID, GameME client, and game name as it is recorded in the game server API. These inputs were extracted in the previous step. These were input in the GameME player API query. The response from this API query is an XML page that shows profile details about a player in the game server, including
a history of actions, maps, achievements, roles, teams, awards, weapons, and others reflecting the player's profile in a game server. This information regarding a player in a game server is collected by scraping the XML content using a regular expression.

- Using Steam API:
  The Steam API has different methods that return results in different formats. One of these formats is JSON. A player’s list of friends was collected from the Steam API. The method of friendship details in Steam API requires a community ID with the API key, and it returns a friend list containing a friend’s community ID and the time the friendship was formed. We converted a Steam ID to a community ID using an algorithm provided by Valve Corporation\(^6\). The community ID was input in the Steam API query to return the friend list in JSON format. This friend list information was collected for the player that was observed in the game server.

- Storing Content in MongoDB:
  MongoDB is a document database and stores data in JSON-like documents. Each document reflects a player’s details in three types of information: details about a player in an on-going match, details on a player's profile, and a player’s friend list. Each one of these types of information is represented in one collection in the database. The player information is stored in a Python dictionary after being scraped from the source. A Python dictionary can be inserted in a MongoDB collection by specifying the key and its corresponding entry or list of entries. We stored the player document in the first collection called `real_time_match` using a key that reflects a player in a current match. Each key is linked to the following entries: Steam ID, IP address and port number, game name, team name, map start time and crawl time. The second collection is called `player_profile`, which stores a key corresponding to player information in a game server. The key contains two indices: Steam ID and IP address with the port number.

\(^6\)https://developer.valvesoftware.com/wiki/SteamID
The third collection is called `player_friend_list` for storing a document of a player’s friends. The key here only contains a player’s Steam ID.

### 3.4.2 Collecting Ban Data

This crawling process collects data relevant to players who were banned or were penalized on active game servers. The data includes a player’s Steam ID, the start time of the ban, expiration time of the ban, ban reason, and some other details. Figure 3.5 shows the crawling steps used to obtain this information. The player ban crawler is similar to the player activities crawling process in the first three steps. From GameMe client, the crawler extracts a URL of web source for players’ ban list. Some GameME clients do not monitor players’ bans or penalties. In the fourth step, the client ban web source shows the total current pages of the players’ ban list. We ran parallel jobs for every page to speed up the performance of the crawling process. In the fifth step, each page contains a list of penalized players and
their details. We scraped a players’ list using the regular expression and prepared a Python
dictionary containing ban information for every player to be represented as a document in
the database collection. Finally, we insert each player dictionary as a single document in the
MongoDB collection, named player bans. The key for this database collection contains the
Steam ID and ban time. Both indices guarantee to prevent the inclusion of duplicates in the
database documents while recording multiple distinct bans for a single player.

3.5 Design Considerations for the Implementation of the Crawler

In order to crawl data from any source, some general considerations should be taken into
account. We present in this section five aspects of the crawling task.

3.5.1 Load Balancing and Resource

With a limited amount of resources, e.g., few machines, collecting data requires determining
the best optimization approach for collecting a representative Utilizing the available resources
in a right way for data collection might require multiple experiments to identify the size of
data. For example in our case, the 20 games throughout five nodes (machines are configured
in master-slave architecture) should not be randomly targeted during the crawling process.
Our objective here is to utilize the machines in a way that mitigates network traffic and
crawling lag. In the case with an unfair distribution of games, some machine could be idle
during some hours of the day. Therefore, we run the crawler in identical spins for each game
in different hours, and we compared the data volume among games. The volume distribution
of 20 games was ranked from high to low volume, and five games were assigned to each
machine. The high-rank games and low-rank games were set in the same machine.

3.5.2 Attribute Patterns in the Web Source

Another consideration in data crawling requires identifying an accurate pattern of the data
attributes when scraping source with a regular expression, especially HTML. This requires
checking patterns of attributes that are embedded in a source multiple times for multiple games. Moreover, the regular expression must be implemented in a way that can accurately capture all cases in the contents. For example, if the symbol `&` is embedded in a pattern of the HTML source, it must be encoded to `&amp;` in order to capture that pattern accurately. Looking at the source in a web browser is not enough to determine the actually encoded symbols. In addition, the contents in the source might not be identical in all games. For example, the URL of ban sources can have different patterns, such as `banlist.<client>` or `<client>/banlist`. The process of crawling might fail when scraping only the GameME client. It is important to dig into the content and find the actual URL for the ban list.

### 3.5.3 Data Attributes Generalization

Generalizing the data attributes during crawling is important for data processing such that these processes are appropriate for all games. Games can differ in their attributes. For example, some games record squad attributes, such as the Battlefield games series, while Steam games do not. Adding `None` to unavailable or missing attributes would simplify the task in the data processing.

### 3.5.4 Data Storing

Finally, a document that was already inserted in the database collection might be stored again during the crawling process. This can cause a huge amount of unneeded data to accumulate. Therefore, a database key collection document/record with a meaningful index or indices was created to prevent this redundant accumulation of data. As explained in the previous section, the key should prevent the storage of duplicate documents and should simplify data retrieval from the database.
3.6 Challenges in Collecting and Processing Data

Some challenges in data crawling and processing are examined in this section. The challenges are sometimes technical or functional.

3.6.1 Data Collection Challenges

Two data collection tasks were presented in the previous section. Both tasks include scraping from web sources and accessing data through an API. Four challenges in both tasks of data collection are discussed here.

3.6.1.1 Dynamic Observations

One challenge regards the dynamic nature of the contents, which might arise in any scraping process. In our case, the number of matches and pages that monitors active games in GameME changes over time. When scraping these contents, the span time of the crawler in each round is not fixed. This can bias the periodicity of the crawler across servers at peak times during game activities. When processing data, the granularity of the observation time is inconsistent. In order to prevent this, a scheduling mechanism was used to control the crawl time based on the previous crawling time span.

3.6.1.2 API Limitations

Another challenge is the limitations of the API. These limitations on data access could sometimes complicate the data collection process. Different web services provide an API for different data access tasks while limiting the number of requests over a specific time interval. The Steam API limit clients to 100,000 calls per day. In the case of reaching a limit on the total number of calls, multiple API keys must be used to avoid restrictions on data access.
3.6.1.3 Crawling Failures

Crawling failure is a common problem that arises due to many issues, such as connection error, DNS error, server error, or URL error. Using a local machine as a resource for crawling is not a practical approach for long-term data collection. A local machine will mostly encounter connection failures. Instead, cloud services with multiple instances or nodes, such as Amazon Web Services (AWS), should be used. Parallel jobs in threads with a large number of requests would appear like a DDoS attack or spidering activities. This would block the IP address of the machine that attempts to execute a crawling process. To avoid that, the number of parallel requests from the same web domain should be reduced. Another cause of failure in crawling is related to the content of the data. In many cases, missing data or changes in data types are common errors. Exception handling for different types of errors should be considered to prevent crawling failure.

3.6.1.4 Storage Space Limitations

Lack of availability of resources, such as the number of machines, storage capacity, and high-speed internet access, can be major challenges in any data collection process. In our case, players’ activities in different gaming servers, including details (more than 70 attributes) of their profiles, were monitored over long periods. Our dataset reached more than 500 GB. The crawling process would fail due to limited storage space. Similarly, slow speed internet can cause delays in the crawling process, leading to a not representative sample set.

3.6.2 Data Processing Challenges

Some challenges regarding data processing and some tactics to overcome them are presented here. These challenges arise when processing any large dataset or parsed data.
3.6.2.1 Inconsistent Data Types and Formats

One of the common issues involved in processing data is the inconsistency of the data formats and data types. For example, different types of players' IDs were found in Steam games. IDs beginning with Steam string are called steamIDs, a newer format for player IDs starts with “[U_” and ends with “]” is called steamID3, and steam community IDs with 17 digits are called steamID64.

Each type of ID can be converted to other types of IDs. All formats were found in the raw data. In order to generalize to a single type of ID, the type of ID was initially identified using a regular expression. Each ID was then labeled by its type. Finally, each ID was converted to a steamID64 based on its label. This process avoids inconsistency in player IDs and provides more precise statistics on player activities in games. Another example related to data inconsistency is in the date formats of penalty data, which includes the start time of the ban, expiration time of the ban, and ban duration. Our data shows that dates are formatted based on geographic region and server admins. Date formats start either with a month (if the game server is in North or South America) or with the day (if the game server is in Europe). Some dates use a hyphen between the date components, while others use a slash.

Searching for date formats is a challenge. Three steps were used to fix and generalize an accurate date format. First, all existing date formats were identified and regular expressions reflecting them were prepared. This step could be done using a brute force method to search for a pattern. When the pattern is found, it could be saved and all records with that pattern could be removed. Then, a new pattern is searched until no record is found.

More than nine date formats were found in our data. Nine patterns are not enough to fix dates because some formats are vague when the day and month are less than 12. In this case, records that are not clear without words were labeled VAGUE so that they could be fixed. The second step for fixing VAGUE dates is to assign clear date formats to the ban client URL (the source that records player bans). This reduces the search space of the VAGUE dates and changes most of the ban dates. However, few ban clients record all ban dates with VAGUE,
including the start and expiration dates of the ban. Here, another factor is required to fix the remaining dates. In the last step, the remaining date formats were fixed based on the duration of the bans. If the ban duration matches the difference between the start time and end time of the ban in a specific format, then this format is correct. Although this step would address the remaining \textsc{vague} dates formats, the ban duration must also be fixed to a general format, such as seconds (\texttt{int} or \texttt{double}). The duration formats are seen in texts (\texttt{string}) with different time units, such as minutes, hours, weeks, or months. They also show different abbreviations for time units. To fix this, all time unit abbreviations were extracted and labeled to generalized time units. A regular expression was used to extract the number that preceded the time units. It allows the ban duration to be generalized into seconds.

3.6.2.2 Data Attributes Languages

Processing data attributes or fields in different languages is challenging. When collecting data on penalties, the raw data shows many fields in different languages. All fields were prepared in a dictionary structure, and all non-English fields were translated into English. Approximately 67 fields were extracted and generalized into 15 fields. This process is infeasible if there are many data fields.

3.6.2.3 Memory Size Limitations

Another challenge in processing large datasets is the limited amount of memory. The processed data can be represented in different structures, such as a graph, dictionary, list, or frames. We usually followed the chunking strategy with multi-threading tasks to speed up processing and avoid memory limitation errors. However, this strategy may be infeasible if the data must be fully processed, such as in a graph.
3.6.2.4 Missing Values

Treating missing data is a necessary task in some problems. There are two types of missing data or values, and each type must be treated differently. One type of missing values depends on the actual configuration of data, such as private values. Missing values are meaningful and removing them might bias the results.

The other type of missing values is independent of the actual data. For example, some of players’ data do not provide either STEAM id and their skills. They could be omitted randomly due to traffic or failure in the data collection. One common strategy to address these missing data is to predict and replace them. However, deciding how to treat missing values depends on the context and objective of the task surrounding this data.
CHAPTER 4

DATASETS

Online games have become a rich source of playing activities. As early of 2018 Steam platform averaged 47 million daily active users and 90 million monthly active users [75]. Different services have been launched to monitor and track live activities of gamers such as GameTracker, Twitch, and GameMe.

In our study, three major data sets are considered and collected by Steam Community and GameMe. The first datasets contain a significant amount of data on the individual or multiplayer activities, collected in three durations: 51, 81, and 174 days. The second dataset contains online friendship networks for players observed in the first dataset. The third dataset contains information about auto-bans and admin-bans. The auto-bans were collected by the Valve Anti-Cheat System (VAC ban). The admin-bans were collected from GameMe.

These datasets have many detailed attributes described in chapter 3. In our work, we focused only on three games: Battlefield-4, Counter-Strike: Global Offensive, and Team Fortress 2. However, we have collected the dataset not only for these three games but also for 17 other games that can be used for future studies.

