Rise of the Bots: Bot Prevalence and Its Impact on Match Outcomes in League of Legends

Choong-Soo Lee and Ivan Ramler
Department of Mathematics, Computer Science and Statistics
St. Lawrence University
Canton, NY 13617
Email: [clee, iramler]@stlawu.edu

***Abstract*—League of Legends is a multiplayer online battle arena game where features are unlocked as players level up their accounts. Because it takes a significant amount of time to reach the max level, there exist accounts that are leveled automatically by illicit "bots" and then sold on the market at the max level. These bots participate in matches like human players but are incapable of either playing intelligently or cooperatively with teammates. This paper presents an investigation into the prevalence of bots in player-versus-player match types and their impact on match outcomes on the North America and Europe West servers, using the data gathered through the Riot Games official application program interface. We demonstrate that bots are present in all major match modes at various levels and that they negatively influence the balance of matches on both servers.**

I. Introduction

Online games have become more popular over the past decade, and the worldwide video game market was 93.3 billion dollars in 2012, with the projection of 111.0 billion dollars by 2015 [1]. One of the most successful online video games is a Multiplayer Online Battle Arena (MOBA) game called League of Legends by Riot Games. League of Legends was released in late 2009, and over 27 million players play it on a daily basis [2]. Riot Games uses a freemium model for League of Legends where players can download and play the game for free while there are champions and items that players can purchase with both in-game currency (Influence Points) and real currency (converted into "Riot Points"). Players, also known as summoners, gain more experience points and in-game currency as they engage in online matches.

Summoners can strengthen their champions with more experience points and in-game currency (both influence points and riot points). Experience points allow summoners to level up their accounts up to 30. Each level grants summoners a mastery point and a rune slot, both of which they can allocate to enhance their champions. Summoners can purchase champions they do not own and items to boost their experience of playing the game. It requires a significant amount of gameplay to reach level 30 when all mastery points and rune slots are unlocked and to earn enough in-game currency to purchase all the desired champions and runes. A level 30 summoner account requires 40,084 experience points, and a typical competitive match awards about 110 experience points.[1] Assuming an average summoner plays 3 games everyday, it takes about four months to reach the final level. A quick eBay search reveals that the number of level 30 accounts for sale is over 300 in North America and over 1,500 in Europe West. These accounts can be generated by "bots" such as *Summoner Factory* that "is based on pixel detection and emulates keyboard and mouse to play the game."[2] Most of these bots follow a set of predetermined instructions and cannot play intelligently and cooperatively with human and/or other bot participants. Despite Riot Games' efforts to place summoners of similar strengths in a match together using their personal matchmaking ratings[3], there are many matches where human and bot participants are playing together in a team. We will demonstrate how bot teammates can result in an unbalance of matches in contrary to Riot Games' system and efforts to provide fair and enjoyable experience to summoners.

Typically, researchers have investigated how network conditions such as latency, loss, and jitter affect players' quality of experience (QoE) in online video games [3]–[5]. Veron *et al.* collect and publish a League of Legends data set, and analyze players' performance based on their latencies [6]. While these works demonstrate direct impact on players with regard to network conditions, Argawal *et al.* propose a matchmaking system based on players' latencies to improve overall quality of experience for all the participants in the same game sessions [7]. However, to date, very little seems to have been done to investigate how aspects such as the use of illicit bots can influence human players' QoE.

This paper presents results of collecting and analyzing data on League of Legends matches and their participants on the North America and Western Europe servers. Our data analyses show that: (1) bots are prevalent in all the main match modes; (2) the probability of having at least one bot teammate is significant; and (3) bot teammates negatively affect gamers' quality of experience in terms of win percentage, match duration, and player performance score.

The rest of this paper is organized as follows: Section II describes the data collection process, bot tagging criteria, and our data sets; Section III focuses on the prevalence of bots and their impact on players' quality of experience. It introduces our player performance metric, and provides details of our analyses and observations; and Section IV summarizes our findings and considers future work.

