Profiling Successful Team Behaviors in League of Legends

Conference Paper · October 2017

DOI: 10.1145/3126858.3126886

Fernando Felix do Nascimento Junior
Universidade Federal de Campina Gra…

Allan Sales da Costa Melo
Universidade Federal de Campina Gra…

Igor Costa
Instituto Federal de Educação, Ciênci…

Leandro Balby Marinho
Universidade Federal de Campina Gra…

Profiling Successful Team Behaviors in League of Legends

Fernando Felix do Nascimento Junior
Federal University of Campina Grande - UFCG
Campina Grande, Paraiba - Brazil
fernandofelix@copin.ufcg.edu.br

Allan Sales da Costa Melo
Federal University of Campina Grande - UFCG
Campina Grande, Paraiba - Brazil
allan.melo@ccc.ufcg.edu.br

Igor Barbosa da Costa
Federal Institute of Paraiba - IFPB
Campina Grande, Paraiba - Brazil
igor.costa@ifpb.edu.br

Leandro Balby Marinho
Federal University of Campina Grande - UFCG
Campina Grande, Paraiba - Brazil
lbmarinho@computacao.ufcg.edu.br

ABSTRACT

Despite the increasing popularity of electronic sports (eSports), there is still a scarcity of academic works exploring the playing behavior of teams. Understanding the features that help to discriminate between successful and unsuccessful teams would help teams improving their strategies, such as determine performance metrics to reach. In this paper, we identify and characterize team behavior patterns based on historical matches from the very popular eSpor League of Legends web API. By applying machine learning and statistical analysis, we clustered teams' performance and investigate for each cluster how and to what extent these features have an influence on teams' success and failure. Some clusters are more likely to have winning teams than others, the results of our study helped to discover the characteristics that are associated with this predisposition and allowed us to model performance metrics of successful and unsuccessful team profiles. At all, we found 7 profiles in which were categorized into four levels in terms of winning team proportion: very low, moderate, high and very high.

CCS CONCEPTS

• **Information systems applications** → **Data mining**; • **World Wide Web** → Web mining; • **Computing methodologies** → Machine learning; • **Information retrieval** → *Retrieval tasks and goals*;

KEYWORDS

Clustering, Game Analytics, MOBA Games, Team Performance

ACM Reference format:
Fernando Felix do Nascimento Junior, Allan Sales da Costa Melo, Igor Barbosa da Costa, and Leandro Balby Marinho. 2017. Profiling Successful Team Behaviors in League of Legends. In *Proceedings of WebMedia '17, Gramado, Brazil, October 17–20, 2017,* 8 pages.
https://doi.org/10.1145/3126858.3126886

WebMedia '17, October 17–20, 2017, Gramado, Brazil

ACM ISBN 978-1-4503-5096-9/17/10…$15.00
https://doi.org/10.1145/3126858.3126886

1 INTRODUCTION

The video game industry is one of the most profitable entertainment segments in the world nowadays, surpassing, for example, the film and music industry [16][7] [14]. An important factor for such success is the possibility to play online and team up with players around the world. A very popular game segment is the electronic sports (eSports) [6] where League of Legends (LoL) appear as the most notable and profitable eSport in the world today [22], having 67 million active players worldwide over which 7.5 million play simultaneously in each daily peak [19].

LoL is a multiplayer online battle arena (MOBA) game, a subgenre of real-time strategy video games. A match in a MOBA consists of a scenario (map) where two teams fight each other in order to destroy opponent's base as the main goal, with no time limit. A map contains three main roads (lanes) that connect a base of each team. In general, a team has five players, and each selects and controls a character with distinct attributes and abilities. Besides, the teams also have the assistance of defense structures and units controlled by artificial intelligence (*minions*) to win the match. Throughout a match, the characters gain gold - which is used to buy items to improve their attributes and abilities - and experience in a variety of ways, such as killing units or characters and destroying structures of the enemy team [13].

The high diversity and dynamicity of players in-match actions [3] as well as their performance (i.e. gold earned, killed champions, damage dealt, healing received, etc.) make MOBA games very competitive. In LoL, this competitiveness is amplified due to its popularity and tournaments, which makes many players behave like professional sportsmen [20].

Mastering LoL is very challenging and requires a substantial investment of time [3], especially for inexperienced teams that do not know at first how to improve their strategy. An alternative that could help these teams is to provide information that leads to better decisions in the game, such as performance metrics based on successful player behaviors. This raises several research questions, for example: (i) Is it possible to compute teams' performance metrics? (ii) Is it possible to find useful patterns in the teams' behaviors based on these metrics? (iii) Is it possible to characterize successful and unsuccessful team behavior profiles using these patterns? To the best of our knowledge, there is still a scarcity of academic works exploring this subject in eSports [3] [17], especially about LoL.

Therefore, we present and discuss in this paper the results of a data-driven approach for identifying and characterizing successful and unsuccessful team behavior profiles in the context LoL, based on summarized metrics of players' performance in matches. To this end, we first collected data matches from the official LoL's match history website. Given these data, we perform a feature engineering, which first generated new features from the player's basic statistics, such as the amount of gold earned, the number of monsters killed and total physical damage was taken. Later, in order to select the most informative features regarding the problem, we perform a series of analysis and transformations in the data, such as data cleaning, near-zero variance analysis, outlier analysis, data transformation, normalization and redundancy analysis. After that, we fed these transformed features into a K-means clustering to discover behavior patterns and, then, cluster similar teams together. Finally, we characterize each cluster through an exploratory data analysis. The results imply that some team clusters have more chances to win than others and the relevance of the features are distinct for each one, which allowed us to define successful and unsuccessful team profiles based on the teams' performance metrics.