4.1 Co-play Activities

The implementation of the data collection framework described in Chapter 3 was initially used to crawl player activities across 20 games between May 9th and July 29th, 2016 (81 days). The crawler collected live activities of matches and profile details of gamers who were observed co-playing live. Table 4.1 provides the numbers of records, players, profiles per users, servers, and bots players in all 20 games. 16 games are from the Steam platform, and the remaining
Table 4.1: Data description of co-play activities in 20 games observed between May 9, 2016 and July 29, 2016 (81 days).

<table>
<thead>
<tr>
<th>Games</th>
<th>Records</th>
<th>Players</th>
<th>Profiles</th>
<th>Servers</th>
<th>Bots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter-Strike: Global Offensive</td>
<td>3,738,695</td>
<td>746,779</td>
<td>1,112,749</td>
<td>718</td>
<td>4,001</td>
</tr>
<tr>
<td>Battlefield-4</td>
<td>2,375,176</td>
<td>233,151</td>
<td>308,621</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>Battlefield Hardline</td>
<td>1,484,461</td>
<td>65,112</td>
<td>81,567</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td>Counter - Strike 1.6</td>
<td>1,367,201</td>
<td>315,134</td>
<td>336,160</td>
<td>81</td>
<td>748</td>
</tr>
<tr>
<td>Counter-Strike: Source</td>
<td>1,204,419</td>
<td>127,669</td>
<td>153,807</td>
<td>120</td>
<td>5,936</td>
</tr>
<tr>
<td>Day of Defeat: Source</td>
<td>1,086,597</td>
<td>42,219</td>
<td>67,135</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>Battlefield 3</td>
<td>989,921</td>
<td>96,668</td>
<td>114,853</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>Counter-Strike: Condition Zero</td>
<td>784,097</td>
<td>38,184</td>
<td>43,715</td>
<td>19</td>
<td>231</td>
</tr>
<tr>
<td>Battlefield: Bad Company 2</td>
<td>689,312</td>
<td>46,895</td>
<td>56,060</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>Day of Defeat 1.3</td>
<td>644,474</td>
<td>9,830</td>
<td>13,081</td>
<td>8</td>
<td>433</td>
</tr>
<tr>
<td>Insurgency 2014</td>
<td>539,342</td>
<td>88,230</td>
<td>119,216</td>
<td>52</td>
<td>1,750</td>
</tr>
<tr>
<td>Team Fortress 2</td>
<td>533,786</td>
<td>85,850</td>
<td>97,294</td>
<td>53</td>
<td>264</td>
</tr>
<tr>
<td>Team Fortress Classic</td>
<td>499,800</td>
<td>5,524</td>
<td>6,387</td>
<td>7</td>
<td>283</td>
</tr>
<tr>
<td>Left 4 Dead 2</td>
<td>406,225</td>
<td>31,159</td>
<td>35,506</td>
<td>30</td>
<td>711</td>
</tr>
<tr>
<td>Nuclear Dawn</td>
<td>335,101</td>
<td>3,656</td>
<td>4,745</td>
<td>4</td>
<td>128</td>
</tr>
<tr>
<td>Half-Life 2: Deathmatch</td>
<td>273,035</td>
<td>17,192</td>
<td>22,974</td>
<td>22</td>
<td>410</td>
</tr>
<tr>
<td>Zombie Panic! Source</td>
<td>263,405</td>
<td>11,004</td>
<td>13,309</td>
<td>2</td>
<td>396</td>
</tr>
<tr>
<td>Deathmatch</td>
<td>111,879</td>
<td>1,438</td>
<td>1,459</td>
<td>3</td>
<td>37</td>
</tr>
<tr>
<td>Fistful of Frags</td>
<td>72,420</td>
<td>8,124</td>
<td>8,293</td>
<td>4</td>
<td>109</td>
</tr>
<tr>
<td>Left 4 Dead</td>
<td>1,004</td>
<td>444</td>
<td>466</td>
<td>3</td>
<td>31</td>
</tr>
</tbody>
</table>

four Battlefield games are from Origin. Bot players are easily identifiable in Steam games by their Steam IDs, but we could not identify them in Battlefield games.

For some games, we found that the popularity of games is consistent with the official statistics on the Steam platform. For example, on the Steam platform, Counter-Strike: Global Offensive (CSGO) and Team Fortress 2 (TF2) is regularly from the top 10 popular games in a number of active players. They ranked as the third and sixth most popular games, respectively\(^1\).

The second crawling process collected data on Feb 16, 2017, and targeted two popular Steam games: CSGO and TF2. We avoided Bot account and collected additional in-game properties related to players and servers, such as friendships and server bans. Table 4.1 describes the details of the co-play activities for both games. CSGO shows more than 1.62

\(^1\)http://store.Steampowered.com/stats/
Table 4.2: Data description of co-play activities in two Steam games observed from Feb 16, 2017 for 174 days for Counter-Strike: Global Offensive and 51 days for Team Fortress 2.

<table>
<thead>
<tr>
<th>Games</th>
<th>Records</th>
<th>Players</th>
<th>Profiles</th>
<th>Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter-Strike: Global Offensive</td>
<td>13,998,841</td>
<td>1,620,493</td>
<td>3,345,944</td>
<td>934</td>
</tr>
<tr>
<td>Team Fortress 2</td>
<td>2,749,963</td>
<td>231,373</td>
<td>403,548</td>
<td>341</td>
</tr>
</tbody>
</table>

million players and 13 million records collected over 174 days. TF2 shows approximately 231 thousand players and 2 million records.

From these datasets, we conducted our study in two social phenomena in the next chapters: team formation and gamers’ mobility. Over this co-play data, we were able to generate interaction networks with different types of edges such as the same game, same team, and same squad. These networks are dynamically changing and evolving with weighted edges that reflect the intensity of interactions between gamers. Moreover, we extracted the migration flows of the gamers between servers. Given the scale of the data collected, we believe these datasets are a representative sample that can lead to reliable analysis.

4.2 Friendship

The crawler collected friendship lists for each player observed in co-play activities. The friendship list includes Steam IDs and the time the friendship was identified. From this data, we generated friendship networks between gamers. We observed more than 62.19 million players forming about 591 million edges. One of the characteristics of this network is that the network is very slow to change. Therefore, we assume the friendship network is static in this dissertation.

4.3 Cheaters and Penalized Players

There are two types of bans in Steam games: auto-bans and admin-bans. The auto-bans are the bans that are conducted by the VAC-ban system [76], which permanently bans players
for cheating. The admin-bans are the bans that are given by a server administrator, who also determines the length of the ban.

Cheating data was collected by a research colleague using the Steam API. The data shows the Steam ID and the time in which the 846,398 cheaters were caught. We also collected banned players by admin using a web crawler over GameME web clients, as explained in Chapter 3. The ban data contains the Steam ID of a banned player, start time of the ban, the expiration time or duration of the ban, ban reason, and other details. The data also shows more than 400 thousand bans for about 290 thousand players. Players are banned for specific reasons, such as bad behavior, cheating, breaking playing or server rules, a vote by server members, or spamming.
Teams are ubiquitous in society, both in offline and online environments. Understanding what people value in their teammates in a competitive environment may potentially lead to understanding what makes team successful. This understanding can have practical applications to the management of human resources in all its facets, from hiring to project management. Online environments offer great potential for systematically exploring the factors that impact team formation because of the fine granularity recordings of digitally-mediated interactions.

The objective of this chapter is to understand the interplay among three factors that impact team formation in online team-based games: 1) positive familiarity, 2) homophily, and 3) competence.

Positive familiarity is the positive past performance with a teammate, which may translate into incentives to team up again in the future. Negative familiarity could also occur, and in this case negative past performance with a teammate translates into a disincentive to team up again. Similarity is the sociological principle of homophily. The principle of homophily suggests that people seek out others of similar socio-demographic background for interaction. Finally, competence is the skill known to contribute to success in a team.

We focus on how these factors shape an individual’s choice of a team in two-team competitive first-person shooter games. We investigate how these factors affect team formation via observations of 60,410 “Battlefield-4” matches, played by 384,066 distinct players, on 63 servers, located in 7 different countries. The Battlefield series is one of the most popular first-person shooter (FPS) multiplayer franchises in the world. Battlefield-4 is designed to support up to 64 players (32 on each team) by default, more than twice the number of players
supported by other popular games, such as Call of Duty or Counter-Strike. By choosing to focus on this game environment, we are able to examine team formation not just at the team level, but also at “squad” level, which is small teams of up to 5 players who can coordinate more tightly. Overall, we find that familiarity is an important factor in team formation, while similarity is not. Further, we discover that competence affects team formation in more nuanced ways: highly skilled players tend to team-up repeatedly, while large variations in competence discourage repeated interactions.

5.1 Dataset

In contrast with previous studies that looked at optional teaming up for task-specific objectives [77], we focused on team-based games, described below, where players must choose one team (out of typically two teams) to play. We collected data by observing GameMe and focused on one highly popular game, Battlefield-4, because of its popularity and its use of squads as mini-teams.

5.1.1 Battlefield-4

Battlefield-4 is a very popular multiplayer first-person shooter game (FPS), played on a simulation of the modern battlefield, and has features like a wide variety of weapons, maps, vehicles, and destructible terrain. The Battlefield series is known for large maps with a great number of players: Battlefield is designed around having over twice the number of players per match than typical FPSs. This focus on large-scale games has resulted in some features that make Battlefield an excellent source for studying team formation. By its very nature, teams in Battlefield are generally too large (up to 32 players on each team by default) to operate as a single unit. To alleviate this, the developers included the squad concept.

A squad is a smaller group of players that are on the same team. Squads can be up to 5 players: 1 leader and 4 other members. Battlefield-4 employs “classes” where certain abilities and weapons are restricted to each class. Thus, it is often wise to build a squad from a diverse
Figure 5.1: The number of distinct Battlefield-4 players observed over the observation period (81 days).

set of classes to ensure that any challenges can be met. For example, it is a good idea for a squad to include not only assault classes, but also a medic class to heal and revive teammates as well as an engineer class to repair any vehicles the squad might come across.

Battlefield squads have concrete advantages. First, the squad leader is able to issue orders using an in-game interface, such as attack or defend a given objective. When squad members successfully execute these orders, bonus points are received. Next, there are certain squad “specializations” that act as bonuses for the entire squad and are unlocked as the squad performs well throughout a match. Squads are also given their own chat channel (text and voice) which allows for increased communication. Finally, squad members can “respawn” (come back to life after dying) on top of any of their squad members instead of at a predefined spawn point. This allows fast redeployment to action.

Our data shows that Battlefield-4 has 359,810 distinct players played in 48,018 matches on 63 servers located in seven countries. The maximum team size in our dataset is 66 players. Figure 5.1 shows the number of players observed in Battlefield-4 over time. The gaps in crawling (about 7 days total) are due to temporary failures in data collection.
5.1.2 Data Attributes

Players have multiple profiles, one associated with each Battlefield game server they played on. Players are identifiable by their globally unique Battlefield ID, yet have a local (server-based) player ID on each server they access. Our dataset for Battlefield-4 shows that about 60% of players have only one profile, less than 37% of players have between two and five profiles. The maximum number of profiles per player in our dataset is 24. Figure 5.2 shows the distribution on the number of profiles per player. Typical attributes used for identifying homophily such as gender, education, race, age, are not reported in the player profiles. Moreover, we know that gender is often misrepresented in profiles of online gamers, due to toxicity and harassment [78, 79]. For homophily, we identified two potentially relevant attributes: country of origin and clan. The country of origin is declared by the player. About 9% of the players do not report a country of origin in any of their server profiles while about 0.22% of the players declare multiple countries (with a maximum of 4) in their different profiles. However, more than 90% of players declare one country of origin, so we concluded that this is a meaningful data attribute for our analysis.

The clan is a free text tag that the player can choose to associate with other players. Unlike guilds [28] in Massively Multiplayer Online Games (such as War of Warcraft), clan membership has no influence on playing mechanics, but has only identification purposes.
In Battlefield, as it turns out from our observations, clan membership does not have much importance on player identity, either: few players declare their clan they are belonging to. Over 82% of the players do not declare a clan in their profile, less than 16% belong to one clan, and the remaining declare membership to more than one clan. We thus ignored this attribute from our analysis.