[1] http://leagueoflegends.wikia.com/wiki/Experience_(summoner)

[2] http://www.summonerfactory.net/

[3] https://support.riotgames.com/hc/en-us/articles/201752954-Matchmaking-Guide

```
SIDs = initial seeds                ▷ New summoner IDs to query
oldSIDs = [ ]                       ▷ Old summoner IDs to query when available
MIDs = [ ]                          ▷ Match IDs to query
```

(a) Shared Variables

```
while true do
    querySIDs = SIDs.dequeue(40);
    queryLen = len(querySIDs)
    if queryLen < 40 then
        querySIDs += oldSIDs.dequeue(40 - queryLen)
    end if
    levels = RiotAPI.getSummonerInfo(querySIDs)
    for SID in querySIDs do
        recentMatches = RiotAPI.getRecent(SID)
        for MID in recentMatches do
            MIDs.queue(MID)
        end for
        oldSIDs.queue(SID)
    end for
end while
```

(b) Summoner Thread

```
while true do
    MID = MIDs.randomDequeue()
    matchInfo = RiotAPI.getMatchInfo(MID)
    participantIDs = matchInfo.getParticipants()
    for PID in participantIDs do
        if PID not in SIDs and not in oldSIDs then
            SIDs.queue(PID)
        end if
    end for
end while
```

(c) Match Thread

Fig. 1: Pseudocode for Querying the Riot Games API to Gather Relevant Information

TABLE I: Returned Values from a Match Information Query

Name	Description
matchCreation	Match creation time. Designates when the team select lobby is create and/or the match is made through match making, not when the game actually starts.
matchDuration	Match duration in seconds
matchMode	Match mode (Summoner's Rift, Howling Abyss, Crystal Scar, or Twisted Treeline)
queueType	Match queue type (Ranked, Normal, Blind, Solo, Team, etc.)
participants	List of Participant Information (See Table II)
teams	List of Team information

TABLE II: Participant Information in the Match Information

Name	Description
championId	Champion ID (the champion used in the match)
masteries	Masteries used in the match
runes	Runes used in the match
participantId	Participant ID (the summoner ID)
teamId	Team ID (the team on which the participant played)
winner	Flag indicating whether or not the participant won

II. Data Collection

In this section, we provide descriptions of the data collection, our offline tagging of bots, and a summary of data collected. We collected our own data instead of using the League of Legends database published in 2014 [6] because it does not contain key pieces of information we rely on to tag bots as described later in Section II-B.

A. Data Collection

Riot Games offers an official API (application program interface)[4] that provides information about summoners and matches. For our data collection, we use three query types: summoner information, recent matches, and match information. The summoner information query requires at least one summoner ID (up to 40), and returns levels and revision dates of the given summoners. The recent matches query requires one summoner ID, and returns the summaries of the last ten matches (including their match IDs) played by the given summoner. The match information query requires one match ID, and returns the details listed in Tables I and II.

Data collection required a carefully designed Python script because the official Riot Games API is limited to ten queries every ten seconds up to 500 per ten minutes while the number of necessary queries grows exponentially. Figure 1 shows the pseudocode that our script follows to gather information. First, we start with a small set of summoner IDs as seeds. These are new League of Legends accounts that we created and played a balanced mix of match types to reach a wide range of summoners who engage in all available match modes. Then, the script executes both summoner and match threads concurrently. The summoner thread focuses on submitting queries to obtain summoner's levels and recent matches, while the match thread focuses on queries to obtain match modes, teams, and participants of matches. When submitting summoner queries, we exploit the fact that the API allows up to 40 summoner IDs in one query. When performing match queries, we select a random match from the match queue ($MIDs$) instead of FIFO to avoid getting matches from only a small set of summoners. At the end of our data collection, we attempt to have as many complete matches as possible, where a "complete" match is defined as a match with all the necessary information (match created, duration, match mode, winning team, participants (team, level, runes, masteries, champion, kills, deaths, assists)). The summoner thread attempts to maximize the number of complete matches by prioritizing new summoners over old ones. The match thread queues the summoner IDs of participants in matches into one of two queues ($SIDs$ and $oldSIDs$). The $SIDs$ queue contains summoner IDs that have not been queried before, while the $oldSIDs$ queue contains summoner

[4]https://developer.riotgames.com/

IDs that we have queried already. The $oldSIDs$ queue allows the script to query the old summoners again to obtain their updated levels and lists of matches.