2 RELATED WORK

A set of publications has investigated MOBAs by applying data analysis and machine learning.

Riolut et al. [20] investigate the behavior of players in Dota 2 - another popular MOBA game developed by Valve Corporation - and its relevance to predict the outcome of matches from in-game positional data of players. Kalyanaraman [9] presents a match outcome predictor and hero recommender based on team composition data of Dota 2, where each entry corresponds a vector encoding the presence or not of a character in the teams. Kinkade and Lim [11] present two win predictors for Dota 2 games: a predictor that uses end-of-game state data and another that uses hero composition data. Ong et al. [17] present an approach based on LoL end-of-game matches to group players' performance behaviors and thus predict match outcomes. Johansson and Wikström [8] create a model to predict the winning team of a Dota 2 match given partial data collected as the match progressed. Schubert et al. [21] present a technique for segmenting matches into spatiotemporal components of Dota 2 players referred to as combat encounters and thus enabling performance analysis and win predictions based on these encounters.

Edge [4] predicts when players will quit a match before it has finished in Dota 2 by modeling players' motivational state in the game. Yang et al. [24] model interactions between the players in matches of Dota 2 as a sequence of graphs to identify successful patterns in combat and thus predict fight results. Eggert et al. [5] present an approach to classify player roles within a team for Dota 2 from summarized data of low-level events in match.

Kim et al. [10] propose a method based on multimodal data (keyboard and mouse usage, game screen, facial expression, volume, and player movement) from the LoL players during the game for automatically detecting the times these players exhibit specific outlier behavior, such as excitement, concentration, immersion, and surprise. Cavadenti et al. [2] propose a method that helps Dota 2 players to improve their skills by discovering outlier successful strategic patterns from historical behavioral traces, i.e., given a model encoding an expected way of playing (the norm), they investigate patterns deviating from the norm that may explain a game outcome.

Pobiedina et al. [18] analyse social behavior patterns of teamwork using data from virtual communities of Dota 2. Drachen et al. [3] present a method to investigate how the behavior of teams in Dota 2 varies depending on the skill level of the players by analyzing their movements and the distance between them over time during the match. Neidhardt et al. [15] investigate the impacts of three types of team factors (players' skills, co-play relations and partnerships with players on prior teams) on performance and duration of team-vs-team matches of Dota 2. Buchan and Taylor [1] explores team play by analysing participants' subjective experiences (such as communication, role, psychological state and level of play) of playing MOBAs to create a conceptual model based on traditional group processes (team roles, group development and perceptions and behavior during the state of deindividuation).

Although these studies play an essential role in literature, none have set out to model team profiles in LoL based on performance behaviors and provide successful teams' performance metrics.

3 FEATURE ENGINEERING

This section describes the collected data and presents a series of performed tasks to extraction, transformation and selection of features.

3.1 Data Collection

LoL developer, Riot Games, provides a web-based Application Public Interface (API) to access match histories in JSON format [19]. A match history contains data such as game mode[1], queue type[2], match duration, winner/loser team and identification number. It also contains data from each player participating in the match as character choice and summarized performance statistics. These statistics indicate, for example, total damage dealt and taken, gold earned and spent, damage dealt and spent, and healing received. Table 1 shows the descriptions of all 37 numeric statistics provided by the API. Due to the huge amount of LoL matches, we randomly collected only 110,000 match histories from February to December 2016 (10,000 for each month). We consider the following filters: *Region* Brazil; *Season* 2016; *Match mode* Classic; *Queue type* Ranked solo; *Total players in each team* 5; *Match version* 6.x.y.

3.1.1 Data Cleaning. In order to avoid false conclusions, we performed data cleaning to detect and remove inconsistencies in teams' dataset. In our analysis, we removed matches with the following conditions:

- Surrendered matches: if a match is extremely unbalanced, a team may request surrender at any time 20 minutes after starting the match.
- Matches containing players that quitted: As the matches are played online, sometimes players can leave during a match due to loss of connection, desistance or other unknown reason, which also causes an imbalance.

[1]The classic mode is the most chosen by the players and a match must have 10 participants dueling each other for two different teams.
[2]A player needs to enter the queuing system to participate in a match, and there are several types of queues.

Table 1: Descriptions of players' performance statistics.

Feature	Description
assists	Number of assists
deaths	Number of deaths
doubleKills	Number of double kills
goldEarned	Gold earned
goldSpent	Gold spent
inhibitorKills	Number of inhibitor kills
killingSprees	Number of killing sprees
kills	Number of kills
largestCriticalStrike	Largest critical strike
largestKillingSpree	Largest killing spree
largestMultiKill	Largest multi kill
magicDamageDealt*	magicDamageDealtToChampions + magicDamageDealtToMonsters
magicDamageDealtToChampions	Magical damage dealt to champions
magicDamageTaken	Magic damage taken
minionsKilled	Minions killed
neutralMinionsKilled*	neutralMinionsKilledEnemyJungle + neutralMinionsKilledTeamJungle
neutralMinionsKilledEnemyJungle	Neutral jungle minions killed in the enemy team's jungle
neutralMinionsKilledTeamJungle	Neutral jungle minions killed in team's jungle
pentaKills	Number of penta kills
physicalDamageDealt*	physicalDamageDealtToChampions + physicalDamageDealtToMonsters
physicalDamageDealtToChampions	Physical damage dealt to champions
physicalDamageTaken	Physical damage taken
quadraKills	Number of quadra kills
sightWardsBoughtInGame	Sight wards purchased
totalHeal	Total heal amount
totalTimeCrowdControlDealt	Total dealt crowd control time
totalUnitsHealed	Total units healed
towerKills	Number of towers the team destroyed
tripleKills	Number of triple kills
totalDamageDealt*	physicalDamageDealt + magicDamageDealt
totalDamageDealtToChampions*	physicalDamageDealtToChampions + magicDamageDealtToChampions
trueDamageDealt*	trueDamageDealtToChampions + trueDamageDealtToMonsters
trueDamageDealtToChampions	True damage dealt to champions
trueDamageTaken	True damage taken
visionWardsBoughtInGame	Vision wards purchased
wardsKilled	Number of wards killed
wardsPlaced	Number of wards placed