Many attributes in player profiles measure aspects of competence over time, such as number of kills, number of kills per minute, number of deaths, etc. However, all these measures can be uniquely represented by the server-based rank of a player, which is computed by sorting the players in decreasing order based on the number of points accumulated. Players start with a fixed amount of (1000) points and acquire more when they “kill” opponents, and lose points when they are “killed”. The number of points lost or gained depend both on the opponent’s relative competence level (such that, for example, being killed by a newbie is more costly than being killed by an advanced player) and on the weapon used (such that, for example, killing using a rifle gains fewer points than killing by knife). Players can thus have different ranks on the different servers they play. In our analysis, we used the rank of the player on the particular server on which we observed an interaction, as reported in that server’s player profile at the time of the observation.

Two crucial attributes are useful for studying team membership. The first is the membership to one of two teams, typically called “US Army” and “Chinese Army” or “Russian Army”. The second is a membership to squads, with predefined names such as Alpha, Bravo, Charlie, Delta.

5.1.3 Data Characterization

We ended up with 359,810 distinct players who played in 48,018 matches on 63 servers located in seven countries. These players collectively report coming from 173 countries across the globe, spanning all continents. Over the observation period, we counted 657,238 squads with at least 2 members each, and 6.5% of the observed squads were formed of only one player.
As visible in Figure 5.3, the vast majority of the squads are of 5 or fewer players, but we notice (a very small percentage of) squads of unexpected size (up to 14 players). We suspect these are due to unusual server configurations that allow such large squads. We observed that the majority player population is from the US (Table 5.1). Surprisingly, the players from the country where the server is located do not always make the majority population on that server. For example, German players are the majority on UK, Netherlands, and Poland servers. However, geographical proximity, as expected, plays an important role in the players’ choice of servers (due to latency considerations): servers attract the majority population from the same continent.

One of the pre-requisites for any dataset used to understand team formation is that it provides plentiful samples of the same player making a decision on which team to join. To
Figure 5.5: Average number of players per-hour, per-day of the week (starting Sunday) for Battlefield-4 (GMT time).

that end, Figure 5.4 plots the distribution of the number of games we observed for each player. The distribution is quite heavy-tailed: while the mean and median number of games played are 7.36 and 3, about 0.62% of players played over 100 games, while the most active player participating in 650 games. Overall, the figure provides evidence that we have plentiful data to understand player preferences with team formation.

Figure 5.5 plots the distribution of the average number of active players observed per hour of the day for each day of the week (GMT timezone). We see expected patterns both in terms of which days and times the players are the most active: players are more active on non-workdays, with peaks occurring in the evenings (biased towards the US time zones) and troughs during typical working/school hours. Although these results are to be expected, they serve to validate our collection methodology.

<table>
<thead>
<tr>
<th>Server location (###) (% population)</th>
<th>First Majority</th>
<th>Second Majority</th>
<th>Third Majority</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States (215,015) (62.75%)</td>
<td>United States (65.03%)</td>
<td>Canada (10.99%)</td>
<td>Brazil (5.03%)</td>
</tr>
<tr>
<td>Germany (69,636) (20.32%)</td>
<td>Germany (19.11%)</td>
<td>Poland (13.57%)</td>
<td>Russian Federation (12.74%)</td>
</tr>
<tr>
<td>United Kingdom (29,293) (8.55%)</td>
<td>Germany (14.16%)</td>
<td>United Kingdom (13.34%)</td>
<td>Russian Federation (10.27%)</td>
</tr>
<tr>
<td>Australia (11,418) (3.33%)</td>
<td>Australia (83.31%)</td>
<td>New Zealand (16.23%)</td>
<td>New Caledonia (0.25%)</td>
</tr>
<tr>
<td>Brazil (10,317) (3.01%)</td>
<td>Brazil (83.76%)</td>
<td>Argentina (8.25%)</td>
<td>Chile (265%)</td>
</tr>
<tr>
<td>Netherlands (6,174) (1.80%)</td>
<td>Germany (15.02%)</td>
<td>United Kingdom (12.81%)</td>
<td>United States (7.14%)</td>
</tr>
<tr>
<td>Poland (212) (0.61%)</td>
<td>Poland (20.59%)</td>
<td>Poland (19.84%)</td>
<td>Russian Federation (10.82%)</td>
</tr>
</tbody>
</table>
5.2 Understanding Team Formation

In order to understand team formation, we focus on analyzing the formation of teams and squads, for reasons already mentioned. We structure our analysis of the three factors for team formation presented before familiarity, homophily, and competence.

5.2.1 Familiarity

Familiarity affects the choice to join a particular team or squad under the intuition that positive past experiences with another player will result in teaming up in the future. Similarly, negative past experiences with another player will result in a disincentive of teaming up again. However, the question remains whether familiarity has any significant effect with respect to team formation in online gaming environments. Unlike the real world, players of online video games typically have millions of other players to play with.

Figure 5.6 plots the distribution of pairs of players with respect to the number of times they were on the same squad, different squads, and different teams. In general, we see a preference towards familiarity: players become increasingly likely to be on the same squad as the number of games played together increases. Conversely, it becomes increasingly unlikely that “familiar” players will choose to be on different squads and teams. Recall that there are concrete benefits to being in the same squad, and this is likely one of the reasons why players who become more familiar with each other tend to play on the same squad repeatedly as

Figure 5.6: Frequency distribution of pairs of players over the games played on the same, different squad, and different team.
shown in Figure 5.7. Once they have played enough together on the same team and start to understand each others’ play style, it only makes sense to join up together. The players can earn more points and get squad perks, as well as having their own private communication channel. (Note that a squad is by definition part of the same team.)

5.2.2 Homophily

Homophily, or the tendency for similar people to form relationships, was shown to have a role in the formation of student teams for class projects [32]. In the gaming context, homophily with respect to the country of origin is intuitive due to shared language and solidarity against a shared political enemy. Moreover, cultural characteristics were shown to play a role in the choice of people to play [80]. We thus look at the shared country of origin for members of the same squad or the same team. First, we analyze the distribution of pairs of players who played on the same team.
Table 5.1 shows server locations and the top three countries of players that play on them. Generally speaking, we can see that servers are populated by players from “near by” countries. Figure 5.8 compares the probability distribution of squad formation for players from the same country with a random distribution, which assigns players to squads based on the population of player origins. From the figure, we see that the smaller squads are more likely than larger ones to be composed of players all from the same country. If homophily with respect to the country of origin were a major factor in team formation, we would expect the number of larger squads with only a single country to be heavily biased towards the majority country of the population. However, we see that two-thirds of squads are formed by at least two countries (Figure 5.9).

5.2.3 Competence

The theory behind competence and team formation is that players will naturally gravitate towards players that are “good” at the game. This makes intuitive sense: everyone wants to be on a team with a winner. However, competency with respect to team formation is a double-edged sword: although everyone might want to team up with the best player, that best player wants to team up with good players, not just any random person. Moreover, the notoriously toxic and unforgiving environment of online gaming makes it potentially unpleasant to play with much better players: the less competent player risks nasty comments.
In this section, we study how competency affects team formation. We considered the competence of a player as described by his/her rank on the server on which the playing is observed. We are particularly interested in divining the interplay between wanting to play with the best and wanting to play with players with similar competence.

Figure 5.10 plots the CDF of the standard deviation of squad members’ ranks compared to squads that are randomly populated. For the random squad generation, we followed the distribution of squad sizes from the real dataset and randomly assigned ranks to the squad members. Since ranks are unique on a server, once a rank was assigned, it cannot be assigned again to another squad. From the figure, we observe that the standard deviation of ranks from real squads is much lower than those from the randomly generated squads. The median standard deviation for real squads is around 1,800 whereas it is over 4,000 for the randomly generated squads. We further note that 90% of real squads have a standard deviation less than the median of randomly generated squads.

What this means is that players are definitively not joining squads at random, but rather joining them in a manner that suggests competency as an underlying metric. I.e., players are naturally forming squads with players that are around their same competency level.
5.3 Squad Membership Prediction

In this section, we build predictive models based on our analysis in the previous section. Out of 63 Battlefield-4 servers, we focus on the four most popular in terms of distinct players and number of games played. We extract ten features that belong to homophily, familiarity, and competence categories.

Our goal is to build a prediction model to determine whether a pair of players in a game will join the same squad. We select a subset of data for each feature based on their characteristic distribution, considering category representation and size of the resulting dataset. After that, we evaluate different models and choose the best classifier for the most popular server. Finally, we evaluate the accuracy of the model trained on the most popular server, by testing on the dataset from the other three servers.

5.3.1 Features and Categories

Since our focus is to predict whether a pair of players will join the same squad, we build different models based on the three categories that influence team formation. Same clan, same country (both binary), and rank distance (RD) (numeric) are three features for pairs that reflect the homophily category. For competence, we extract four numeric features that formulate pairs performance in a game: average rank (AR), average skill change rate (ASCR), average kills per deaths (AKPD), and average head shots (AHS).

In the preliminary process of building our models, we split the pairs of players on each server into two sets: the training set includes the first 60 days of our dataset, while the testing set contains the remaining 21 days. We consider same side team frequency (SST) (i.e., the players were on the same team during the game) and different side team frequency (DST) for the first 60 days as features belonging to the category of familiarity. The combination of SST and DST defines the frequency of appearing in the same game.

Because the player activity and attributes are highly variable, we select subpopulations with similar characteristics such that we can better map them into categories. For example,
we select the players with average rank (AR) $< 50$, thus with a high competence level. Our choice of thresholds (e.g., 50 for AR and RD in Table 5.2) is informed by the distribution of the features and the objective of selecting at least 10,000 pairs of players above that threshold.

For the binary features, pairs of players in the same clan represent 0.075% of all pairs, a very small proportion of our dataset. However, this set includes 66% of the pairs in the same squad and 76% of the pairs in the same team. In addition, pairs of players in the same country represent about 59% of all pairs. However, this set includes only 5.2% of the pairs in the same squad and 49% in the same teams.

Table 5.2: A comparison of five classifiers for squad prediction over homophily, competence, and familiarity as categories of features.

<table>
<thead>
<tr>
<th>Classes</th>
<th>Features</th>
<th>Random Forest</th>
<th>AdBoost</th>
<th>Decision Tree</th>
<th>Naive Bayes</th>
<th>Logistic Reg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categories</td>
<td></td>
<td>Accuracy  AUC</td>
<td>Accuracy AUC</td>
<td>Accuracy AUC</td>
<td>Accuracy AUC</td>
<td>Accuracy AUC</td>
</tr>
<tr>
<td>Homophily</td>
<td>Same clan</td>
<td>77.08</td>
<td>68.32</td>
<td>72.65</td>
<td>70.44</td>
<td>66.00</td>
</tr>
<tr>
<td></td>
<td>Same country</td>
<td>89.49</td>
<td>52.22</td>
<td>89.14</td>
<td>52.93</td>
<td>80.31</td>
</tr>
<tr>
<td>Competence</td>
<td>AR $&lt; 50$</td>
<td>85.92</td>
<td>60.49</td>
<td>85.33</td>
<td>59.50</td>
<td>74.92</td>
</tr>
<tr>
<td></td>
<td>ASCR $&gt; 5$</td>
<td>89.79</td>
<td>52.69</td>
<td>89.62</td>
<td>56.49</td>
<td>81.59</td>
</tr>
<tr>
<td></td>
<td>AKPD $&gt; 7$</td>
<td>89.72</td>
<td>53.67</td>
<td>89.4</td>
<td>53.26</td>
<td>81.87</td>
</tr>
<tr>
<td></td>
<td>AHS $&gt; 7$</td>
<td>88.28</td>
<td>54.55</td>
<td>88.44</td>
<td>56.70</td>
<td>81.00</td>
</tr>
<tr>
<td>Familiarity</td>
<td>SST $&gt; 20$</td>
<td><strong>80.59</strong></td>
<td><strong>81.57</strong></td>
<td><strong>78.60</strong></td>
<td><strong>81.92</strong></td>
<td><strong>69.77</strong></td>
</tr>
<tr>
<td></td>
<td>DST $&gt; 20$</td>
<td><strong>81.77</strong></td>
<td><strong>72.61</strong></td>
<td><strong>75.65</strong></td>
<td><strong>69.00</strong></td>
<td><strong>70.15</strong></td>
</tr>
</tbody>
</table>

Our dataset is unbalanced, with the majority of pairs from different squads and only approximately 10% from the same squad. We use the synthetic minority over-sampling technique (SMOTE) [81] algorithm to create a balanced dataset from the unbalanced one in the training set. The Area Under the Curve (AUC) and Accuracy measures provide a summary of the quality of the classifier.