B. Bot Tagging Criteria

Before we put together the data collected for analysis, we must determine which participants are bots. Bot detection has been an active area of research in the last decade. Most of these bot detection methods require extensive, if not complete, log data from games, especially First Person Shooter (FPS) or Massive Multiplayer Online (MMO) for identifying bots. For example, Thawonmas *et al.* rely on action frequency and types in an MMO game backlog data provided by the game publisher [8]. Mitterhofer *et al.* analyze player action data on the server side of an MMO game to detect a specific sequence of constantly repeated actions [9]. Kang *et al.* analyze party play behaviors in game activity logs [10] and chatting logs [11] in an MMO game called AION. Chen *et al.* propose a manifold learning technique to analyze players' movements [12] and avatar trajectories [13] in any game where players control movements. Lee *et al.* analyze full action sequence data from an MMO game for bot detection [14]. Han *et al.* propose a bot detection mechanism based on behavioral pattern for an FPS game [15]. Platzer proposes a sequence-based mechanism that can be deployed at both client and server sides to detect bots [16].

We do not have direct access to game logs from Riot Games, and our goal is not to detect bots during gameplay in real-time but instead to analyze their prevalence and impact on matches. Our data is limited to information returned by the official Riot Games API as explained in Section II-A, and the methods discussed in the previous paragraph are inapplicable. Instead, we depend on three in-game features to tag a participant as a bot: champion, masteries, and runes. League of Legends, like many other MOBA games, has a weekly rotation of free champions. This allows summoners to play the game without owning a champion because summoners need to spend either in-game currency or real money to purchase champions. Because a bot is unlikely to purchase a champion, a participant using a non-free champion is likely a human player. Masteries and runes are the other game features that help boost our confidence in bot tagging. Summoners unlock one slot per level to spend into both their masteries and runes. Masteries can be reset at any time without a loss of any mastery points, and allocating all available mastery points can only strengthen summoners' champions. Runes are a similar feature to masteries, except summoners need to spend in-game currency to buy them. Like masteries, runes are designed to enhance summoners' champions. Based on the choice of a champion and the number of masteries and runes allocated, we tag a participant as a bot if the participant uses a free champion and allocate no mastery points or runes. These criteria potentially tag low level summoners as bots because they may still be unfamiliar with the game. However, since the game reminds summoners of unlocking a new mastery and rune slot every level, we are confident that our tagging criteria become more accurate with higher summoner levels.

C. Data Sets

We executed our data collection script from June 13, 2015 to July 16, 2015 (34 days) on both North American (N. Amer) and Western European (EU West) servers. Table III summarizes our data sets for each server. There are 1,406,218 and 1,481,482 unique summoners in N. Amer and EU West servers respectively. Each row in our data sets represents one participant, and contains the number of mastery points and runes allocated, two of the three criteria discussed in Section II-B. The Riot Games official API does not provide a way to find out if a champion was free on a specific date or in a specific match. We were able to cross-reference a data set from our LoL champion usage research project [17] to determine if a champion was free on a given date. Although a new set of ten champions becomes free every Tuesday, it is undocumented and unclear what time of day the switch takes place. Therefore, we identify a champion as free if the champion was free on the previous day of the match as well as on the day of the match. The data sets used for our analysis in Section III are available at Zenodo.[5]

TABLE III: Summary of Data. Match modes are abbreviated by Summoner's Rift (Normal Blind pick) = SR, Howling Abyss (ARAM) = HA, Twisted Treeline (Normal Blind pick) = TT, Crystal Scar (Normal Dominion) = CS, and all other modes = Misc.