3.2 Feature Extraction

Then we extracted the features to model our teams' performance dataset from collected matches. Let M be a set of historical matches where $|M| = 110000$, T a set of teams where $|T| = |M| * 2 = 220000$, $P = \{p_1, \ldots, p_{|P|}\}$ a performance dataset where each $p \in P$ is a vector of $d = 38$ a player performance basic statistics in a match, such as *kills*, *mininonsKilled* and *assists*. A match consists of two distinct teams $m = \{t_a, t_b\}$ and a team consists of 5 distinct players $t = \{p_k | p_k \in P^t; P^t \supset P; 1 \geq k \leq 5\}$. We selected the players' performance statistics in matches and summed up each one $P_j \in P$ by team in order to form each feature or statistic $X_j \in X$ of our teams' performance dataset $X_{220000,38}$ so that $X = \{x_i = \sum_{k=1}^{5} p_k^i | t_i \in T; p_k^i \in P\}$.

Finally, to guarantee data atomicity, we subtracted *physicalDamageDealt* from *physicalDamageDealtToChampions* to compute *physicalDamageDealtToMonsters* and *magicDamageDealt* from *magicDamageDealtToChampions* to compute *magicDamageDealtToMonsters*. We also removed compounded features to avoid redundancy, for example, *totalDamageDealt* which is the sum of *physicalDamageDealt* and *magicDamageDealt*. Table 1 also shows the compounded features marked with (*) we removed.

3.3 Feature Transformation

3.3.1 Calculating the metrics. The modeling data only has a snapshot of players' cumulative end-of-game performance, i.e., summarized statistics. As a match duration has no time limit, we needed to use a measure of performance that would be comparable regardless of duration played. Thus, we computed the division of each entry $x \in X$ in teams' dataset by the duration of the match $m \in M$ in which the team participated:

$$\frac{x}{m^{duration}}$$

In this way, the features of our dataset stop being basic statistics to become performance metrics. For example, *assists* per minute, *deaths* per minute, *doubleKills* per minute, and so on.

3.3.2 Normalization. We performed boxplot on transformed teams' dataset and found the range of values varies widely between features. Therefore, we applied range or min-max normalization [25] in all features so that each feature contributes proportionately in the same scale to perform clustering and redundancy analysis. For each $X_j \in X$ feature, we applied the normalization based on min and max values to scale X_j by the range r:

$$x'_j = \frac{x_{ji} - min(X_j)}{r_j} = \frac{x_{ji} - min(X_j)}{max(X_j) - min(X_j)}$$

After transformation the new feature takes on values in the range $[0, 1]$.

3.4 Feature Selection

3.4.1 Near-zero Variance Analysis. Sometimes the values of some features have zero or near-zero variance which implies that they are noninformative and may cause noise when performing a machine learning task. Therefore, we performed descriptive statistics on data to identify and removed near-zero variance features in untransformed teams' performance dataset. After this analysis, we removed the following features: *doubleKills*, *inhibitorKills*, *largestMultiKill*, *pentaKills*, *quadraKills*, *sightWardsBoughtInGame* and *tripleKills*.

3.4.2 Outlier Analysis. We performed a boxplot analysis in teams' performance dataset X and observed the data had many outliers. So we removed teams with some outlier in their characteristics based on an Interquartile Interval $factor = 1.5$ so that the data behavior were not affected abruptly. Next, we performed a descriptive statistics again to identify and remove near-zero variance features. Based on this analysis, we removed the following features: *totalUnitsHelead* and *visionWardsBoughtInGame*.

3.4.3 Redundancy Analysis. In order to identify and remove redundant features we performed a correlation analysis in transformed teams' performance dataset X. Based on Spearman's correlation test [23], we considered that strong or very strong correlated paired features are redundant. Figure 1 illustrates the correlation plot between the remaining $d = 24$ features of X. According to the plot, we can observe there are several redundant features.

Therefore, for each paired features $(X_a, X_b) \in X; a, b = \{1, \ldots, d\}$; $a \neq b$ with high or very high correlation $r_{ab} >= 0.65$, we removed from X the feature that has the highest correlation in the pair, i.e., the feature in the pair in which the sum of all its correlations with the other features of X has the highest value:

$$max\{\sum_{i=1}^{d} r_{ai}, \sum_{i=1}^{d} r_{bi}\}; a \neq i; b \neq i$$

Therefore, we removed the following features: *assists*, *goldEarned*, *goldSpent*, *kills*, *largestKillingSpree*, *magicDamageDealtToChampions*, *physicalDamageDealtToChampions* and *towerKills*.

