Project Report Group 2: LoLNetworkAnalysis.gg

Stefan Alexander Stingl (sstingl@mail.uni-mannheim.de), Alicia Lynn Böhret (aboehret@mail.uni-mannheim.de), Tobias Bode (tobias.bode@uni-mannheim.de), Tuananh Vu (tuvu@mail.uni-mannheim.de), and Weiqing Zhu (wezhu@mail.uni-mannheim.de)

Universität Mannheim, Schloss, 68161 Mannheim, Germany

Github

1 Introduction

With developing technology and increasing connectedness, the popularity of computer games, in particular online games, has increased a lot over the past decades. Online games can be divided into different genres, which mainly differ in their game mechanics.

In 2009 Riot Games developed and published a multiplayer online battle arena (MOBA) video game, which is called *League of Legends* (LoL). It is a team-based competitive game which is based on strategy and defeating opponents by breaking a structure in the enemy base. In each match there are ten players, five in each team, which are either randomly assigned or groups individually formed by the users. LoL is one of the most popular games in the world and nowadays has around 180 million active players [1]. Therefore, LoL offers a huge network of players which until now is surprisingly unexplored despite the popularity of the game itself.

1.1 Project Targets

One of the targets of our project will be to analyze friend groups from the network and determine the overall network structure of LoL. This leads to a better understanding of the user behavior and could be very beneficial for further developments of LoL or even new online games. Furthermore, by also taking the rank of the players into account, one could draw conclusions and explanations regarding the individual placement of one player in the network. In particular, the network analysis aims to answer the research questions whether the motivation of players with a high rank differs to the motivation of players with an average rank. The hypothesis we expect to verify is that the motivation of average players is more leisure oriented, so they are more likely to play with the same people, namely their friends. In contrast to that, pro players show a more professional behavior by playing with more random players to improve their rank. On top of that, the following report is also dealing with the research question whether pro players play more frequently than average players. We assume that professional players play more matches in order to improve their rank.

Another idea we initially had was to identify so-called *smurf* accounts, which are second accounts with a lower level compared to a player's main account [2]. As these accounts violate the legal framework, it would have been an interesting finding, but when we were looking at the data we mined, we realized that it is not feasible without combining additional external data, which would have put the focus too much on the mining part instead of the network analysis. To sum it up, in the network analysis of LoL many behaviors and patterns can be discovered and will be therefore the topic of the following report.

2 Experimental Setting

The first step is to collect the data by exploiting certain collection tools and methods. Then certain preprocessing steps need to be executed, before applying the algorithms on the mined data to build a graph and discover patterns.

2.1 Collection Tools & Methods

To collect the match and player data, we utilized the API provided by Riot Games[3], combined with the framework Cassiopeia. This framework provides a more comfortable interface for pulling data from the API, as well as a type system for working with the data and optimizing the amount of data by using the rate limit and a caching system. [4]

We decided early on not to randomly choose the players, for which we collect the match history, because of the size of the active player base. With the resources we have and especially the rate limit of the API, it would be almost impossible to build a meaningful network that way. Instead, we decided to start at a specific player and move outwards from that player. Currently, it is only possible to collect the last 20 matches of each player, in theory we should be able to get more, but we get a faulty response from the API, when we request more.

We came to the conclusion that the best process for our project would be to collect the whole 20 matches in the match history. We added every player, with whom the starting player played with to a list, which we later processed from top to bottom, thus also collecting their full match history. In addition to that, the rank of players, for which we mined the whole match history, are also collected and saved. We can only meaningfully compare features of players for which we collected the whole 20 games match history for. Additionally, collecting the rank took an extra valuable API-Call, therefore we only collected the rank information for said (fully collected) players.

Unfortunately, it was discovered during the collection process, that a current bug with the Riot-API exists. This means, if a player is ranked in an experimental game-mode (Double Up (Workshop)), we will get a faulty response for the rank in the main 5vs5 player mode. Those players were assigned the rank "Unknown" by us. After a careful consideration, we concluded, that we see no reason, why the rank distribution of those players would be significantly different from the distribution of the players for which we were able to obtain the ranks. Therefore, we decided to keep them in the dataset and also include them in the analysis.