Region	Match Mode	Matches	(%)	Participants	(%)
	Total	350,413	(100.00%)	3,341,588	(100.00%)
N. Amer	SR	126,679	(36.15%)	1,266,790	(37.91%)
	HA	59,092	(16.86%)	590,920	(17.68%)
	TT	4,811	(1.37%)	28,866	(0.86%)
	CS	2,029	(0.58%)	20,290	(0.61%)
	Misc.	157,802	(45.03%)	1,434,722	(42.94%)
	Total	298,441	(100.00%)	2,867,839	(100.00%)
EU West	SR	104,587	(35.04%)	1,045,870	(36.47%)
	HA	44,270	(14.83%)	442,700	(15.44%)
	TT	5,189	(1.74%)	31,134	(1.09%)
	CS	2,229	(0.75%)	22,290	(0.78%)
	Misc.	142,166	(47.64%)	1,325,845	(46.23%)

III. Data Analysis

In this section we investigate two main aspects of bots, prevalence and match impact, in player-versus-player (PvP) match modes. Unless otherwise noted, we limit our analyses to teams with an average summoner level of between 10 and 29 (rounded down to the nearest integer). Using the lower end at 10 helps reduce the number of players that have yet to learn the mastery and/or rune system while excluding level 30-only teams allows the analyses to focus on accounts that would have the largest benefit to running a bot program.

A. Bot Prevalence in PvP Games

While bot prevalence can be measured in several ways, we use three basic metrics for it: the proportion of teams with at least one bot participant (p_b), the average number of bots per team ($\bar{b}$), and, p_{all}, the proportion of teams that only have bot participants (i.e., no human members). These metrics give information about the chance that a human player will encounter a bot (p_b), the typical bot prevalence on teams ($\bar{b}$),

[5]https://zenodo.org/record/21708

and the chance that a human player will face a team of five bots in supposedly “player-versus-player” matches (p_{all}).

Table IV summarizes the prevalence of bots for the four main Normal PvP game modes (Summoner’s Rift, Howling Abyss, Twisted Treeline, and Crystal Scar) in both the North America and Western Europe servers. For both servers, Crystal Scar (aka “Dominion”) tends to be the least popular mode (i.e., have the fewest number of matches), but has the largest average number of bots per team (0.72 for N. Amer and 1.50 for EU West). The problem is so severe in EU West that roughly 50% of teams have bot participants and nearly 10% of teams consist of only bots. Additionally, while less a problem in N. Amer, roughly 30% of teams in Dominion have bot participants with just over 5% consisting of only bots.

Further, bots regularly appear in both Howling Abyss (also called “ARAM”) and Twisted Treeline. While neither of these modes are as dominated by bots as Crystal Scar, the high percentage of teams with bots in ARAM (22.1% for N. Amer and 17.7% for EU West) is particularly troubling given the mode’s popularity. That is, in ARAM, many more human players are forced to interact with bots which in turn increases the number of players that have potentially bad experiences due to being teamed with bots. Thankfully, in both servers, bots are far less common in League of Legends’ flagship mode, Summoner’s Rift, with only an average of 0.08 bots per team and only about 7% of teams having bot participants.

TABLE IV: Bot prevalence by match mode for teams with an average summoner level between 10 and 29.

Region	Match Mode	n_T	p_b	b	p_{all}
N. Amer	SR	113,449	0.064	0.08	0.000
	HA	30,172	0.221	0.49	0.023
	TT	3,962	0.309	0.53	0.071
	CS	1,500	0.299	0.72	0.051
EU West	SR	90,527	0.070	0.08	0.001
	HA	20,098	0.177	0.32	0.013
	TT	4,499	0.239	0.36	0.030
	CS	1,844	0.501	1.50	0.095

The popularity of ARAM in the North America server (and as a result the large amount of match data available for it) lends itself to a more detailed analysis of the prevalence of bots. In particular, we can further break down these data by both average team level (Figure 2) and the time of day the match is played (Figure 3).