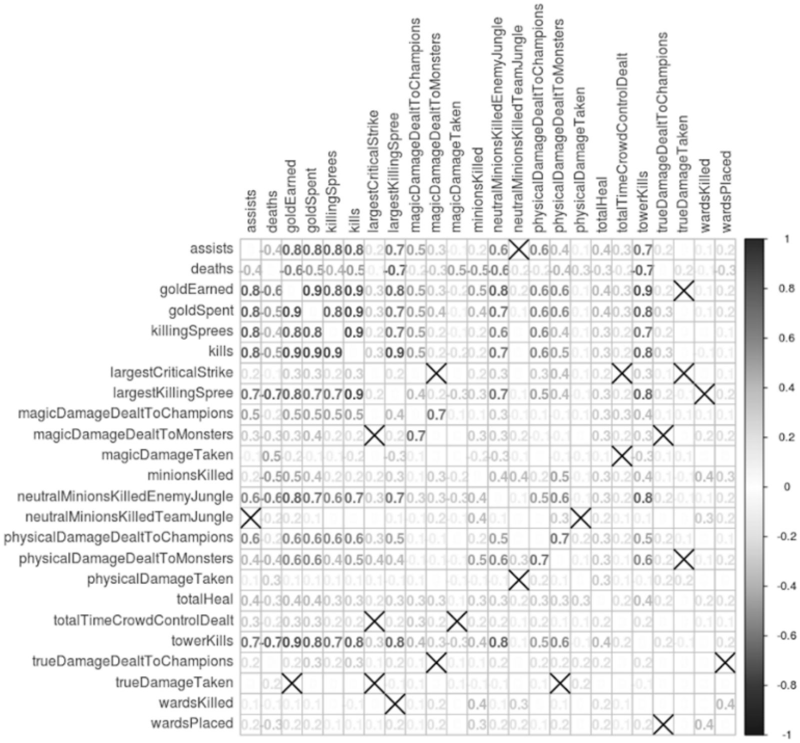


Figure 1: Correlation plot for teams' dataset features. Each cell represents a correlation. The grayscale pallet indicates the correlation intensities.

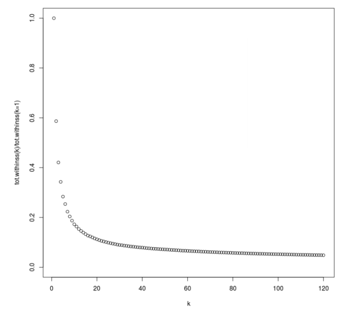


Figure 2: K-means SSE curve: $BBSSE(k)/TSSE$ **for each** $k \in 1:120$

4 CLUSTERING

The problem of finding team profiles is equivalent to identifying a set of clusters of teams with similar behavior. We used K-means clustering [25] to approach this problem. Given a set of dimensional entries $x_1, \ldots, x_n$, K-means clustering aims to partition the n entries into $k(\leq n)$ clusters, denoted by $C = \{C_1, \ldots, C_k\}$, so as to minimize the within-cluster Sum of Squared Error (SSE); its objective is to find the minimizer $C*$ of the following function:

$$SSE(C) = \sum_{i=1}^{k} \sum_{x_j \in C_i} ||x_j - \mu_i||^2$$

Where μ_j is the mean of the entries in C_i, and indicates the *ith* centroid of C.

K-means uses an iterative greedy approach to find a cluster that minimizes the objective SSE and, as such, can converge to a local optimum instead of a global optimum grouping [25].

For our teams' performance dataset we used the Lloyd algorithm, a heuristic consisting of randomly select entries as centroids of k clusters C and assign iteratively each entry $x \in X$ to the nearest centroid and then update the centroids with the mean of its respective clusters [17].

To define the optimal k number of clusters we used a heuristic method that finds the *elbow* of the error curve. This method tries to find an appropriate number of clusters analyzing the curve of a generated graph from a test conducted for each possible number of clusters [12]. In this case, the test was based on the SSE function.

Figure 2 illustrates the plot of SSE test for each possible number of clusters $K = \{1, \ldots, 120\}$. As we can observe, the teams' dataset X can be partitioned into $k = 7$ clusters. By assigning each $x \in X$ in a cluster $c_j \in C$, we decreased the data variability by approximately 78%, i.e., a proportion of Between-cluster Sum of Squared Errors (BSSE) and the Total Sum of Squared Errors (TSSE) by approximately $BSSE/TSSE = 78\%$.

5 PROFILING

In this section, we characterize the teams' performance dataset clusters in order to put their main features or performance metrics into perspective and thus define behavioral profiles. In order to understand how the clusters i.e. profiles) found differs, we analyzed: (i) The number of teams and winning team proportion (win rate) and losing team proportion (loss rate); (ii) The centroids that summarize the features of the profiles; (iii) To what extent the features have influence or relevance in the profiles.

Figure 3 shows how the profiles differ in terms of number of teams and win/loss rate. Figure 4 shows the heatmap of the feature relevance analysis based on information gain applied to the teams' performance metrics dataset without normalization (discussed in section *3.3.1*) M to indicate how they influence each profile. Table 2 shows the centroids that represent each profile features and how they differ in terms of performance metrics. Let A be the profile centroids matrix of teams' performance metrics M, each line $a \in A$ indicates the centroids of a profile and each column $A_j \in A$ indicates the centroids of a feature or metric over the profiles. Table 2 shows the transposed profile centroids $t(A)$ and denotes how the profiles differ in performance. Figure 5 shows the radar plots of the profiles, where each radar represents a profile performance metrics, each axis represents a metric and the axis length indicates the profile score in a specific metric. The performance metrics used to model the radar plots are based on the centroids A normalized by feature $normalization(A_j)$, so that an axis length is proportional across the profiles and assumes a value between $[0, 1]$.

By observing the results, we split the profiles into 4 levels regarding win rate: very low, moderate, high and very high.