### 5.3.2 Model Evaluation and Results

The accuracy and AUC of various classifiers for our models can be seen in Table 5.2. The Random Forest classifier performs better than other classifiers in both metrics on all subsets of features. Moreover, we find that familiarity features are the most influential features. The accuracy and AUC of the set of data controlled for SST exceed 80%. For DST, accuracy exceeds 80% and AUC exceeds 70%.
Figure 5.11: Local predictive models compared to generalized predictive models in squad formation using random forest classifier on different frequencies of games, which includes same side and different side teams, for all pairs of players in the four most popular servers.

Next, we evaluate how different sampling strategies affect the results of our models based on different thresholds. We have two types of models: the local models are trained and tested from the data drawn from each server, and the generalized model is trained only from the most popular server but tested on each server’s local population.

As shown in Figure 5.11, the upper left plot demonstrates the quality of each local model as we vary the threshold of the number of games played together. The local models perform quite well if we choose our threshold to be between 50 and 80 games played together. For the number of games played together, we used SST+DST as the familiarity feature. The remaining plots show how the generalized model compares to the local models. By comparing two predictive models; local and global which are controlled by the frequency of games, we found that the generalized predictive models perform similar to the local model in server 2, but outperform it in server 3 and 4. Overall, our findings indicate that we can discover a global model from the most popular server.
5.4 Summary and Discussions

Gaming environments offer rich potential for studying how individuals choose teammates for a goal-oriented activity. Our results provide evidence consistent with the factors identified by Hinds et al. [32] as important in team formation: familiarity, homophily, and competence. We find that the more often two players play together, the more likely it is that they are playing on the same (large) team and even on the same (smaller) squad within that team. We presume that this effect occurs in part because pairs that had success in previous interactions continue to select one another as partners while those that failed in previous interactions differentially drop out of the future association. A more fine-grained dynamic examination of partner choice that would condition on the previous game outcome is needed to substantiate this interpretation.

Homophily as a factor clearly shows up with respect to the country. To be fair, this fact could be due simply to a time/distance difference between players in different countries serving as a disincentive to play together at the same time on a server in one or another’s country, especially considering that distance can heavily affect latency. Nevertheless, the extent to which interactions between those in the same country exceed chance expectations is quite large, and not just for dyadic teammate relations but also when we examine the composition of squads of different sizes. Homophily also occurs with respect to skill level. Our results make it clear that teammates and squad mates are far more likely to be close to each other in skill than could be expected by chance.

Finally, it is also clear that competence is a driver for interaction, but the connection is complex. First, we found that the most frequent pair interactions occur between players on the same squad with a relatively high rank (i.e., highly competent players). Likewise, pairs that are on average less competent have less frequent interaction. Second, we saw a similar story with respect to the absolute difference in rank between the pairs. I.e., players that were of approximately the same rank were increasingly likely to have more interactions together.
CHAPTER 6

SIGN PREDICTION IN ONLINE GAMING

In this chapter, we formulated the team formation behaviors in online gaming into a sign prediction problem. We classify interactions in online team-based games into four classes; i) cooperative, which indicates players belong to the same teams; ii) non-cooperative, which indicates players belong to different teams; iii) strong cooperative, where cooperative players belong to the same squads, and finally; iv) weak cooperative, where cooperative players belong to different squads in the same team;

Figure 6.1: Illustration of data representation for the sign prediction.

Figure 6.1 illustrates the sign prediction problem in this context. We modeled two scenarios of sign prediction: teams versus squads. A pair of players can have either positive (i.e., cooperative and strong cooperative players) or negative (i.e., non-cooperative and weak cooperative players). The positive and negative signs of squad membership can only be defined
when players belong to the same team (e.g., third pair of players in pairs representation table in Figure 6.1).

We introduced determinants of team formation to generate a number of features leveraging the hypotheses defined in Chapter 5 to predict the signs of every individual pair of a player. Using these features, we built two predictive models; one is for the team and the other one is for the squad. The objective of the first task is to predict whether a pair of players belong to the same team, if they do, the second objective is to predict whether they are in the same squad.

![Figure 6.2: Illustration of familiarity determinants in sign prediction.](image)

6.1 Familiarity Determinants

Familiarity is a function of cumulative shared experiences. Players are familiar with each other when joining the same team or same squad together or opposing teams or squads to play against each other frequently. We introduce two categories of familiarity determinants based on the concept of social ties in social networks [82, 83, 84]. Figure 6.2 illustrates these familiarity determinants. The first reflects shared experiences of direct interactions (single-hop edges). The second brings shared experiences of two-hop interactions [85, 86, 87, 88] which reflect the influence of common neighbors. These common neighbors are associated with
different types of ties and believed to contribute to the evolution of players’ interactions. In the following subsections, we define these two determinants.

### 6.1.1 Explicit Determinants

Four types of direct interactions can be observed in the gaming environments; i) same team, ii) opposing team, iii) same squad, and iv) different squad. The history (weights) of each of these four types of interactions reflects meaningful relations which are explicitly considered as an indicator of familiarity. For example, let $x$ and $y$ be two players who joined a game at time $\tau$. Considering previous interaction of type $k$ between $x$ and $y$ before time $\tau$, the weights of these four different interactions are defined as $W_{\tau}^k(x, y)$ where $k$ is the edge type which can be same side team (SST), different side team (DST), same side squad (SSQ), or different side squad (DSQ). Depending on the type of edges, we extracted four features as the explicit determinants of familiarity in the sign predictive models.

### 6.1.2 Implicit Determinants

The common neighbors concept has been introduced as an effective factor in the social network analysis (SNA) [89, 90] in both signed [91] and weighted signed [92] link prediction. Common neighbors in our case can be considered as an external influence in the determination of the type of relationship between pairs of gamers. We note that online gaming is a competitive environment for players to form allies to achieve their goals. Based on the balance theory [93, 94], the role of common neighbors is relevant in determining the type (sign) of the relationships in competitive environments.

Therefore, we extracted statistical properties relevant to the common neighbors for each edge type. These properties summarize shared experiences with common neighbors and capture potential relationships between them. Let’s suppose that $x$ and $y$ are two players who joined a game at time $\tau$. We define a set of shared experiences with common neighbors as,
\[ CN^*_k(x, y) = \{ min(W^*_k(x, j), W^*_k(j, y)) \}, \forall j \in |\Gamma x \cap \Gamma y| \] (6.1)

where $\Gamma x$ and $\Gamma y$ represent the set of all neighbors for $x$ and $y$. The reason behind choosing the minimum edge weight for each common neighbor is to balance the shared experiences. For example, Figure 6.2 shows that player $X$ and player $Y$ have a number of common neighbors with edge weights. Considering the edge weights of player $X$ and $Y$ with common neighbor CN2, the shared experience is defined as the minimum weight of interactions. This is because player $X$ has comparatively less experience with CN2.

We extracted four statistical properties for each type edge which include cardinality, maximum, mean, and variance of shared experiences expressed as the common neighbors along with each type of interaction $k$. These features are used in the sign prediction.

### 6.2 Competence Determinants

There are two prevalent competency criteria for the gamers: real-time performance and overall performance.

#### 6.2.1 Real Time Performance

Two variables represent gamers’ performance in real time during a match: kills per deaths and headshots. We considered both features named as Average Kills Per Deaths (AKD) and Average Head Shots (AHS). The averages of the variables are calculated between pairs of players in the sign predictive models.

#### 6.2.2 Overall Performance

Generally, players seek to be highlighted in their gaming community. Top-ranked players are more likely to be attractive than others in team formation. Gaming ranking system reflects a combination of the overall player’s performance features. Accordingly, we used the
Average Rank (AR) between pairs of players to quantify the overall performance as a feature in the sign predictive models.

6.3 Similarity Determinants

Homophily explains how players with similar attributes make similar choices and decisions. Huang [66] showed that proximity, as well as homophily in age and game experience, have strong impacts on players’ behavior in creating relations.

6.3.1 Explicit Determinants

The country and the clan are two explicit determinants that could reflect homophily between gamers. The similarity in the clan or country might affect gamers’ decision in selecting their partners in the team or squad. Thus, we extracted two binary features between pairs of gamers that explicitly identify similarity in the country and the clan: SameCountry and SameClan.

6.3.2 Node Similarity Determinants

Due to the limited number of explicit attributes, the structural equivalence of the interaction networks is another measure of similarity between gamers. The objective is to find nodes (gamers) who share the same characteristics with other nodes. We proposed two squad-specific similarity measurements. The first calculates the degree similarity ($DegreeSim$) between two nodes while the second calculates the weighted degree similarity ($WeightedDegreeSim$). Both measurements capture the structural equivalence between gamers based on node degree.

Let $x$ and $y$ be two nodes (gamers) who were observed at time $\tau$. We define the degree similarity as follow:

$$DegreeSim^\tau(x, y) = 1 - \frac{|Degree_{SSQ}^\tau(x) - Degree_{SSQ}^\tau(y)|}{Degree_{SSQ}^\tau(x) + Degree_{SSQ}^\tau(y)}$$

(6.2)
where SSQ is the aggregated squad interaction network. We use squad interaction network because the gamers have a stronger affiliation in the squad more than team and game interaction network. Similarly, the weighted degree similarity is defined as follow:

$$\text{WeightedDegreeSim}_\tau(x, y) = 1 - \frac{|\text{WeightedDegree}_{\tau\text{SSQ}}(x) - \text{WeightedDegree}_{\tau\text{SSQ}}(y)|}{\text{WeightedDegree}_{\tau\text{SSQ}}(x) + \text{WeightedDegree}_{\tau\text{SSQ}}(y)}$$ (6.3)

where \(\text{WeightedDegree}\) is the sum of weights of the node degree. The resulting similarity ranges from 0 (meaning not similar) to 1 (meaning perfectly similar), while in-between values indicate intermediate similarity or dissimilarity.

![Figure 6.3: Distribution of servers size.](image)

### 6.4 Experimental Setup

Our dataset shows 63 servers with a total of 108.8 million pairs of players. Figure 6.3 shows the distribution of servers sizes in regards to the total number of pairs of players, where the top server contains 6.27% of these total pairs. Considering the problem of sign prediction as a binary classification task, we focused on the most popular server to train our classifiers. The signs of the teams in this topmost server were found to be balanced with 49.2% positive signs and 50.08% negative. In contrast, squad signs are normally unbalanced due to the existence of multiple squads in a team (usually five squads) with the majority of signs being negative (approximately 90%).

We used 27 features extracted from the aforementioned three categories of determinants. Table 6.1 describes these features. The total duration of the top gaming server was split into
two disjoint intervals: the training duration was 51 days and testing duration was set to the remaining 30 days. Five different classification algorithms, Logistic Regression, Gaussian Naive Bayes, Adaptive Boosting, Decision Tree, and Random Forest, were used to build squad and team signs prediction. To compare the performance of the classifiers, both the Area Under the Curve (AUC) and Accuracy measures were used to assess the quality of the classifiers.

Table 6.1: Description of the features that are used in the sign predictive models.