As seen in Figure 2, bot prevalence remains fairly high for lower level games (roughly defined as levels 10 to 17), with between 40% to 50% of teams having bot participants. By the mid to upper levels, bot prevalence steadily decreases, with roughly 30% of teams having at least one bot at level 20, down to about 12% at level 25, and roughly 5% at level 29. This steady decrease is likely due to more human players shifting their play style from the “player vs AI” modes commonly played by low level people to the PvP modes. (For example, based on our data, at the lower levels, only about 10% of all matches are ARAM, while at the upper levels it increases to approximately 18% of all matches.) Thus, while the overall chance of having bots on a team in ARAM is only 22% (Table IV), low level players are subjected to bots at a much higher rate than upper level players. Players are expected to learn mechanics, strategies, and tactics from

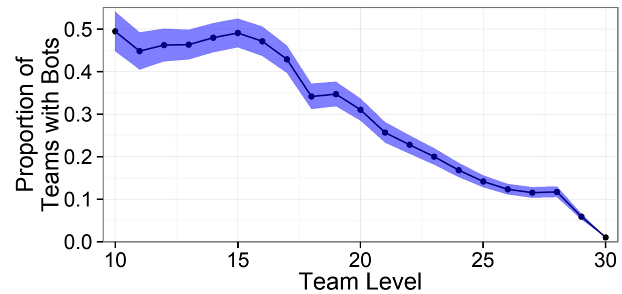

Fig. 2: Proportion of teams with at least one bot participant in North American ARAM matches by average team level. (Band represent a 95% confidence interval for the proportion at each average team level.)

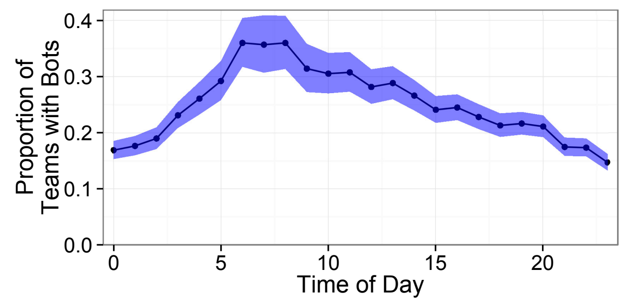

Fig. 3: Proportion of bot participants in North American ARAM matches by hour of day (UTC-05). (Band represent a 95% confidence interval for the proportion at each hour.)

playing more matches, but low level players do not learn anything from bot teammates. Playing against bot opponents may teach low level players inferior strategies and tactics that are ineffective against human opponents. Uncooperative and unintelligent bot teammates and/or opponents also discourage players’ engagement because League of Legends is designed to be a team game. Further, players are more susceptible to lose when their teams have more bot teammates than the opposing teams, as we will demonstrate later in this section. Note that teams with all level 30 players have minuscule bot participation. This is not surprising as the main incentive to running a bot program (leveling up) has been accomplished.

When broken down by time of day (Figure 3), bot prevalence is highest during the morning hours of 6 to 8 am (UTC-05), with between 30% and 40% of teams having bots, then steadily decreasing to about 15% to 19% around midnight. This trend can be largely explained by the main demographic of League of Legends players (i.e., males in their teenage years and low twenties). As most of these players are likely to play in the evenings and nights (as opposed to mornings), the additional volume of human players helps reduce the relative prevalence of bots. Thus for players hoping to avoid playing with (or against) bots, they should generally avoid playing ARAM during the morning and midday hours.