2.2 Datasets

We first collected a few test-datasets and ran into problems with the API rate limit, since we needed one API-call per player in a match. After optimizing our collection method, we reduced it to only one API-call per match, plus one additional API-call every 20 matches in order to get the match history of the next player.

As an alternative for the smurf identification, we decided to compare how average players differ from the top players, in terms of their network. To achieve this, we collected two additional datasets, one with the starting point of an average player (AP) and one with the starting point of a pro player (PP). Firstly, we need to define, what exactly we understand as an AP and what as a PP.

An AP is not clearly defined. Most definitions have two aspects in common: A AP usually does not spend a significant part of their time playing or learning about the games they play, and an AP isn't primarily interested in the competitiveness of a game, but rather in socially interacting and just having "fun".[5][6]. In contrast to that, a PP commonly takes part in professional leagues and tournaments and is in general able to generate income from his gaming activities. Be it by salary, prize money, streaming or coaching. Also, investing large amount of time into strategizing and actively improving one's skills are traits of a pro player. In League of Legends, a players' skill level is measured by an Elo-like rating system (MMR) in the backend[7] and represented as a ranking ranging from Iron IV to Challenger to players in the frontend, as illustrated in the total rank distribution of all players

With those definitions and the distribution in mind, we decided to take a player ranked in "Master" as our starting point for the pro-player-dataset. Only 0.2% of players are currently Master-Rank or higher. We didn't select the two higher ranks as a starting point, because we feared, that the player size could be too small at the top rank (there are only 100 players in the challenger rank). This could hurt the network analysis, since we can't differentiate, whether two players queued for a game together or if they were matched randomly by the queue-system. In lower ranks this effect is negligible, but with only 100 potential players to choose from, the queue-system inevitably matches the same players against each other.

For our average-player-dataset we decided to use a player ranked Gold IV as our starting point, that is not the average rank, but slightly above. We went slightly higher than the average Rank, because a higher rank, in general, also correlates with being a more active player. And since currently we can only mine the last 20 games, it was important that those 20 games are not spread out over a large timeframe.

We collected those two datasets, one with the PP as a starting point and one with the AP. The Datasets included the matchID, the participants of every match, the timestamp of every match and for every player we collected the whole match history for, we also included the rank.

The PP dataset consisted of 3051 fully collected players and, 49079 unique matches.

The AP dataset was intended to be the same size, but due to an unrecoverable error in the collection [1], we had to stop at 1982 fully collected players and 37683 unique matches. The Rank distribution of the two collected datasets can be seen in the figure below.

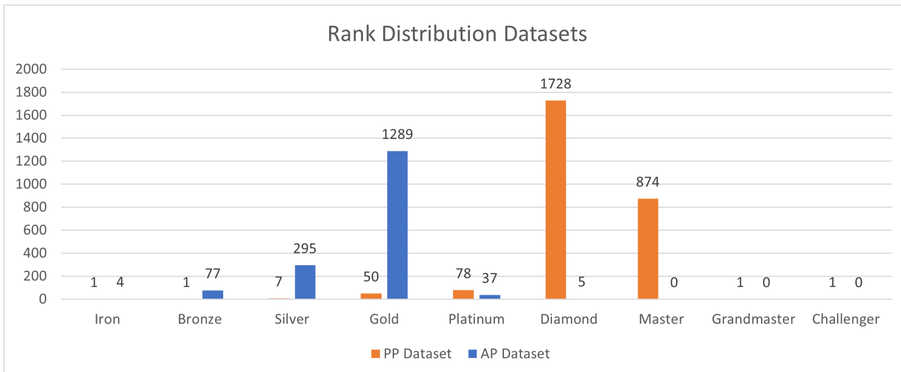


Fig. 1: Rank Distribution in the Datasets

2.3 Preprocessing

With the goal of getting clean and well-organized data from the original data, we conducted a first analysis of the dataset looking for missing values, duplicates and other potential errors. In the matches data set, we first removed brackets and other unnecessary punctuation, converted the columns in the necessary format (e.g. string into datetime) and moved the timestamp in the correct column for matches with less than ten players. After that, we melted the dataframe to get individual columns for each player. After merging the data set, we also excluded bots and empty player columns (from matches with less than ten players). Also, we encountered some duplicates which occurred due to some difficulties when mining the dataset. As the duplicates only affected a small part of the dataset, we still had enough match-player combinations to build a graph on.