<table>
<thead>
<tr>
<th>Category</th>
<th>Determinant</th>
<th>Type</th>
<th>#Features</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Familiarity</td>
<td>Direct/Explicit</td>
<td>Team Signs</td>
<td>2</td>
<td>History of positive signs (SS1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>History of negative signs (DST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Squad Signs</td>
<td>2</td>
<td>History of positive signs (SSQ)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>History of negative signs (DSQ)</td>
</tr>
<tr>
<td></td>
<td>Indirect/Implicit</td>
<td>Team and Squad Signs</td>
<td>16</td>
<td>Total common neighbors (CNs)[Sign Type]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Maximum shared history of signs across CNs [MaxCn][Sign Type]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Average shared history of signs across CNs [AvgCn][Sign Type]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Variance shared history of signs across CNs [VarCn][Sign Type]</td>
</tr>
<tr>
<td>Competence</td>
<td>Performance</td>
<td>Kils Per Deaths</td>
<td>2</td>
<td>Average kills per deaths between two players (AKD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Head Shots</td>
<td></td>
<td>Average head shots between two players (AHS)</td>
</tr>
<tr>
<td></td>
<td>Ranks</td>
<td>Ranks</td>
<td>1</td>
<td>Average ranks between two players (AR)</td>
</tr>
<tr>
<td>Similarity</td>
<td>Explicit</td>
<td>Country</td>
<td>2</td>
<td>A binary value shows two players have the same country (SameCountry)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clan</td>
<td></td>
<td>A binary value shows whether two players have the same clan (SameClan)</td>
</tr>
<tr>
<td></td>
<td>Node Similarity</td>
<td>Degree Similarity</td>
<td>2</td>
<td>Similarity value between two players in the degree</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Weighted Degree Similarity</td>
<td></td>
<td>Similarity values between two players in the weighted degree</td>
</tr>
</tbody>
</table>

6.5 Performance Evaluation

Table 6.2 shows the performance of the classifiers trained with all 27 features. Our objective of this task is to identify the best classifiers for the prediction tasks. Two different testing methods were applied to each classifier. The first method considered all available pairs (All Set) of players irrespective of their pairing patterns. The second method was constrained to consider only those pairs of players who had at least one previous interaction (Prior Team). In other words, we considered whether these pairs had any previous interaction within the same team. Similarly, for team signs prediction of a pair of players, we considered whether a pair of players had at least one previous interaction in the same game (Prior game). The underlying reason for considering the prior game and team information of players is to evaluate the concept of familiarity in assessing the performance of the classifier.

The Random Forest classifier achieved 90% accuracy considering all pairs of players. It also outperformed others in AUC (73%). The accuracy of the squad sign prediction was
Table 6.2: A comparison of five classifiers for signs prediction on the most popular server.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Squad Signs Prediction</th>
<th>Team Signs Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All Set</td>
<td>Prior Team</td>
</tr>
<tr>
<td>Random Forest</td>
<td>73.32%</td>
<td>91.90%</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>68.19%</td>
<td>90.61%</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>70.80%</td>
<td>91.11%</td>
</tr>
<tr>
<td>GaussianNB</td>
<td>58.26%</td>
<td>80.87%</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>57.79%</td>
<td>89.54%</td>
</tr>
</tbody>
</table>

better than the team sign prediction, which can be attributed to the unbalanced classification dataset (90% negative signs). Considering the prior information of players in teams and games, the same classifier achieved 94% accuracy and AUC in squad signs prediction. From the table, it is evident that prior information let the classifier to achieve higher AUC and accuracy. Thus, prior experiences between gamers have a significant impact in predicting squad and team signs.

6.6 Model Generalization

We presume that players behaviors are identical in gaming servers. In other words, the significant factors of predicting signs between gamers are similar across gaming servers. Here, we generalize the signs prediction tasks by training the classifier with the data from one server and evaluating the performance on the data from the other independent servers. Due to its performance, we only used the Random Forest classifier in this experiment. Thus, we used training data as the data from the most popular server to build the predictive models for team and squad signs. We tested the models on the remaining 62 independent servers. Five evaluators—Precision, Recall, F1 score, Accuracy and Area Under Curve (AUC)—were used to evaluate the prediction performance.

Table 6.3 shows the signs prediction results of teams and squads. For evaluating All Set, we observe that AUC does not achieve more than 62% for both sign prediction tasks. Similarly, Recall and F1-score for the positive signs are lower than those in negative signs. This suggests that predicting a new pair of gamers with no history of interactions is difficult. In contrast,
the models can predict more than 83% of the team signs between players who have played in a game at least once. For predicting squad signs between players who have played in a team at least once, accuracy exceeds 96% and AUC exceeds 89%. From all these results, it is evident that the determinants identified in this study are useful for sign prediction tasks. Moreover, the prior interactions between gamers play an important role to predict their signs.

6.7 Summary

In this chapter, we introduced sign prediction tasks based on the determinants of team formation. In Chapter 5, we performed squad membership predictive task on a static form, while the predictive tasks in this chapter treat instances and its features in a dynamic form. We tracked any changes in these determinants in every observation. We extracted time-based features from these determinants to show that our determinants are effective in predicting signs between gamers, indicating that the prior interactions between gamers (familiarity) is more likely to predict accurately current signs. In addition, the results of this study provide insights into the underlying social mechanisms in a dynamic competitive environment. The sign prediction mechanism leveraged social dynamics and employed them as features for predicting the type of interactions in competitive settings.

For future work, this work can be improved through clustering mechanisms to identify signs of the groups instead of pairs. Finally, this work can further be applied to a variety of application scenarios including match recommendation or group recommendation.
CHAPTER 7

PLAYERS MOBILITY NETWORKS

One important aspect of online gaming is the spatiotemporal mobility of players across different gaming servers. In this chapter, we introduce the players’ mobility. The term mobility refers to the movement of players from one gaming server to the other hosting the same game. Since this resembles the migration of an entity from one place to the other, “migration” and “mobility” will be used interchangeably in this chapter. In team-based online games, players move to different servers over time due to various reasons, including technical performance (latency, computation speed), server/game preferences, peer familiarity, or personal endorsements. For example, first-person shooter games have hundreds or thousands of servers when the same game can be played with different teammates. Previous studies showed that players tend to join games repeatedly with a set of familiar players with whom they shared past experience [15, 33].

Understanding players’ mobility between gaming servers is important in multiple aspects, such as server provisioning, traffic redirection in case of server failure, and game promotion. In addition, the migratory patterns of players can be leveraged in modeling information dissemination or behavior adoption. For example, a player may introduce a new set of gimmicks or may affect the server culture via positive or toxic social behavior. Mapping players’ movement to different servers, this study analyzes the temporal migration patterns to quantify their mobility between servers by leveraging temporal interaction patterns.

We specifically focus on social interactions as a factor to characterize players’ mobility patterns. We developed machine learning-based models to predict, first, the popularity of players over time with respect to the number of neighbors following their mobility patterns,
and second, how fast a player moves between servers relative to the others. We present our results using data from two popular online games, *Team Fortress 2* (TF2) and *Counter Strike: Global Offensive* (CSGO), that involve millions of players across a thousand servers over four months. The contributions of this study are multi-fold:

1. It empirically characterizes mobility patterns of players across servers.
2. It models another dynamic process, namely user mobility across servers through temporal mobility networks mechanism built upon players interactions.
3. It identifies the features relevant to the prediction of players’ popularity, including early and late movers in the temporal mobility networks.
4. It shows empirically that the growth and the pace of the mobility can be predicted.

The implication of this study can be attributed to many factors affecting the gaming environment including server provisioning and predicting games and server popularity.

### 7.1 Dataset

We focused on two highly popular games on the Steam platform, CSGO and TF2. CSGO is a tactical combat first person shooter video game where players compete as part of the terrorist or the counter-terrorist team. TF2 is a team-based and objective-oriented first-person shooter game, where players compete on two different teams and can pick a role from different categories, such as pyro, medic, scout, or soldier. The games have similar features including a wide variety of weaponry, maps, in-game voice chat, etc.

We collected data on friendship and temporal gaming interactions through a web crawler that uses the APIs provided by Steam and GameMe. In CSGO, the duration of the collected data range from February 16 - August 9, 2017 (175 days), whereas in TF2, it is from February 16 - April 7, 2017 (51 days). The final dataset recorded over 13 million observations of 1.62 million players and 934 servers in CSGO. For TF2, the dataset contains over two million
Table 7.1: Statistical description of servers’ lifespan and number of matches in CSGO and TF2.

<table>
<thead>
<tr>
<th>Statistics</th>
<th>CSGO</th>
<th>TF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Duration (days)</td>
<td>Games</td>
</tr>
<tr>
<td>mean</td>
<td>66</td>
<td>1245</td>
</tr>
<tr>
<td>25%</td>
<td>31</td>
<td>130</td>
</tr>
<tr>
<td>50%</td>
<td>83</td>
<td>854</td>
</tr>
<tr>
<td>75%</td>
<td>102</td>
<td>2117</td>
</tr>
<tr>
<td>max</td>
<td>102</td>
<td>7149</td>
</tr>
</tbody>
</table>

observations of 231 thousands players in 344 servers. BOT accounts and spectators (i.e., inactive players) were removed from the final dataset.

7.1.1 Game Server

A game server is an authoritative host of game matches. Online multiplayer gaming environments such as first-person/third-person shooter games, and role-playing games, provide a list of servers hosting active matches for players. Players can select server(s) and game matches based on different criteria, including server name, player count, match mode, and network latency.

Servers in online gaming have variable lifespans. The lifespan of a particular server is the duration of that server being active excluding intermittent downtime. In Table 7.1, we present the statistics of servers lifespan duration in both games including the number of games they are hosting. In the left most column of this table, 25/50/75% represent that statistics by considering the first, second and third quartiles. It is observable that more than half of the servers were found active for 83 (out of 102) days in CSGO, and 48 (out of 52) days in TF2. Moreover, the average server lifespan in CSGO was 66 days whereas in TF2, it was 39 days. Similarly, the average number of matches in CSGO was 1245 (maximum 7146) in comparison to 228 (maximum 3103) found in TF2.

Servers can accommodate different numbers of players and matches (from Table 7.1) depending on their life spans. Figure 7.1 presents the distributions of players against games.
for both CSGO and TF2. From the figures, it is observable that the distribution of players across games demonstrates a heavy tail in both CSGO and TF2. This denotes that players tend to fluctuate less in regards to their match preferences.

On the contrary, from Figure 7.2, it is evident that players tend to move across servers within the Steam platform in both games. This signifies the migration propensity of players in online gaming environment and thus the rationale behind this study. It is also observable that there were 84 different servers used by the players in CSGO whereas in case of TF2, the number is 29.

Table 7.2: Characteristics of aggregated interaction networks and friendship networks. Note: ‘NCC’ denotes the number of connected components.

<table>
<thead>
<tr>
<th></th>
<th>Games</th>
<th>Nodes</th>
<th>Edges</th>
<th>Density</th>
<th>NCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>CSGO</td>
<td>969,646</td>
<td>22,502,100</td>
<td>4.80e-05</td>
<td>4,003</td>
</tr>
<tr>
<td></td>
<td>TF2</td>
<td>201,390</td>
<td>5,764,877</td>
<td>2.84e-04</td>
<td>1,644</td>
</tr>
<tr>
<td>Friendship</td>
<td>CSGO</td>
<td>928,863</td>
<td>9,525,587</td>
<td>2.21e-05</td>
<td>2,068</td>
</tr>
<tr>
<td></td>
<td>TF2</td>
<td>154,038</td>
<td>832,944</td>
<td>7.02e-05</td>
<td>4,258</td>
</tr>
</tbody>
</table>
7.1.2 Friendship Network

The friendship networks of the players were also collected from the Steam platform using Steam APIs. Inactive players were excluded from the friendship networks. Table 7.2 summarizes the characteristics of the friendship networks in both CSGO and TF2.

7.1.3 Interaction Network

In online gaming environment, an interaction between two players denote that both of them played together (in the same match of a server). The nature of these interaction is temporal and weighted (multiple interactions between two players over time in different matches across servers). Considering all interactions among players, this study generated dynamic interaction networks. These networks are directed and weighted where links are timestamped. Table 7.2 also describes basic statistics of the dynamic interaction networks aggregated over 102 days in CSGO and 51 days in TF2.

7.2 Data Characteristics

We extracted players’ mobility information over time through their Steam IDs which is unique across servers. In this section, we analyzed the temporal characteristics of servers and players. Among different patterns of players’ mobility, three different observations were considered in this study: 1) players only move to one server, 2) players move to multiple servers once, and 3) players move to multiple servers many times including returns to old servers.