While these figures (and related analyses) are only presented for North American ARAM matches, we provide sim-

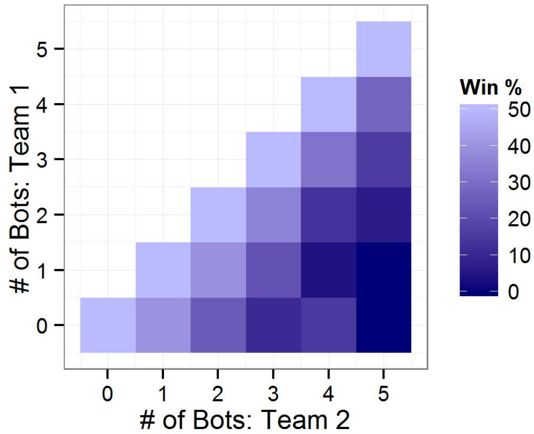

Fig. 4: Win percentage for teams with varying degrees of bot participation for North American ARAM matches. Recorded percent is for Team 2 vs Team 1 (defining Team 2 to be the team with at least as many bots as Team 1).

ilar plots for all four PvP game modes listed in Table IV for both servers in an online app.[6] We leave the investigations of these modes to the reader, noting that they generally have similar tendencies to those seen in Figure 3 (and to some extent Figures 2) and that the time displayed for the EU West server is Coordinated Universal Time (UTC).

B. Bot Impact in PvP Matches

In addition to bot prevalence in PvP matches, another important aspect to consider is the impact bots have on the outcome of matches. In particular, understanding how they impact human players' experiences and the balancing of skill levels of teams. We once again focus on North American ARAM matches and consider team winning percentage, duration of match, and a player performance score as our main metrics for evaluating player experience and balance.

In Figure 4, we illustrate the win percentage of teams with varying numbers of bot participants (defining "Team 2" to be the team with at least as many bots as "Team 1"). Down the diagonal of the plot, the two teams are balanced and, by definition, each team has a 50% chance to win. As the number of bots on the two teams becomes more unbalanced (i.e., shifting away from the diagonal), the win percentage for the team with more bots rapidly decreases. More specifically, when "Team 2" has one more bot, they only win between 30%-40% of matches, about 20% of the time when they have two more bots, and rarely when the bot differential is three or more. Note that the total number of matches for each combination of bot prevalence is different and decreases as more bots are involved in the match. However, the win percentage for each is statistically significantly different from 50% in all cases of unbalanced teams (at the 1% level after using the "Bonferroni-Holm" adjustment for multiple comparisons). This implies that, while Riot Games is attempting to balance matches (such that teams are "equally skilled"), having additional

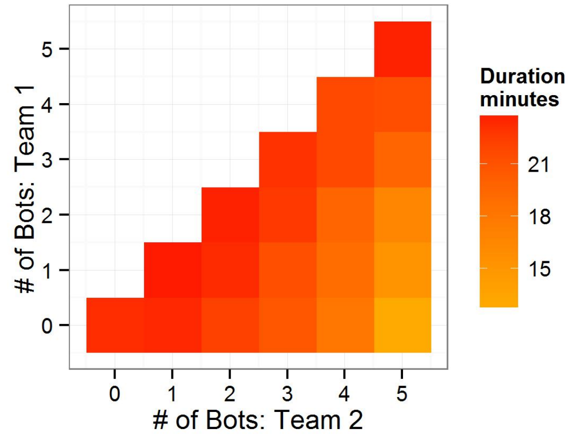

Fig. 5: Average duration of matches (in minutes) for teams with varying degrees of bot participation in ARAM matches.