5.1 Very Low

The very low performance level consists of Profile 1 and Profile 4. This level is distinguished by a very low win rate $w_r = 10\%$ and very high loss rate $l_r = 90\%$ (Figure 3). There are 10 relevant features for Profile 1 and 7 relevant features for Profile 4 (Figure 4). The most relevant ones ordered by importance are: *neutralMinionsKilledEnemyJungle*, *deaths*, *killingSprees*. Table 3 summarizes the similarities and differences between the feature scores for profiles (Figure 5). The scores that most differentiate between the profiles are: *magicDamageDealtToMonsters* (middle for Profile 1 and very low for Profile 4) and *trueDamageTaken* (very high for Profile 1 and middle for Profile 4).

5.2 Moderate

The moderate performance level consists of Profile 2 and Profile 5. This level is distinguished by a moderate win rate $w_r = 55\%$ and low loss rate $l_r = 45\%$ (Figure 3). There are 14 relevant features for Profile 2 and 15 relevant features for Profile 5 (Figure 4). The most relevant ones ordered by importance are: *deaths*, *neutralMinionsKilledEnemyJungle*, *killingSprees*, *magicDamageTaken*. Table 4 summarizes the similarities and differences between the feature scores for profiles (Figure 5). The score that most differentiate between the profiles is: *magicDamageDealtToMonsters* (middle for Profile 2 and very low for Profile 5).

5.3 High

The high performance level consists only Profile 6. This level is distinguished by a high win rate $w_r = 67\%$ and very low loss rate $l_r = 33\%$ (Figure 3). There are 12 relevant features for Profile 6 (Figure 4). The most relevant ones ordered by importance are: *deaths*, *neutralMinionsKilledEnemyJungle*, *killingSprees*, *magicDamageTaken*. Table 5 shows the performance scores of the profile classified for this level.

5.4 Very High

The very high performance level consists of Profile 3 and Profile 7. This level is distinguished by a very high win rate $w_r = 85\%$ and very low loss rate $l_r = 15\%$ (Figure 3). There are 8 relevant features for Profile 3 and 11 relevant features for Profile 7 (Figure 4). The most relevant ones ordered by importance are: *deaths*, *neutralMinionsKilledEnemyJungle*, *killingSprees*. Table 6 summarizes the similarities and differences between the feature scores for profiles (Figure 5). The scores that most differentiate between the profiles are: *magicDamageDealtToMonsters* (middle for Profile 3 and very low for Profile 7) and *totalTimeCrowdControlDealt* (very high for Profile 3 and middle for Profile 7).

6 DISCUSSION AND CONCLUSION

In this study, we aim to answer three research questions: (i) is it possible to compute teams' performance metrics? (ii) is it possible to find useful patterns in teams' behavior based on these metrics? (iii) is it possible to characterize successful and unsuccessful team behavior profiles using these patterns? Therefore, we propose an approach that goes through various feature engineering tasks to compute teams' performance metrics from LoL matches, uses K-means clustering to find team clusters (patterns) from theses metrics and, then, analyzes different aspects of the team clusters in order to characterize successful and unsuccessful behavior profiles.

Our results show that teams in the collected matches share several similarities and differences. We identify 7 distinct teams' profiles that put into perspective such similarities and differences according to the summarized and transformed players' performance statistics. We compute for each profile the win/loss rate based on the proportion of winning and losing teams, and we categorize the profiles into four win rate levels as follows: very low, moderate, high and very high. Each win rate level can be interpreted as one big cluster. Regarding the distribution of the teams along the profiles, 28% of teams fall into the Very Low level, 36% into the Moderate level, 11% into High Level and 25% into the Very High level. Regarding the feature relevance, those that seem to have more influence on the behavior of the profiles are *deaths*, *killingSprees* and *neutralMinionsKilledEnemyJungle*. In the Very High win rate cluster we find teams that usually present high levels of *killingSprees*, *minionsKilled*, *neutralMinionsKilled*, *totalHeal* and *wardsPlaced*. Based on data, we could conceive that these teams use a strategy based on maximizing its farming and minimizing enemy's farming, obtaining an advantage on gold - which is converted to items - and experience, and increasing the probability of obtaining kills when fighting enemy heroes. The Very Low win rate cluster could be interpreted exactly the opposite of Very High cluster. The teams found in this cluster show a very low level of *minionsKilled*, being unable to obtain items and becoming vulnerable to the opponents' attack. The Moderate cluster presents low levels of deaths, *magicDamageTaken* and *trueDamageTaken*; this may indicate that teams in this cluster focus heavily on defense. The High cluster presents low level of deaths, but a very high level of *magicDamageDeltToMonters* or *magicDamageDeltToChampions*; this may indicate that their main strategy is the surprise attack, since they do not focus so much on defense and have few deaths.

The findings of this study suggest some very concrete elements that can be used to enrich inexperienced teams' strategies of LoL. By analyzing the performance of successful profiles, an inexperienced team could evaluate their metrics in matches and thus support decisions according to the profile in which the team most fits.

For the future, we could to extend our work in three directions: further analyzing the rich datasets and investigate more data correlations by including time-dependent cumulative statistics of matches or categorical data; building models to classify team behavior profiles, and understanding the similarities between online and real world team behavior.