Figure 7.3 shows mobility patterns of players across servers in both games. Each flow between two servers denotes that more than 1000 players in the first game and 100 players in the second game moving from a source server to a destination server. From this figure, the irregular disposition of players’ migratory behaviors is evident, which reflect patterns of their movements across servers in the online gaming environment. We observed that CSGO had more than 1.8 million movements including 56% new movements (players move to a
Figure 7.3: Flow of players’ migration across servers in two games within Steam platform observed in circular tracks, where the inner circular track (lane) represents the source servers and the outer (peripheral) track represents the destination servers.

Figure 7.4: Number of pairs of players observed in the same games over the number of servers.

server first time), while TF2 had about 321 thousand movements including 47% movements denoting players returning to a server they already had played on.

Migration decisions are believed to be socially influenced [95]. Previously (Chapter 5 and Chapter 6), we showed that familiarity is a factor for gamers to play with others. Therefore, players’ movements across server can be described as a socially-motivated phenomenon. Players tend to move with other players with whom they feel comfortable or who have gained their personal association or social identity. Figure 7.4 shows the total pairs of players playing together on a number of servers in CSGO and TF2 compared to players who join the servers randomly. In this way, we see that nearly a million pairs of players play together on multiple servers in CSGO, and about 100,000 pairs of players play together in TF2, while players who
Figure 7.5: Probability distribution of players’ movements to a new destination that followed at least one neighbor (\( \rho \)) vs. the minimum number of games played on the source.

join the servers randomly form much fewer pairs joining multiple server. For the random server user mapping, we followed the players’ observations from the real dataset and assumed that players randomly choose the server to play on (and thus servers are randomly assigned to a player’s activity at a given time). Once a server was assigned, it is removed from the servers list that followed the same distribution as the real data. We extracted pairs of players who were observed at the same time on the same server, as presented in the plots labeled ”Random” in Figure 7.4. This approach maintains the same user activity distribution, the same time of play, and the same server popularity distribution, but disconnects the choice players make on the server they play on.

It is evident from this figure that many players tend to move to specific servers with their peers, which can be an indication that their movements are due to social motivations. In addition, we noticed that the players are definitely not joining servers at random, but rather joining them in a manner that suggests that social motivations could be an underlying factor.

The interaction network, defined in Section 7.1.3, can be leveraged to understand the motivations behind player’s mobility. Therefore, we built temporal interaction networks including server information and time of interaction. We looked at the players who moved to new servers whether they followed their neighbors in the interaction networks. Figure 7.5 shows
the probability of a player following one of his/her acquaintances from previous interactions (prior interaction in a game) to new server versus the minimum number of games played by the migrant before moving. We notice that players with extended activities on a particular server are reluctant to move unless they find familiar peers on the destination server.

For the null models, we followed the players’ observations who moved at the first time in the real dataset and randomly assigned the number of games played before moving. We used the same procedures of random permutation as implemented in Figure 7.4. Consequently, the null models preserve the relationship between players who move on a new server following another player with whom they played before, but assign a random number of games played on the original server from the observed distribution. As Figure 7.5, the null models (random in a number of games played) show low probability for players to follow their peers. In other words, the more games a player payed on a particular server, the more likely that players is to move ONLY when a player with whom he played already moved to the new server. This suggests that gamers are socially influenced by their peers to join another server.

Figure 7.6 shows the probability of a player following one of his/her co-players from previous interactions (prior interaction in a game) to a new server versus the delay of leaving the prior server in days. Observed data shows that the temporal aspect has to be considered
as a factor for gamers to follow their peers in moving to servers: gamers who moved between servers in a shorter time window are more likely to follow their peers. The null models in this situation show no clear trend in the probability of a player to follow another function of the moving delay. To generate the null model, we maintained the same group of players who followed particular other players, but shuffled the delays with which they followed.

7.3 Temporal Mobility Networks

To capture the pattern of players following other players from one server to another, we model players’ move as directed networks called temporal mobility networks built on top of the underlying interaction network. Intuitively, players’ movements across servers can be explained by social interactions, common experiences related to the characteristics of the home server (e.g., over or under-populated, players’ skill, etc.), personal factors (such as the player moving to a different geographical location), and many others. We only capture in this study—due to the inherent limitations of the dataset we collected—reasons due to shared experiences, thus captured by the in-game interactions.

Figure 7.7 illustrates the determinants of social adoption mechanism in the gaming environment that triggers the propagation of mobility among players. In this figure, $\tau_1$ denotes the duration where player $u$ had multiple interactions with player $v$ where the frequency of interactions is denoted by a scalar weight value $\omega_{uv}$. During observation period $\tau_2$, player $u$ was found in server $Q$ and player $v$ was in server $R$. In the next observation $\tau_3$, player $u$ moved to server $P$. During observation interval $\tau_4$, player $v$ was found to adopt player $u$’s migration behavior by following $u$ to server $P$. The intuitive assumption here is that $\omega_{uv}$ has a great impact on the diffusion of mobility behavior among players in online gaming.

We define a temporal mobility network $G = (V, E)$ in which nodes are players and a directed link from node $u$ to $v$ exists iff

- node $v$ moved to server $P$ at time $t_m$.
Figure 7.7: A temporal mobility scenario of two players $u$ and $v$ consists of four servers: $P, Q, R,$ and $S$, and four different observation periods $\tau_1, \tau_2, \tau_3,$ and $\tau_4$, where $\tau_1 \leq \tau_2 < \tau_3 < \tau_4$, and the graph at $\tau_1$ represents the interactions graph between players including $u$ and $v$. $\omega_{uv}$ denotes their weighted interaction count.

Figure 7.8: Building a mobility network from time stamped edges found in the interaction networks.

- node $u$ moved to server $P$ at time $t_n > t_m$;
- nodes $u$ and $v$ have preceding interactions at time $t_i < t_m$.

In this context, node $u$ is considered to adopt/follow node $v$ in his movement to server $P$. It is noteworthy that all directed edges in the resultant mobility network are time stamped which will allow us to explore the temporal path of players’ mobility change. We build a temporal mobility network based on the player movements in a given server. Therefore, for a given server, in the corresponding mobility network’s context, ‘mover’ and ‘adopter’ will be used interchangeably in the rest of the text. In Figure 7.8, we demonstrate a metaphorical representation of developing one mobility network from time-stamped edges employed in this study to construct the mobility network. It is noteworthy that players can be found to follow or adopt multiple players to different servers. In these cases, a different mobility network will
be generated per move to a particular server. Therefore, servers can have multiple mobility networks with different lengths.

In the next section, we explore these networks constructed in this study by following the mobility pattern of players across servers. Before examining different features of mobility networks, emerged from different gaming servers, it is worth mentioning that each network is acyclic and only the earliest (first) move to a particular server by a pair of players is considered. However, future studies can explore weighted mobility networks where multiple moves to the same server by the same pair of players can be considered as a weighted directed edge in the corresponding mobility network. Finally, the edges are time-stamped to allow the study of temporal patterns.

### 7.3.1 Mobility Networks Characterization

In this section, we examine the characteristics of the mobility networks for two games. Table 7.3 presents the main statistics on the mobility networks for both games and servers in games. Servers in the mobility networks are the destinations in the mobility process. Each server will attract disconnected networks of players. The number of disconnected groups (temporal mobility networks) per server for the two games are similar: on average, four groups join each server. The maximum number of mobility networks for two games was 15 and 10 respectively. However, larger groups move in CSGO (maximum is above 8,000 players) compared to TF2 (where the maximum is under 3,000 players).

The distribution of networks’ sizes is highly skewed across servers in both games. Figure 7.9 presents the complementary cumulative distribution functions (CCDF) of the mobility networks’ sizes, calculated by considering the total number of nodes per network, and reveals

<table>
<thead>
<tr>
<th>Games</th>
<th># Networks</th>
<th># Nodes per Network</th>
<th># of Servers</th>
<th># Networks per Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Mean</td>
<td>Max</td>
</tr>
<tr>
<td>CSGO</td>
<td>2,816</td>
<td>2</td>
<td>202</td>
<td>8,434</td>
</tr>
<tr>
<td>TF2</td>
<td>1,316</td>
<td>2</td>
<td>51</td>
<td>2,937</td>
</tr>
</tbody>
</table>

Table 7.3: Basic statistics of mobility networks in each game.
heavy-tailed distributions. From this figure, the heterogeneity of mobility network sizes in both games is evident. Further, in both games, few networks have a high number of node participation. Figure 7.10 shows the average weighted in-degree distribution of players in the mobility networks. The weight in our case would help to differentiate links (in-game interactions) of varying importance, influence, and role. In general, the average weighted degree is average of sum of weights of the edges of nodes. The figure shows that mobility networks follow other real networks in the weighted degree distribution (fat tailed) [96].

7.3.1.1 Influencers and Adopters

The diffusion and adoption process in social networks is a form of collective behavior demonstrated by social groups. Being stimulated by influential friends or neighbors, individuals enthusiastically follow incitement(s) for a finite time period. Cascading behavior that models the diffusion process can be found in any dynamic systems where individuals take a decision or act in response to an impulse triggered by actions of other(s), contrary to his/her own personal preferences. The fundamental reason behind the emergence of this ‘resonance’-like phenomenon is the adoption tendency of individuals due to influence from others [97]. In cascade study, influencers are the seed users where other individuals those follow the seed users’, or adopt their behaviors are called adopters.
Influence has long been studied in various fields like sociology, communication, marketing, and political science. In social network studies, identifying influential users is an important aspect in the case of information dissemination, viral marketing, opinion collection, and trends prediction. Further, social influence also works as a determinant on what individuals adopt and when they adopt it. Consequently, identification of influencers has engulfed with abundant models from social network researchers. Considering the time-stamped edges in the mobility networks, as mentioned earlier, in this study, we adopt a novel mechanism to define popular players (influential) based on the temporal growth rate of their adopters. For example, let $\tau_1$, and $\tau_2$ be two sampling intervals for the total temporal duration $t$ of the mobility network $G_t(V_t, E_t)$ where $\tau_1 < \tau_2$, and $\Gamma_{\tau_1}^v$ denote the followers of node $v_i$ during the observation period $\tau_1$. Considering the number of adopters of node $x$ at the observation period $\tau_2$, we define its followers/adopters growth rate:

$$\delta_{x_i}(t) = |\Gamma_{\tau_2}^v| - |\Gamma_{\tau_1}^v| : \Gamma_{\tau_1}^v \cap \Gamma_{\tau_2}^v \geq 0, \quad i = 1, 2, 3, ...N$$  \hspace{1cm} (7.1)$$

where $N=$total number of nodes in the mobility network. Then,

$$x_i = Popular \quad| \quad \delta_{x_i}(t) > median(\delta_{x_i}(t)) : \quad x_i \in V_{\tau_1}$$  \hspace{1cm} (7.2)$$

In this study, a node can be classified as a popular or non-popular depending on its appearance in the first (initial) observation period of the mobility network formation. The underlying reason behind this is to develop a prediction model (explained in the later sections) to successfully predict players’ popularity instigating the diffusion of migration until the terminal phase of the mobility network formation.

### 7.3.1.2 Early and Late Adopters

Once we have defined popular players in the mobility networks, the next important determinant in the mobility process is the pace of adopters in the mobility networks.
We extracted a set of *temporal-paths* from each mobility network formed in this study using pathpy [98]. A temporal-path consists of a sequence of edges in the mobility network ordered by the node adoption time. In Figure 7.11, we present the sizes of the paths vs the total number of paths found in the mobility networks. We notice CSGO consists of relatively longer chains of migrations than TF2. The timestamped edges in the mobility networks allow us to find out the delays associated with the adopters along their temporal path of adoption. The probability distribution of the median delays in nodes’ adoption time, by considering the temporal paths of a mobility network, is presented in Figure 7.12. Considering the median value of the delays associated with the adopters in the adoption process, we differentiate between early and late adopters.

The median values in both identifying popular players and categorization of the early and late adopters was adopted from two studies. One study by Cheng et al [99] performed a prediction task for the growth of cascades using Facebook shared photos for building cascades and considered the median of the cascades sizes to characterize the growth of the cascades. The other study by Chenhao Tan [100] performed a genealogical study in Reddit to predict the growth of communities. He considered the median of the community sizes to characterize the growth of the communities.