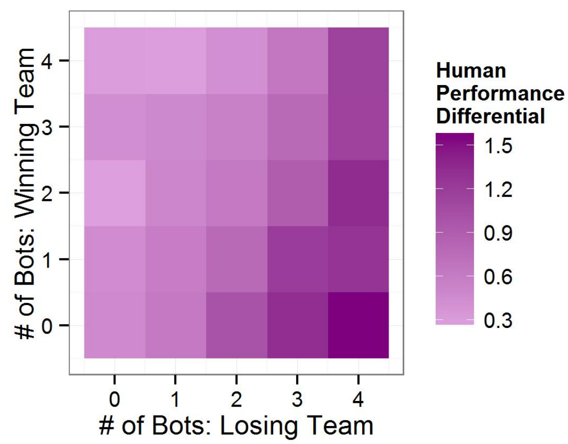

Fig. 6: Difference in the average human Performance Score (kills + assists per minute) from matches with varying bot participants in the ARAM matches.

bot teammates makes teams worse than expected (and hence having win percentages significantly lower than 50%).

Figure 5 displays the average duration of matches for teams with different combinations of bot participation. Matches with no bots tend to last around 23 minutes and as the teams become more imbalanced matches tend to become slightly shorter. However, except in the most extreme cases where one team has all bots, these matches are still relatively long, lasting (on average) between 18 and 22 minutes. Further, when the two teams both have the same number of bots, the average duration is still approximately the same as when no bots are present. This can be an issue as players are then stuck in games with bot teammates for a long period of time unless they prematurely shut down and take a penalty for abandoning the game.[7]

The last aspect of player experience and match balance that we consider is human player performance. We define our

[6]http://shiny.stlawu.edu:3838/LoLbots

[7]http://leagueoflegends.wikia.com/wiki/Leaver_Buster

Performance Score to be the number of kills and assists per minute of the match. (Note that the more common metric, kills + assists divided by deaths, cannot be used here as some players in individual games do not have any deaths.) We then compare the average performance score for human participants between teams within a match. That is, we use an average human performance differential as our specific metric for a match. Figure 6 displays this performance differential for all combinations of bot participation that have at least one human player on each team. In the rare occasions when the team with more bots wins, the humans perform very similarly (with differences in average performance score near zero). In the more common matches where the winning team has fewer bots, the humans on the winning side greatly outperform those on the losing side. This is likely because the winning team's humans are able to take advantage of the poor play of the bots, killing them numerous times. Similar to the shorter durations and lopsided winning percentages, this implies these unbalanced matches become very one-sided as the winning team is able to gain early advantages by killing bots. (This is commonly referred to as "snowballing" in League of Legends.[8])

IV. Conclusion

In this paper, we have analyzed bot prevalence and its impact on match outcomes as a measure of quality of experience in League of Legends. Bots are present in all PvP match modes at varying degrees, and we focus on the significance of playing with bot teammates and/or opponents for players who are still working on reaching the level cap. In Summoner's Rift, which is more popular and an official e-sports match mode, the probabilities of having at least one bot teammate are 6.4% and 7.0% on the North America and Europe West servers respectively. We focus our analysis on Howling Abyss, for which we have the most samples, where there are 22.1% and 17.7% probabilities of having at least one bot teammate in the North America and Europe West servers respectively.

The prevalence of bots negatively influences the balance of the game which will then have a negative effect on a player's experience. Teams with additional bots are less likely to win, remain approximately the same length as games without bots, and tend to become very lopsided. The winning ratios decrease from 50% down to close to 0% as the difference in the number of bot teammates goes from zero to four. Players matched with bot teammates can become frustrated because their performances suffer due to uncooperative and incompetent bot teammates and they cannot leave the match without taking penalties. Further, in the spirit of competition, even players that are on the winning side of "bot prevalent" matches are likely to find these matches unfulfilling. Reaching the level cap eliminates most of these problems, but the journey of leveling up can feel tedious and discouraging.

The Riot Games API match query returns the spells, items, damage dealt and taken, gold earned and spent, and other detailed information about all the participants. A further investigation about these on bot participants can reveal their unique characteristics. The API also provides the time and date of revisions to summoner profile, and it can also allow tracking of bots with regard to the leveling rate and estimated point of sale. League of Legends is one of the most popular and competitive e-sports game, and the presence of participants' usage of champions enables analyses into what team compositions are common and more likely to contribute to the teams' wins.