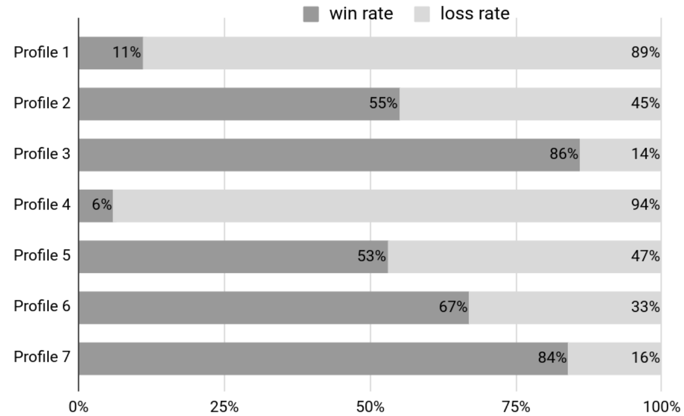


Figure 3: Proportion of winning/losing teams: Profile 1 (win: 662, loss: 5829), Profile 2 (win: 5255, loss: 4273), Profile 3 (win: 6357, loss: 1051), Profile 4 (win: 502, loss: 7676), Profile 5 (win: 4776, loss: 4233), Profile 6 (win: 3650, loss: 1785), Profile 7 (win: 4485, loss: 840).

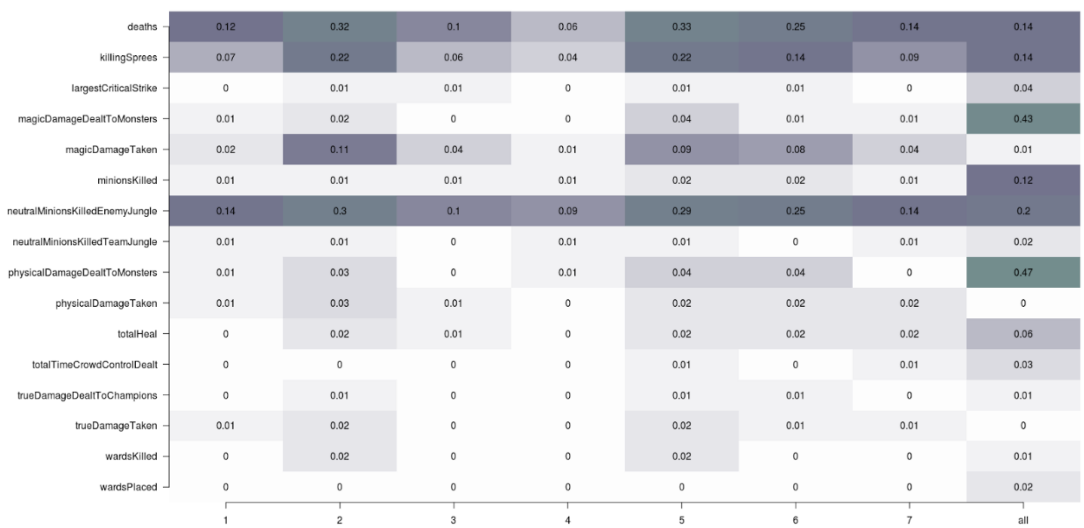
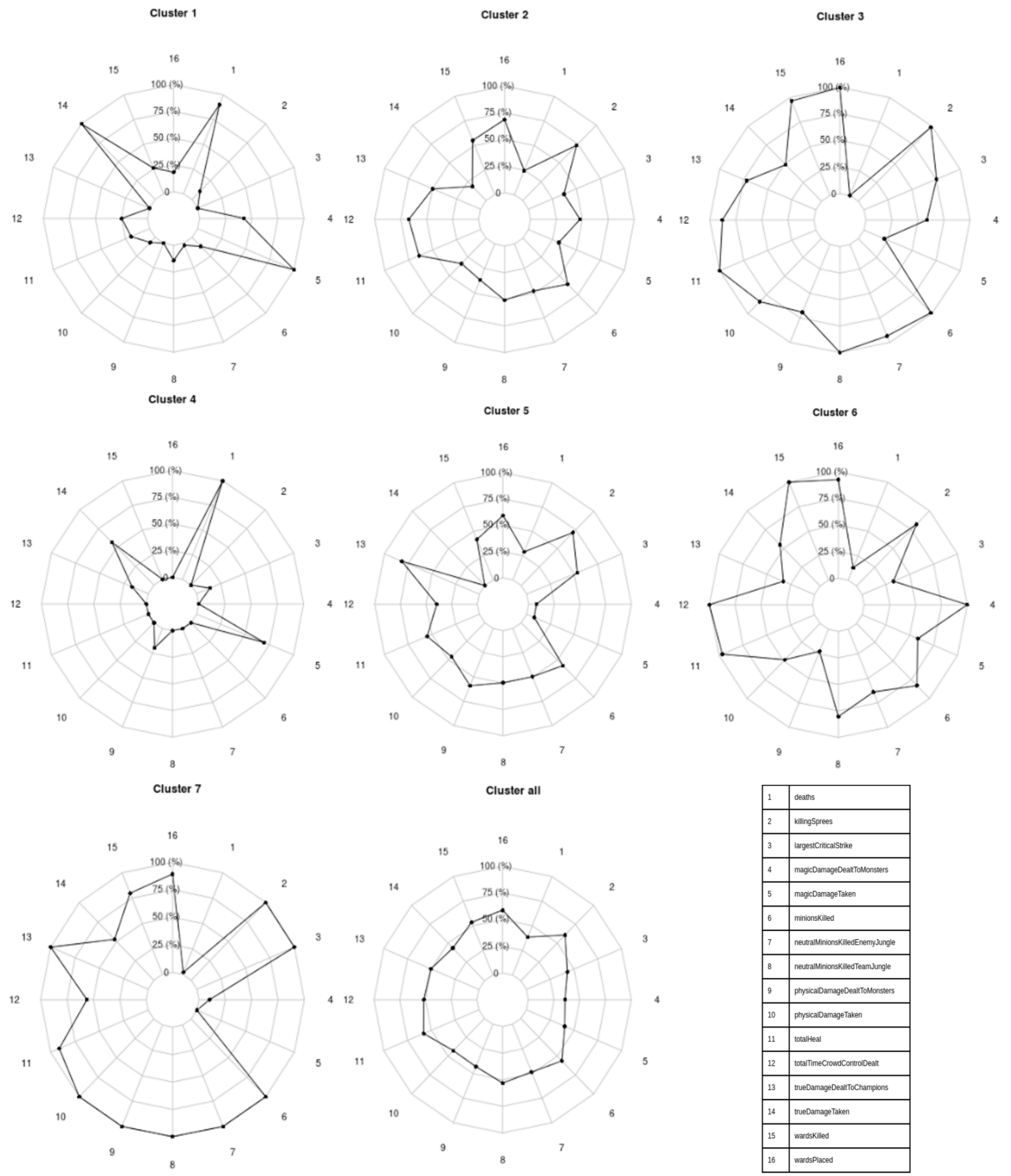