We adopted the same approach for both popularity and pace predictions tasks due to the following considerations. First, by setting the threshold between classes at the median...
value, we build a generalizable solution that can be adopted for various datasets. That is, the threshold is meaningful for each dataset. Second, using the median value leads to well balanced classes, which makes it prediction accuracy measurable, as it can easily be compared with the 50% baseline accuracy. And finally, we could have modeled the problem as a regression problem instead of a classification problems. That is, we could have attempted to predict the exact value of popularity or delay of moving. However, this approach is heavily sensitive to the distribution of values in the dataset, which hinders the generalizability of the results.

7.3.1.3 Adopters vs Non-Adopters

After the identification of popular players, triggering mobility among their adopters, and exploring the early/late adopters, the final thing for us to explore is the difference between adopters and non-adopters. For this purpose, we consider the study by Kleinberg [101]. Following his concept on what motivates an individual to adopt the behaviors of his neighbors, we analyzed the weighted degree distribution of adopters against non-adopters found in the corresponding interaction networks in each game. Suppose $V_x$ and $V_y$ represent two sets of nodes who joined (i.e., adopted the migration) and did not join the mobility (i.e., non-movers) respectively from the interaction network $G_i(V, E)$ where $V_x \cup V_y \in V$. Then, for each node $v_i$, we considered the weighted degree centrality by considering both its neighbors in $V_x$ and $V_y$.
to calculate the ratio $P_{v_i}$ for a player $v_i$ between player’s neighbors who moved with respect to all his neighbors as:

$$P_{v_i} = \frac{\sum_{v_j \in V_x} w_{ij}}{\sum_{v_j \in V_x \cup V_y} w_{ij}}$$

where $w_{ij}$ denotes the weight of the edge between nodes $v_i$ and $v_j$. Figures 7.13 represent sampled distributions over 1000 movers and non-movers in both games. An extract of the corresponding interaction network in TF2 with both the adopters and non-adopters is presented in the network snapshot of Figure 7.14. The sizes of the nodes represent their corresponding weighted degree centrality (i.e., $w_{ij}$ in 7.3) in the interaction network constructed for TF2. From Figure 7.13 and network extract in Figure 7.14, it appears that the players who do not move have a lower ratio of players who moved in their neighborhoods. This exemplifies that players tend to migrate with their partners who are personally endorsed through their interactions trends.

![Network Snapshot](image)

Figure 7.14: Sample of the interaction network with both adopters (red colored nodes) and non-adopters (green colored nodes). Nodes’ sizes represent their weighted degree centrality in the interaction network.
Table 7.4: Features used in the pace (P) and growth (G) prediction tasks. Note: CC denotes clustering co-efficient.

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>Weight of edge to the parent node</td>
<td>P</td>
</tr>
<tr>
<td>In-degree</td>
<td>Node in-degree</td>
<td>G&amp;P</td>
</tr>
<tr>
<td>In-degree_NF</td>
<td>Node in-degree from non-friends.</td>
<td>G</td>
</tr>
<tr>
<td>In-degree_F</td>
<td>Node in-degree from friends.</td>
<td>G</td>
</tr>
<tr>
<td>Out-degree</td>
<td>Node out-degree</td>
<td>G&amp;P</td>
</tr>
<tr>
<td>Out-degree_NF</td>
<td>Node out-degree toward non-friends</td>
<td>G</td>
</tr>
<tr>
<td>Out-degree_F</td>
<td>Node out-degree toward friends.</td>
<td>G</td>
</tr>
<tr>
<td>Weighted In-degree</td>
<td>Sum of the weighted in-degree.</td>
<td>G</td>
</tr>
<tr>
<td>Adoption Rate</td>
<td>Total #adopters per unit time for the node</td>
<td>G</td>
</tr>
<tr>
<td>CC_out</td>
<td>CC of out-going edges</td>
<td>P</td>
</tr>
<tr>
<td>CC_in</td>
<td>CC of in-coming edges</td>
<td>G&amp;P</td>
</tr>
<tr>
<td>CC-NF_in</td>
<td>CC of in-coming edges from non-friends</td>
<td>G</td>
</tr>
<tr>
<td>CC-F_in</td>
<td>CC of in-coming edges from friends</td>
<td>G</td>
</tr>
<tr>
<td>Time Lag/Adoption Duration</td>
<td>Interval between the first and last adoption</td>
<td>G</td>
</tr>
<tr>
<td>In-degree_parent</td>
<td>The in-degree of the node’s parent</td>
<td>P</td>
</tr>
<tr>
<td>Out-degree_parent</td>
<td>The out-degree of the node’s parent</td>
<td>P</td>
</tr>
<tr>
<td>CC-parent_out</td>
<td>The parent’s CC_out</td>
<td>P</td>
</tr>
<tr>
<td>CC-parent_in</td>
<td>The parent’s CC_in</td>
<td>P</td>
</tr>
<tr>
<td>isFriend</td>
<td>If node and its parent are friends</td>
<td>P</td>
</tr>
</tbody>
</table>

7.4 Prediction Tasks

We have two prediction objectives: (i) identify the popular players in the early stage of the mobility networks formation, and (ii) distinguish early and late movers over the lifetime of the mobility networks. The underlying objectives behind these two classification tasks are complementary. First, the identification of popular players helps us detect whether a particular mobility network grows during our observational period. Second, the classification of early/late movers (adopters) measures the speed of growth. We also examine the features that are most useful for the two prediction tasks.

7.4.1 Predicting Popularity

This task of prediction aims to identify the popular gamers responsible for disseminating mobility among others.
7.4.1.1 Methodology

For this task, we select temporal mobility networks with lifespans as long as our observation period. We extracted 178 such mobility networks in CSGO and 82 in TF2. To conform with our definition of popular, described in section 7.3.1.1, we split the network lifespans into four quartiles. We define a node’s popularity growth by comparing its in-degree as observed in the first quartile with its in-degree in the last quartile. We consider a node as being popular if its growth is higher than the median of the nodes’ growth in that particular mobility network. The classification dataset is constructed by considering each node (player) as a prospective candidate of being popular or non-popular. Each datapoint is described by a set of features (listed in Table 7.4 under the task of growth ‘G’) constructed from the structural properties of each node in the mobility networks in the earlier stage. Although the features described in the table is self-explanatory, however, $CC_{in}$ denotes the clustering co-efficient which was computed by considering the triangles among two adopters of the corresponding node. Similarly, $CC-NF_{in}$ and $CC-F_{in}$ denote the triangles between non-friend adopters and friendly adopters respectively. It is noteworthy that friends were extracted from the friendship network. These features were used as input to a supervised learning algorithm, Random Forest, to predict the popular nodes in the later phase of the mobility network. The ratio of the training and testing datasets was 3:1 (75% training data, 25% testing data out of 140 thousand and 14 thousands instances in CSGO and TF2, respectively). The two datasets are nearly balanced: 57% in CSGO and 59% in TF2 are nodes in the non-popular category.

7.4.1.2 Results and Evaluation

The performance of the classifier is described in Table 7.5. In this table, the class one denotes the positive labels and the zero represents the negative labels. In this binary classification task, in addition to accuracy measure, we also used three other popular performance metrics used in machine learning: precision, recall, and $f_1$-score. While recall expresses
the ability to find all relevant instances in a classification dataset, precision expresses the proportion of the data points a classifier classified as relevant were actually relevant.

Considering the Random Forest classifier, we found that our classifier achieved high recall but low precision in classifying the popular player from unpopular ones. The underlying reasons behind the better performance are the size of the classification datasets and rich feature values without significant overlap between positive and negatively labeled data points.

The list of features are ranked according to their importance, calculated by the Random Forest classifier, in Figure 7.15. The out-degree of a node was found the most important feature in predicting the player’s popularity. More surprisingly, the out-degree of a node towards his neighbors those are absent its neighborhood of the friendship network were found to be the most important features in both games. In addition, the in-degree, the delay associated with movement (adoption duration), and weighted in-degree were also found to be important factors to predict popularity in TF2. From both plots in Figure 7.15, it is evident that friendship has minimal impact in predicting how many players will follow on a new server.

7.4.2 Predicting Movement Pace

In temporal path-based networks, a node may end-up joining multiple paths in a different time. Considering the distribution of temporal paths in all mobility networks, an adopter can be classified as early or late adopters (as described in section 7.3.1.2). Here, we described a
Table 7.5: Classification results for predicting popularity, defined by the number of followers, in the mobility networks for both games using Random Forest. Note: class 1 indicates who becomes popular and class 0 who does not.

<table>
<thead>
<tr>
<th>Game</th>
<th>Accuracy</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF2</td>
<td>0.73</td>
<td>1</td>
<td>0.54</td>
<td>0.72</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0.85</td>
<td>0.73</td>
<td>0.79</td>
</tr>
<tr>
<td>CSGO</td>
<td>0.75</td>
<td>1</td>
<td>0.62</td>
<td>0.76</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0.85</td>
<td>0.75</td>
<td>0.80</td>
</tr>
</tbody>
</table>

prediction methodology to successfully predict the pace of adoption by classifying nodes in the mobility networks as early and late adopters.

7.4.2.1 Methodology

For the second task, predicting the pace of growth, we classify nodes in the mobility networks as early and late movers. We extracted a set of temporal-paths from each mobility network formed in this study using pathpy [98]. The temporal-paths is the sequence of edges ordered by the nodes’ adoption time of migration. The time-stamped edges in the mobility networks allow us to determine the adoption delays of the migrating nodes. Considering the definition of early and late adoption presented in subsection 7.3.1.2, the median value of the temporal adoption duration works as a threshold value for the nodes to be characterized as early/late adopters. In each temporal path, the list of nodes having delays less than the median time of the total duration of the temporal path is considered as early adopter and nodes with the adoption delay more than the median value is considered as late adopters.

To predict the pace of adoption or movement, we developed node-specific features that are described in Table 7.4. We build our classification dataset with the nodes along a particular temporal path as a data point and these data points are labeled as one (i.e., positive) to denote late adopter and zero (i.e., negative) as an early adopter. The features in the table describe each a data point and also considered as input to the classifier used in this study. In addition to the Random Forest classifier, we used a Long-Short Term Memory network [102] for the classification task. Long Short Term Memory is a special kind of RNN capable of
learning long-term dependencies. Remembering periodical information for longer duration is their default behavior and its emergence has supported both scalable and effective models for several learning problems related to sequential data. One of the principal benefits of LSTM is that it is free from the optimization problem that hinders the performance of Simple Recurrent Networks. LSTM is a natural choice for our prediction task since the nodes are sequentially ordered by the activation time. In our case, the LSTM used consisted of two blocks of memory cells with two different layers of hidden units. The first layer contains 32 and the second one contains eight units. We also used Adam algorithm as the optimizer with a learning rate of 0.001. The LSTM-based neural network was used to distinguish early and late adopters relative to a given path.

We split the \textit{temporal-paths} set of the mobility networks into two sets: the training set includes 60\% of the paths out of 1.7 million and 155,281 paths in CSGO and TF2 consecutively, while the testing set contains the remaining 40\% of paths.

Table 7.6: Prediction performance demonstrated by both LSTM-based neural network and the Random Forest classifier in predicting the movement (adoption) pace. Note: class 0 indicates early adoption and class 1 indicates late adoption.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Game</th>
<th>Accuracy</th>
<th>Class</th>
<th>Precision</th>
<th>recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>TF2</td>
<td>0.70</td>
<td>1</td>
<td>0.70</td>
<td>0.72</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0.70</td>
<td>0.68</td>
<td>0.69</td>
</tr>
<tr>
<td>LSTM</td>
<td>CSGO</td>
<td>0.72</td>
<td>1</td>
<td>0.70</td>
<td>0.77</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0.73</td>
<td>0.76</td>
<td>0.70</td>
</tr>
<tr>
<td>RF</td>
<td>TF2</td>
<td>0.66</td>
<td>1</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0.65</td>
<td>0.65</td>
<td>0.65</td>
</tr>
<tr>
<td>RF</td>
<td>CSGO</td>
<td>0.69</td>
<td>1</td>
<td>0.67</td>
<td>0.76</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0.71</td>
<td>0.62</td>
<td>0.66</td>
</tr>
</tbody>
</table>

7.4.2.2 Results and Evaluation

Table 7.6 presents prediction performances demonstrated by both the Random Forest classifier and the LSTM-based neural network. As intuitively expected, the performance demonstrated by the LSTM has outnumbered the performance by the Random Forest classifier.
The underlying reason behind the performance improvement by LSTM is its capability of learning the sequence data and consecutive dependency between feature values to successfully classify binary labels. Improved performance by LSTM also proves that in this context, recurrent neural networks can be a better classifier due to the temporal nature of the mobility network paths.