[8]http://gaming.stackexchange.com/questions/179949/what-does-snowballing-mean

References

[1] R. van der Meulen and J. Rivera. (2013, October) Gartner says worldwide video game market to total $93 billion in 2013. [Online]. Available: http://www.gartner.com/newsroom/id/2614915

[2] P. Tassi. (2014, January) Riot's 'league of legends' reveals astonishing 27 million daily players, 67 million monthly. [Online]. Available: http://www.forbes.com/sites/insertcoin/2014/01/27/riots-league-of-legends-reveals-astonishing-27-million-daily-players-67-million-monthly/

[3] K.-T. Chen, P. Huang, and C.-L. Lei, "How sensitive are online gamers to network quality?" *Communications of the ACM - Entertainment Networking*, vol. 49, no. 11, pp. 34–38, November 2006.

[4] M. Claypool and K. Claypool, "Latency and player actions in online games," *Communications of the ACM - Entertainment Networking*, vol. 49, no. 11, pp. 40–45, November 2006.

[5] M. Claypool and D. Finkel, "The effects of latency on player performance in cloud-based games," in *Proceedings of 13th Annual Workshop on Network and Systems Support for Games (NetGames)*, Nagoya, Japan, December 2014.

[6] M. Veron, O. Marin, and S. Monnet, "Matchmaking in multi-player on-line games: studying user traces to improve the user experience," in *Proceedings of the 24th International Workshop on Network and OS Support for Digital A/V*, Singapore, Singapore, March 2014.

[7] S. Agarwal and J. R. Lorch, "Matchmaking for online games and other latency-sensitive p2p systems," in *Proceedings of the ACM SIGCOMM conference on Data communication*, Beijing, China, October 2009.

[8] R. Thawonmas, Y. Kashifuji, and K.-T. Chen, "Detection of mmorpg bots based on behavior analysis," in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, Yokohama, Japan, December 2008, pp. 91–94.

[9] S. Mitterhofer, C. Platzer, C. Kruegel, and E. Kirda, "Server-side bot detection in massive multiplayer online games," *IEEE Security & Privacy*, vol. 7, no. 3, pp. 29–36, June 2009.

[10] A. R. Kang, J. Woo, J. Park, and H. K. Kim, "Online game bot detection based on party-play log analysis," *Computers and Mathematics with Applications*, vol. 65, no. 9, pp. 1384–1395, May 2013.

[11] A. R. Kang, H. K. Kim, and J. Woo, "Chatting pattern based game bot detection: Do they talk like us?" *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 6, no. 11, pp. 2866–2879, 2012.

[12] K.-T. Chen, H.-K. K. Pao, and H.-C. Chang, "Game bot identification based on manifold learning," in *Proceedings of the 7th Workshop on Network and Systems Support for Games (NetGames)*, Worcester, MA, USA, October 2008.

[13] K.-T. Chen, A. Liao, H.-K. K. Pao, and H.-H. Chu, "Game bot detection based on avatar trajectory," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 3, pp. 162–175, September 2010.

[14] J. Lee, J. Lim, W. Cho, and H. K. Kim, "In-game action sequence analysis for game bot detection on the big data analysis platform," in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, Singapore, Singapore, November 2014.

[15] M. L. Han, J. K. Park, and H. K. Kim, "Online game bot detection in fps game," in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, Singapore, Singapore, November 2014.

[16] C. Platzer, "Sequence-based bot detection in massive multiplayer online games," in *Proceedings of the 8th International Conference on Information, Communications and Signal Processing (ICICS)*, Singapore, Singapore, December 2011.

[17] C.-S. Lee and I. Ramler, "Investigating the impact of game features and content on champion usage in league of legends," in *Proceedings of the Foundation of Digital Games 2015*, Pacific Grove, CA, USA, June 2015.