Figure 4: Relevance of dataset features based on information gain. The more intense the color, the more relevant is the feature.

Table 2: Centroids of the features (metrics) for each profile. The *min* and *max* values are indicated by row.

Features	Profile 1	Profile 2	Profile 3	Profile 4	Profile 5	Profile 6	Profile 7	All
deaths	1.17	0.85	0.73^{min}	1.22^{max}	0.87	0.79	0.74	0.92
killingSprees	0.14	0.22	0.26^{max}	0.12^{min}	0.22	0.23	0.26^{max}	0.21
largestCriticalStrike	22.85^{min}	31.63	40.80	26.11	35.52	30.65	46.64^{max}	33.10
magicDamageDealtToMonsters	4805.31	5077.61	5621.34	2662.17^{min}	3043.93	7716.26^{max}	3174.08	4462.33
magicDamageTaken	1187.75	1118.98	1108.49^{min}	1157.09	1095.07^{max}	1145.94	1087.15	1127.58
minionsKilled	15.55	17.73	19.39	15.03^{min}	17.57	18.69	19.45^{max}	17.52
neutralMinionsKilledEnemyJungle	0.20	0.48	0.77	0.18^{min}	0.49	0.58	0.81^{max}	0.48
neutralMinionsKilledTeamJungle	1.93	2.10	2.33	1.87^{min}	2.10	2.24	2.33^{max}	2.12
physicalDamageDealtToMonsters	4661.69^{min}	7414.61	9875.30	6138.25	9103.80	6361.48	12198.87^{max}	7899.13
physicalDamageTaken	1986.43	2014.44	2066.65	1979.37^{min}	2026.98	2029.57	2083.44^{max}	2023.80
totalHeal	489.48	603.20	694.18^{max}	440.94^{min}	580.17	685.86	671.94	587.95
totalTimeCrowdControlDealt	51.15	58.48	62.06^{max}	46.86^{min}	53.69	64.16	56.57	55.78
trueDamageDealtToChampions	85.38^{min}	100.79	107.55	90.51	110.61	95.45	116.25^{max}	100.94
trueDamageTaken	109.38^{max}	103.26^{min}	105.52	106.13	101.85	105.99	105.82	105.12
wardsKilled	0.20	0.23	0.27^{max}	0.17^{min}	0.21	0.27^{max}	0.25	0.23
wardsPlaced	1.62	1.76	1.85^{max}	1.56^{min}	1.73	1.83	1.82	1.73



1	deaths
2	killingSprees
3	largestCriticalStrike
4	magicDamageDealtToMonsters
5	magicDamageTaken
6	minionsKilled
7	neutralMinionsKilledEnemyJungle
8	neutralMinionsKilledTeamJungle
9	physicalDamageDealtToMonsters
10	physicalDamageTaken
11	totalHeal
12	totalTimeCrowdControlDealt
13	trueDamageDealtToChampions
14	trueDamageTaken
15	wardsKilled
16	wardsPlaced

Figure 5: Normalized centroids of performance features

Table 3: Performance score level for Profile 1 and Profile 4.

Score level	Profile 1 and Profile 4	Profile 1	Profile 4
Very low	killingSprees (2), largestCriticalStrike (3), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8), physicalDamageTaken (10)	physicalDamageDealtToMonsters (9), trueDamageDealtToChampions (13)	magicDamageDealtToMonsters (4), totalHeal (11), totalTimeCrowdControlDealt (12), wardsKilled (15), wardsPlaced (16)
Low		totalHeal (11), totalTimeCrowdControlDealt (12), wardsKilled (15), wardsPlaced (16)	physicalDamageDealtToMonsters (9), trueDamageDealtToChampions (13)
Middle		magicDamageDealtToMonsters (4)	trueDamageTaken (14)
High			magicDamageTaken (5)
Very High	deaths (1)	magicDamageTaken (5), trueDamageTaken (14)	

Table 4: Performance score level for Profile 2 and Profile 5.

Score level	Profile 2 and Profile 5	Profile 2	Profile 5
Very low			magicDamageDealtToMonsters (4), magicDamageTaken (5) and trueDamageTaken (14)
Low	deaths (1)	magicDamageTaken (5), physicalDamageTaken and trueDamageTaken (14)	
Middle	largestCriticalStrike (3), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8), physicalDamageDealtToMonsters (9), wardsKilled (15)	magicDamageDealtToMonsters (4) and trueDamageDealtToChampions (13)	physicalDamageTaken (10), totalHeal (11) and totalTimeCrowdControlDealt (12)
High	killingSprees (2) and wardsPlaced (16)	totalHeal (11) and totalTimeCrowdControlDealt (12)	trueDamageDealtToChampions (13)
Very High			

Table 5: Performance score level for Profile 6.