To enrich the data with further information, we additionally mined the ranks of the players that we collected the full 20 games match history for. To join this dataset with the match data set, we removed brackets and punctuation as we did for the match data set. We focused on the players where we have the full 20 games history and the ranks for and were able to match the ranks for 92532 matchID - player_puuid combinations of the pro players. Due to the error in the collection process, we resulted with a bit lower number of 43133 matchID-player_puuid-combinations for the average players. These ranks can be exploited

[1] The kernel crashed during the append operation to our backup-file, corrupting the file as a result and preventing a restart of the script.

to draw conclusions from the network. Later in the graph, we only included the players with a node degree of at least 10.

3 Methodology

In this part, our procedure of building a weighted, undirected graph (networkx library) out of the players in the matches we have gathered through the connections of common games in their match histories is described. On the basis of our hypotheses, then certain properties of the graphs are measured and evaluated with corresponding metrics and techniques.

3.1 Building the Graph

As described in the preprocessing step, the data for players with 20 games mined were used to connect each player with the players they played with or against in at least one of their last 20 games. To uncover players who regularly play together, the weight of the edges are increased according to the amount of common games. Additionally, the rank for each player was added as a node attribute and can later be used to filter them according to their rank.

3.2 Community Detection

In order to iteratively improve our model, we first want to create a baseline. Our baseline is a simple model which acts a reference in this project. As baseline for the community detection, we chose the approach of Clauset-Newman-Moore because it delivers fast results in large networks [8].The algorithm uses modularity in order to find community partitions. Modularity is a measure regarding the structure of a graph, it measures the density of connections within a community [9]. Thus, a high modularity has many connections within its community, however only few pointing outwards. Clauset-Newman-Moore joins nodes in order to maximize the modularity of communities until no further increase is possible. In addition, in its calculation it considers the weights of the edges.

Another approach of finding communities is via k-cores and cliques. Unlike Clauset-Newman-Moore k-cores finds communities via the degree of nodes. K-cores are connected components of the graph after removing nodes with degree less than k [10]. Cliques are complete subsets of nodes of an undirected graph. In order to find community structure via cliques, we can look for maximum cliques. These are cliques with the largest possible size. With these methods, it's possible to find community structure quickly. However, weights of edges are not considered in both methods and nodes might be removed with k-cores which are originally in the corresponding community. Therefore, these methods should be used more than indicators for community detection.

The Girvan–Newman [11] detects communities by iteratively removing edges depending on their betweenness centrality. Betweenness centrality is a way to measure the influence of a node. High betweenness centrality is given if an edge has a high number of the shortest path between nodes passing through them [12]. By eliminating these edges, communities are highlighted.

4 Discussion and Evaluation

In the following section, the results of the graph building and community detection phase are discussed. Therefore, conclusions in regard to our hypothesis can be made. To evaluate if our results are meaningful, the results of another research paper are compared with our conclusions.

4.1 Results

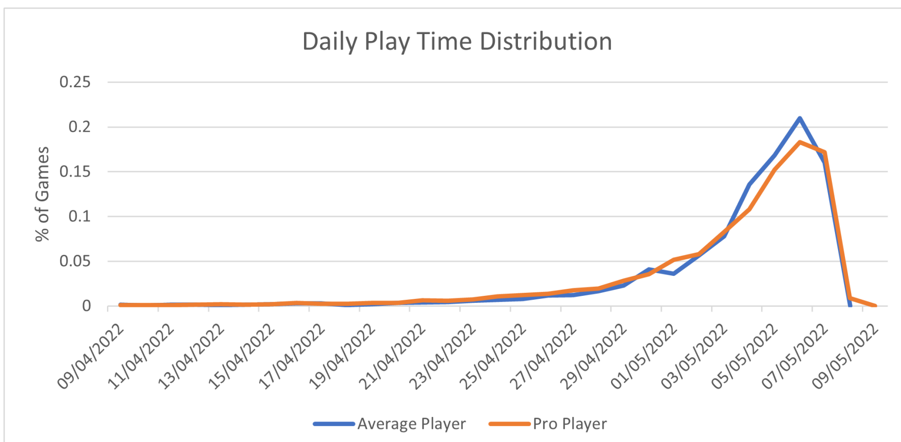
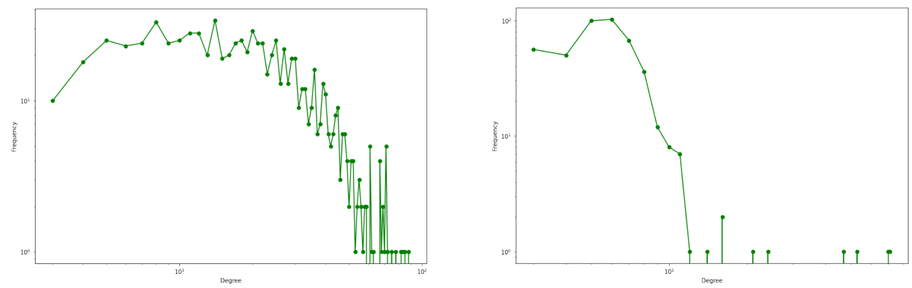