Due to the improved performance by the LSTM over Random Forest classifier, the feature importance of the pace prediction tasks for both games were presented as the Spearman’s rank correlation coefficient $\rho$ between the predicted outcomes versus the ground truth of the test data, as shown in Figure 7.16. The results demonstrate that the in-degree of a node’s parent in the temporal path of the mobility network works as the best performing feature. Alternatively, the weighted interaction between the nodes and their parents with a large number of followers are the principal determinants in predicting their pace of movement. On the contrary, the clustering co-efficient of the nodes’ parents by considering their out-degree neighbors were found to have negative Spearman correlation in both games. Finally, the friendships between nodes and their parents represent only a small proportion of the instances in both games (2%). Thus, it is irrelevant to measure the correlation of the features incorporating the friendship networks.

Figure 7.16: Feature importance by Spearman’s rank correlation coefficient between the predicted outcome and the ground-truth in accurately predicting the adoption pace.
7.5 Summary

This study focused on modeling the temporal mobility patterns of online gamers by tracing the chronological movement of players between two servers. We developed two machine learning-based prediction strategies to predict the growth and pace (speed) in the mobility networks. Our main finding is that a player’s mobility decision is affected by the co-players with the maximum number of interactions and not by the declared friends in the friendship network. This study can further be extended to explore the impact of community-level network structure over player’s mobility across servers.
CHAPTER 8

CONCLUSIONS AND FUTURE DIRECTIONS

Online games support a large economic sector that includes an entertainment ecosystem worth billions of dollars across the world. These games are designed to encourage and reward social play that results in generating a precious recording of dynamic social interactions. Analyzing these temporal interactions and their associated social environment can not only help in understanding human behavior, but also verify whether they exhibit the resemblance of known social interactions in other social contexts. Despite their distinctiveness from other social contexts, researchers believe that online gaming environment should be considered as a legitimate source of data on human interactions. Further, it can serve as a great utility to extract new insights from empirical temporal graph patterns and collective network behavior.

8.1 Summary and Contributions

This dissertation presents data-driven analyses of two collective behaviors demonstrated by players in online gaming environments: team formation and player mobility. This work is an attempt to understand the underlying dynamic of these social phenomena. Our primary contributions and findings can be summarized as follows.

1. We presented a practical framework for retrieving information from online gaming through crawling. The collected datasets are to be made publicly available after proper anonymization, which may serve different research problems. Our web crawler performed longitudinal data collection and can be used for the same purpose in different time granularity. We discussed general considerations and challenges in data collection.
collection that are likely to be of general applicability to other data sources. To the best of our knowledge, there is no effort in collecting massive and longitudinal records of co-play activities in many games with similar characteristics.

2. This dissertation introduced an empirical study of team formation. It specifically investigated how homophily, familiarity, and competence affect team formation. The study showed the similarity between offline interactions and online environments in understanding the factors of team formation. It is the first large-scale quantitative study on the formation of teams in online games based on the combination of familiarity, homophily, and competence. The results of this study provided evidence consistent with the factors identified by Hinds et al. [32] as crucial in team formation: familiarity, homophily, and competence.

3. We proposed sign prediction tasks derived from the team formation problem. We developed machine learning-based models that enable temporal prediction of team and squad signs. We found that dynamic topological features of the gamers’ interaction network are contributing significantly to the performance of sign prediction of teams and squads. This work confirmed that familiarity is an essential factor in predicting signs as a fine-grained form of the team formation problem. Our approach in building predictive models may be used in other competitive environments, such as competitions in online projects between groups in Kaggle or Massive Open Online Courses (MOOCs).

4. This dissertation introduced player mobility across servers in online gaming. It models the dynamic process of players’ movements between servers as temporal mobility networks built upon players’ interactions. It identified the features relevant to the prediction of players’ popularity, including early and late movers in the temporal mobility networks. We found that the growth and pace of mobility can be predicted. Our predictive models can be applied in understanding mobility behaviors in other online platforms such as mobility between social media platforms. In addition, the
models can also be used in offline environments such as job-hopping. The features that are extracted for the models are network-based and can be used in any context. Finally, we found that a player’s mobility decision is affected by the maximum number of interactions with co-players and not by the declared friends in the friendship network.

The rest of this chapter provides future directions related to this dissertation.

8.2 Future Work

The future work directions can be presented in two categories: work that augments the research in this dissertation, and work that addresses other research problems supported by the dataset provided by this dissertation.

In terms of the research that augments the work already done, we identified three lines of work. First, as shown in Chapter 5, 6, and 7, our findings show that the interactions between gamers (co-play activities) through teams or smaller subcomponents of squads are substantial factors in team formation and mobility. Representing gamers interactions via temporal weighted hypergraph leads to extract dynamic topological properties. Previous researchers looked into the use of hypergraphs for representation of complex networks [103, 104]. Extracting these properties for large scale data is complex and challenging. One future research task here is to explore ways of reducing this complexity that keeps important factors with respect to the predictions of teams and the predictions of the mobility.

Second, in Chapter 7, the performance demonstrated by the LSTM was better than the performance by the Random Forest classifier. The underlying reason behind the performance improvement by LSTM is its capability of learning a chronological sequence of data. A future work direction is thus to improve the prediction tasks for both phenomena, through more extensive parameter tuning and better features via graph embedding approach [105].

Third, there are many aspects we wish to tackle that are explicitly related to the design of online games. We wish to explore the impact of game modes on team formation and server selection. For example, casual mode or competitive mode in TF2 might reflect gamers options
in play. We also wish to investigate other factors in server selection, including pings, gamer penalties, server sizes, and map types. These elements might be a complement to the factors that we explored.

Finally, another direction for future work is to quantify the gamers’ migration across servers based on social cohesiveness. We consider that the dynamic process of gamers movements between servers is a complex problem because it combines both spatial and temporal factors. Thus, we propose a Socio-Cohesion metric that measures the social cohesiveness of nodes toward communities in any such social networks. In our case, nodes are the players and communities are the gaming servers. The central assumption of this metric is that players are influenced to play games by their peers with who have strong relationships (interactions). Previous research showed that strong relationships are related to the amount of interactions [84, 106] and the more meaningful than declared relationships [85].

The dataset this dissertation provided can extend research into new directions. One direction for future work is to investigate the effect of informal penalties on the offending behaviors in online gaming. This research is inspired from sociology, which has long recognized the importance of informal controls in the maintenance of social order such as control of criminal activity [107, 108]. Online communities can be more prone to offending behaviors due to the disinhibition effect specific to online environments [109].

In our dataset, admin bans (described in Chapter 4) are the bans that are given by a server administrator, who also determines the length of the ban. Players are banned for specific reasons, such as bad behavior, cheating, breaking playing or server rules, or spamming. In addition, players could be banned because members of the server vote so. The ban data contains the Steam ID of a banned player, start time of the ban, the expiration time or duration of the ban, ban reason, and other details.

One objective of this future research would be to understand the effect of admin bans on gamers’ social position and interaction patterns through characterizing the players’ status before and after the ban. By employing social networks analysis tools, comparing the charac-
teristics of offenders’ ego interaction networks before and after the ban would be an approach to determine the consequences of the ban.

One limitation of applying this approach is that the admin bans do not provide information about banning servers. We do not know where players got banned. However, mapping the ban to a server based on the time gap between the observation of the last activity and the ban time would be a way to avoid this limitation.
LIST OF REFERENCES


[38] Sears Merritt and Aaron Clauset. Social network dynamics in a massive online game: Network turnover, non-densification, and team engagement in halo reach. In *11th Workshop on Mining and Learning with Graphs*, 2013.


[68] Kathleen M Carley, Il Chul Moon, Mike Schneider, and Oleg Shigiltchoff. Detailed Analysis of Factors Affecting Team Success and Failure in the America’s Army Game. *Available at SSRN 2726846*, 2005.


APPENDICES
Appendix A Copyright Permissions

Below is a permission for the use of materials in Chapters 2,5.

<table>
<thead>
<tr>
<th>License Number</th>
<th>4590920489766</th>
</tr>
</thead>
<tbody>
<tr>
<td>License date</td>
<td>May 08, 2019</td>
</tr>
<tr>
<td>Licensed content publisher</td>
<td>ACM (Association for Computing Machinery)</td>
</tr>
<tr>
<td>Licensed content title</td>
<td>Proceedings, Association for Computing Machinery</td>
</tr>
<tr>
<td>Licensed content date</td>
<td>Jan 1, 1900</td>
</tr>
<tr>
<td>Type of Use</td>
<td>Thesis/Dissertation</td>
</tr>
<tr>
<td>Requestor type</td>
<td>Academic institution</td>
</tr>
<tr>
<td>Format</td>
<td>Print, Electronic</td>
</tr>
<tr>
<td>Portion</td>
<td>chapter/article</td>
</tr>
<tr>
<td>The requesting person/organization is:</td>
<td>University of South Florida</td>
</tr>
<tr>
<td>Title or numeric reference of the portion(s)</td>
<td>Chapter 1, Chapter 5, Chapter 9, Chapter 10</td>
</tr>
<tr>
<td>Title of the article or chapter the portion is from</td>
<td>NA</td>
</tr>
<tr>
<td>Editor of portion(s)</td>
<td>NA</td>
</tr>
<tr>
<td>Author of portion(s)</td>
<td>NA</td>
</tr>
<tr>
<td>Volume of serial or monograph.</td>
<td>NA</td>
</tr>
<tr>
<td>Page range of the portion</td>
<td>NA</td>
</tr>
<tr>
<td>Publication date of portion</td>
<td>NA</td>
</tr>
<tr>
<td>Rights for</td>
<td>Main product</td>
</tr>
<tr>
<td>Duration of use</td>
<td>Life of current and all future editions</td>
</tr>
<tr>
<td>Creation of copies for the disabled</td>
<td>no</td>
</tr>
<tr>
<td>With minor editing privileges</td>
<td>no</td>
</tr>
<tr>
<td>For distribution to</td>
<td>Worldwide</td>
</tr>
<tr>
<td>In the following language(s)</td>
<td>Original language of publication</td>
</tr>
<tr>
<td>With incidental promotional use</td>
<td>no</td>
</tr>
<tr>
<td>The lifetime unit quantity of new product</td>
<td>More than 2,000,000</td>
</tr>
<tr>
<td>Title</td>
<td>An Empirical Study on Team Formation in Online Games</td>
</tr>
<tr>
<td>Institution name</td>
<td>other</td>
</tr>
</tbody>
</table>
Appendix A (Continued)

Below is a permission for the use of materials in Chapters 2, 7.

<table>
<thead>
<tr>
<th>Copy Right Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frontiers Editorial Office</td>
</tr>
<tr>
<td>To: Essa Alhazmi</td>
</tr>
</tbody>
</table>

Dear Essa,

Thank you for your email.

Under the Frontiers Terms and Conditions, authors retain the copyright to their work. Furthermore, all Frontiers articles are Open Access and distributed under the terms of the Creative Commons Attribution License (CC-BY 3.0), which permits the use, distribution and reproduction of material from published articles, provided the original authors and source are credited, and subject to any copyright notices concerning any third-party content. More information about CC-BY can be found here: [http://creativecommons.org/licenses/by/4.0/](http://creativecommons.org/licenses/by/4.0/)

You can therefore freely reuse your article, or parts of it, in your thesis.

Best regards,

Gearoid

---

Senior Manager, Research Integrity: Gearoid O Faolain, PhD

Frontiers

[www.frontiersin.org](http://www.frontiersin.org)

Avenue du Tribunal-Fédéral 34

1005 Lausanne, Switzerland

**Office T +44 203 514 26 98**

For technical issues, please visit our Frontiers Help Center [frontiers.zendesk.com](http://frontiers.zendesk.com)