Fig. 2: Normalized distribution of games per day (Red = PP, Blue = AP)

The time distribution of the average_player_dataset and the pro_player_dataset are mainly concentrated in January-May 2022. The time distribution of daily times of games is partial shown (last 30 days) in Fig. 2. It can be seen, that the daily times of games for the AP is slightly higher compared to the PP. This means, that the AP included in our dataset seem to play slightly more often than PP. The data about the average players is shifted to the left by one day, because of the consecutive data collection.

Fig. 4 shows the social network structure of average_players. There are approximately 2000 nodes (players) and, 12000 edges (common matches). Fig. 5 shows the social network structure of pro_players. There are approximately 3000 nodes (players) and, 115000 edges (common matches). The average edge weight for average players (3.2) is higher than the average edge weight for pro players (2.23) which means average players play slightly more often with the same people.



(a) Pro Players (PP) (b) Average Players (AP)

Fig. 3: Degree distribution

From the degree distribution diagram (see Fig.3) of average player, and it can be seen that most nodes have degrees between 1 and 50. For pro players, it can be seen that most nodes have degrees between 1 and 100. This indicates that professional players play with more different players during their games.

	average_player_network	pro_player_network
Density	0.01	0.03
Average clustering coefficient	0.66	0.15
Average_shortest_path_length	8.94	5.33
Triadic closure	0.35	0.095

Table 1: Basic network comparison table

As can be seen in table 1, the density of the network graph for APs is lower than that of PPs, which indicates that the PPs are more closely connected in their network. The average clustering coefficient for PPs is 0.15, which is lower than the random clustering coefficient (0.19). The clustering coefficient of APs is 0.66, which is higher than the average clustering coefficient. It indicates that the network of APs has some community structure, which corroborates with our previous hypothesis. The average shortest path length of APs is higher than PPs, which means the efficiency of information transformation is higher in the network of PPs than APs. The transitivity of the AP is higher than the PP, which means in the AP network if two people in the network have a friend in common, then there is a higher probability that they will become friends at some point in the future. In short, the AP's network is more social in nature.

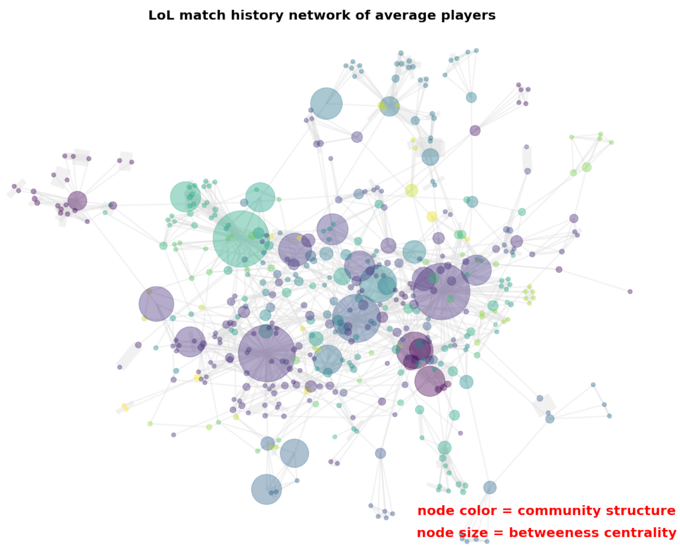


Fig. 4: Match history network of avg_players[**avg_network**]

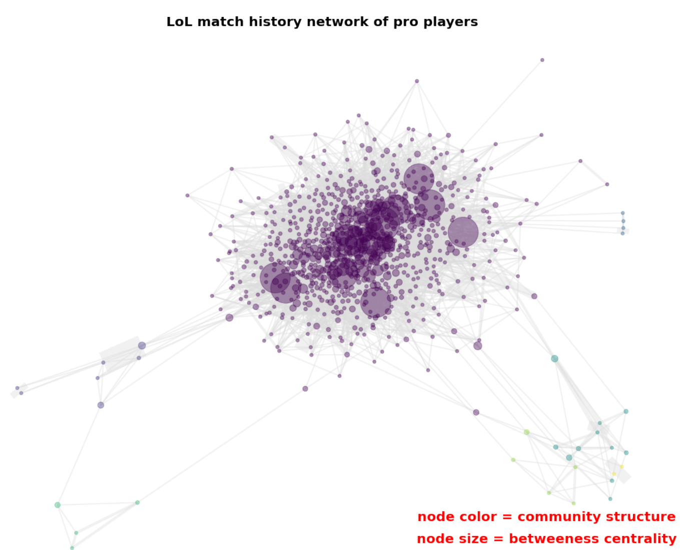
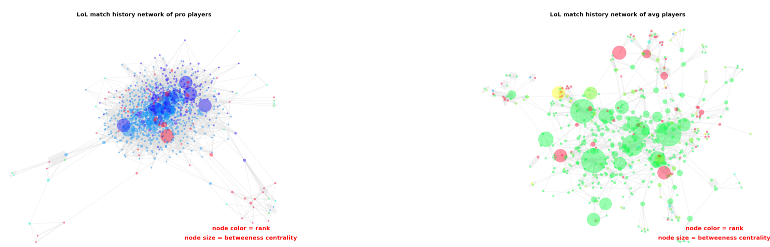


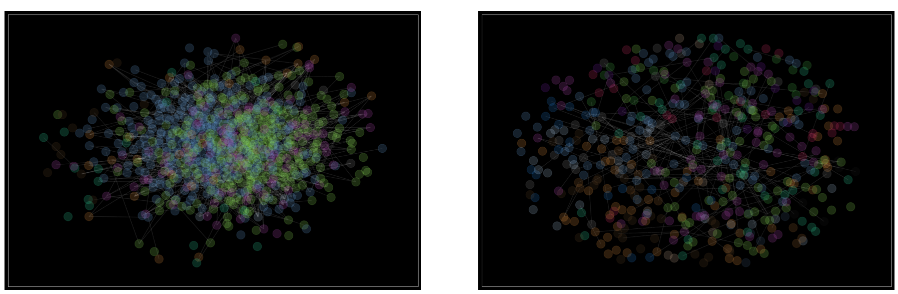
Fig. 5: Match history network of pro_players[**pro_network**]



(a) Pro Players (b) Average Players

Fig. 6: Match history network (higher ranks go towards blue in the rainbow scale)

Furthermore, we looked at the graphs when considering the rank attributes of the nodes (see Fig. 6). In general, it can be seen that the ranks don't mix as much in the more clustered middle of the graphs and on the borders some more ranks creep in, but considering the communities and ranks, no clear trend could be observed.



(a) Pro Players (b) Average Players

Fig. 7: Community detection after Clauset-Newman-Moore

As already mentioned in chapter 3.2 we use Clauset-Newman-Moore to build our baseline graphs. In Fig. 7 we see on the left side the graph of the PPs and on the right side the graph of the APs. The algorithm in the PP graph can detect **8 communities** and in the AP graph **21 communities**. The nodes in Fig. 7 (a) have a higher attractive force to each other than in Fig. 7 (b), this is due to the weights of the graph.

Table 2 shows the result using k-cores and cliques. Unfortunately, the Girvan-Newman could not be applied to our dataset because the computational costs were too big for the scope of this project.

	average_player_network	pro_player_network
Number of nodes	499	833
Number of nodes with at least degree 5	301	786
Number of nodes with at least degree 8	9	706
Number of cliques	332	6845
Biggest clique	9	7

Table 2: K-Cores and Cliques Comparison

4.2 Evaluation

As already described in the previous part, our hypothesis that professional players are more likely to play with random players can be confirmed. This result is due to the fact that the average edge weight of AP's is higher compared to the PP's, and the average clustering coefficient is also significantly higher for the AP's. The clustering-coefficient of the PP network (0.15) might suggest, that the communities detected in the PP network can be attributed to random matchmaking of the queue-system, especially when taking the small player population into account.

To evaluate our results and to ensure that we did not pick a biased sample, which does not represent the real structure of the network, we compared our results to a different research paper. While we were analyzing the average edge weight, Park and Kim were comparing the average number of friends [13]. They found out that there is a slight tendency that PP have fewer friends than AP. This conclusion aligns with our findings regarding the average edge weight. This means that AP's tend to play more with their friends, as they are playing LoL as a leisure-activity. PP are motivated to win and to play matches which help them to get a higher rank, thus they are likely to have less interest in playing with friends. Consequently, the motivation of a player can be distinguished by looking at the variety of players he is playing with.

Looking at the results of table 2 there is the tendency that PPs play more often with friends than expected. Our community detection gives us for the PPs 8 and for the APs 21 communities. Unlike in the AP graph, the number of cliques in the PP graph is higher than the total number of nodes, which means nodes have to be in multiple cliques. In addition, almost 90% of the nodes in the PP graph have a degree of at least 8. The named insights indicate that PPs play with more friends than APs, nonetheless we cannot fully confirm this due to external factors.

One possible reason that could explain our findings is that in ranked competition games, usually there are fewer players in the higher ranks. Therefore, PP end up more often with the same players due to the limited availability of other users. While for AP playing with the same players leads to a higher probability of those being actually friends, for PP it cannot be confirmed. On the contrary, those players could even be enemies.

Our second hypothesis that pro players play more frequently we could not confirm, as the analysis of the timestamp when we fully mined the 20 matches did not show any significant difference (see fig. 2). This could be explained by the fact that our datasets do not represent enough of the most extreme cases, meaning the player's highest and lowest ranks. Another reason could be, that higher ranked players often have multiple accounts or secondary smurf accounts where they play with lower ranked friends. However, those are just guesses based on anecdotal evidence. It could be, very well, that Silver/Gold ranked players don't play more frequently than Master/Diamond ranked players. This thesis could be further investigated by increasing the differences of rank in the datasets.

4.3 Error Analysis

The starting players of our data collection were manually picked. The starting player of the PP dataset was chosen because the account played in a current high tier match and was tagged a pro player by the website op.gg.[2] The AP starting player was chosen randomly out of the Gold IV league rank. Depending on the type and behavior of the starting players, this can have an effect on the whole network. If one was a very social player, who mostly played with friends and the other one played only by themselves, this would not only affect the node of the specific starting player but also recursively the adjacent nodes in the network. This occurs because when entering the queue system as a solo player it is likely, that the system tries to match you with other solo players and not a group of 3 or 4 friends. The AP dataset was only around 2/3 the size of the PP Data set, this could also have resulted in a different network structure.

Another potential source of error could be that the data wasn't gathered over a specific timeframe, but over the last 20 games, regardless of when they took place. The task of identifying popular players within the network were heavily obstructed by the fact that we could only gather the last 20 matches of each player, combined with the problem that we currently can't differentiate if players entered a game together or were thrown together by the queue system. Moreover, we can't differentiate if the players were teammates or enemies. This can improve the network analysis, especially in higher ranks, but took too much API-Calls to be viable for our datasets. Enlarging the amount of matches we collect per player or observing the players over a larger time frame would most definitely improve this situation.

[2] op.gg is a global gaming platform and provides insights into gamers for League of Legends, Overwatch, PUBG and esports in general.

5 Conclusion and Future Work

To conclude our report, it can be seen that many patterns can be discovered, which lead to interesting conclusions. As there has not been made much research with regard to the social network of LoL, our analysis shows that by analyzing the social network, there could be different player behaviors detected. These findings are not only interesting with regard to the developers of the games to understand the players better, but could also be exploited by players who want to reach a higher rank. In summary, we can say, that we were able to find reasonable arguments to support the thesis, that average players are more likely to play with their friends. For our second thesis, we could neither confirm nor deny, that pro players are more likely to play more games than average players.

Starting points for future exploration of the data and the player networks can be seen on different aspects of the project. Foremost, the data could be enlarged and even more levels of player ranks could be collected on their own. Besides collecting more data at large, the existing data could also be enriched with more extra information besides time and rank. The API currently provides a very large set of options. Especially data about the matches themselves (state of the in-game economy over time, statistics about which team and player take which objectives and buy what items). This data is already used in other community projects to build 3rd party recommender-systems regarding in-game items in the Item shop, but could also be utilized in network analysis to differentiate the play-style of average players from that of pro players and also identify friend groups by their play-style. Finding popular players within the network could also be remedied in the future by either gaining access to friend-list information, by collecting a dataset that reaches more than 20 games in the past, or collecting the team the players played in. Other ideas like additionally comparing different regions (in our case we looked only at Europe) or comparing various game modes could bring different angles to compare data and explore interesting aspects in the networks.

Another interesting research question would be, if it could be explored, is whether there are matches where a lot of players from different ranks are playing with each other. These cases could reveal friends in the real world within the network, even for PP. Our analysis already gives some first insights, looking at the pro network colored by ranks (Fig. 6), but the distribution of ranks in the dataset could be improved.

Finally, we can say, that the conducted analysis was fascinating and a great learning experience, but shows that the network of League of Legends players offers a lot more potential which can be exploited.

References

[1] Spezzy. *League of Legends player count 2022*. URL: https://leaguefeed.net/did-you-know-total-league-of-legends-player-count-updated/. (accessed: 19.04.2022).

[2] Spezzy. *5 Ways Smurfing Improves you on League of Legends*. URL: https://leaguefeed.net/ways-smurfing-improves-you-on-league-of-legends/. (accessed: 19.04.2022).

[3] Riot Games Inc. *Developer Portal*. URL: https://developer.riotgames.com/docs/portal. (accessed: 19.04.2022).

[4] meraki-analytics. *cassiopeia*. URL: https://github.com/meraki-analytics/cassiopeia. (accessed: 19.04.2022).

[5] SLANG DEFINE. *What is a Casual Gamer*. URL: https://slangdefine.org/c/casual-gamer-25df.html. (accessed: 18.05.2022).

[6] Boy Robbie et al. *Definiton Casual Gamer*. URL: https://www.urbandictionary.com/define.php?term=Casual%5C%20Gamer. (accessed: 18.05.2022).

[7] *Rank (League of Legends)*. URL: https://leagueoflegends.fandom.com/wiki/Rank_(League_of_Legends). (accessed: 18.05.2022).

[8] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. "Finding community structure in very large networks". In: *Physical Review E* 70.6 (Dec. 2004). DOI: 10.1103/physreve.70.066111. URL: https://doi.org/10.1103%2Fphysreve.70.066111.

[9] M. E. J. Newman. "Modularity and community structure in networks". In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582. DOI: 10.1073/pnas.0601602103. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0601602103. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0601602103.

[10] Joan Adler. "Bootstrap percolation". In: *Physica A: Statistical Mechanics and its Applications* 171.3 (1991), pp. 453–470. ISSN: 0378-4371. DOI: https://doi.org/10.1016/0378-4371(91)90295-N. URL: https://www.sciencedirect.com/science/article/pii/037843719190295N.

[11] M. Girvan and M. E. J. Newman. "Community structure in social and biological networks". In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826. DOI: 10.1073/pnas.122653799. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.122653799. URL: https://www.pnas.org/doi/abs/10.1073/pnas.122653799.

[12] Linton C. Freeman. "A Set of Measures of Centrality Based on Betweenness". In: *Sociometry* 40.1 (1977), pp. 35–41. ISSN: 00380431. URL: http://www.jstor.org/stable/3033543 (visited on 05/24/2022).

[13] Hyunsoo Park and Kyung-Joong Kim. "Social Network Analysis of High-Level Players in Multiplayer Online Battle Arena Game". In: (2011). DOI: 10.1007/978-3-319-15168-7_28.