Score level	Profile 6
Very low	
Low	deaths (1), largestCriticalStrike (3), physicalDamageDealtToMonsters (9), trueDamageDealtToChampions (13)
Middle	magicDamageTaken (5), physicalDamageTaken, trueDamageTaken (14)
High	killingSprees (2), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8)
Very High	totalHeal (11), totalTimeCrowdControlDealt (12), wardsKilled (15), wardsPlaced (16)

Table 6: Performance score level for Profile 2 and Profile 5.

Score level	Profile 3 and Profile 7	Profile 3	Profile 7
Very low	deaths (1)		magicDamageDealtToMonsters (4), magicDamageTaken (5)
Low		magicDamageTaken (5)	
Middle	trueDamageTaken (14)	magicDamageDealtToMonsters (4)	totalTimeCrowdControlDealt (12)
High		largestCriticalStrike (3), physicalDamageDealtToMonsters (9), physicalDamageTaken (10), trueDamageDealtToChampions (13)	wardsKilled (15)
Very High	killingSprees (2), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8), totalHeal (11), wardsPlaced (16)	totalTimeCrowdControlDealt (12), wardsKilled (15)	largestCriticalStrike (3), physicalDamageDealtToMonsters (9), physicalDamageTaken, trueDamageDealtToChampions (13)

ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support received from CAPES/Brazil.

REFERENCES

[1] Alexandra Buchan and Jacqui Taylor. 2016. A qualitative exploration of factors affecting group cohesion and team play in multiplayer online battle arenas (mobas). *The Computer Games Journal* 5, 1-2 (2016), 65–89.

[2] Olivier Cavadenti, Victor Codocedo, Jean-François Boulicaut, and Mehdi Kaytoue. 2016. What Did I Do Wrong in My MOBA Game? Mining Patterns Discriminating Deviant Behaviours. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*. IEEE, 662–671.

[3] Anders Drachen, Matthew Yancey, John Maguire, Derrek Chu, Iris Yuhui Wang, Tobias Mahlmann, Matthias Schubert, and Diego Klabajan. 2014. Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (dota 2). In *Games media entertainment (GEM), 2014 IEEE*. IEEE.

[4] Robert Edge. 2013. *Predicting Player Churn in Multiplayer Games using Goal-Weighted Empowerment*. Technical Report. Technical Report 13-024.

[5] Christoph Eggert, Marc Herrlich, Jan Smeddinck, and Rainer Malaka. 2015. Classification of player roles in the team-based multi-player game dota 2. In *International Conference on Entertainment Computing*. Springer, 112–125.

[6] Forbes 2017. *New Report Details How eSports Is An Effective Engagement And Marketing Tool*. Forbes. https://www.forbes.com/sites/insertcoin/2015/02/25/new-report-details-how-esports-is-an-effective-engagement-and-marketing-too.

[7] IFPI 2017. *Global Music Report 2017*. IFPI. http://www.ifpi.org/downloads/GMR2017.pdf.

[8] Filip Johansson and Jesper Wikström. 2015. Result Prediction by Mining Replays in Dota 2. (2015).

[9] K Kalyanaraman. 2014. *To win or not to win? A prediction model to determine the outcome of a DotA2 match*. Technical Report. Technical report, University of California San Diego.

[10] Young Bin Kim, Shin Jin Kang, Sang Hyeok Lee, Jang Young Jung, Hyeong Ryeol Kam, Jung Lee, Young Sun Kim, Joonsoo Lee, and Chang Hun Kim. 2015. Efficiently detecting outlying behavior in video-game players. *PeerJ* 3 (2015), e1502.

[11] Nicholas Kinkade, L Jolla, and K Lim. 2015. *DOTA 2 Win Prediction*. Technical Report. tech. rep., University of California San Diego.

[12] Trupti M Kodinariya and Prashant R Makwana. 2013. Review on determining number of Cluster in K-Means Clustering. *International Journal* 1, 6 (2013), 90–95.

[13] League of Legends 2017. *Game Info*. League of Legends. http://gameinfo.na.leagueoflegends.com/en/game-info.

[14] MPAA 2017. *2016 Theatrical Market Statistics Report*. MPAA. http://www.mpaa.org/wp-content/uploads/2017/03/MPAA-Theatrical-Market-Statistics-2016_Final.pdf.

[15] Julia Neidhardt, Yun Huang, and Noshir Contractor. 2015. Team vs. Team: Success Factors in a Multiplayer Online Battle Arena Game. In *Academy of Management Proceedings*, Vol. 2015. Academy of Management, 18725.

[16] NewZoo 2016. *2016 Global Games Market Report*. NewZoo. https://cdn2.hubspot.net/hubfs/700740/Reports/Newzoo_Free_2016_Global_Games_Market_Report.pdf.

[17] Hao Yi Ong, Sunil Deolalikar, and Mark Peng. 2015. Player Behavior and Optimal Team Composition for Online Multiplayer Games. *arXiv preprint arXiv:1503.02230* (2015).

[18] Nataliia Pobiedina, Julia Neidhardt, Maria del Carmen Calatrava Moreno, and Hannes Werthner. 2013. Ranking factors of team success. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 1185–1194.

[19] Riot Games 2017. *Our games*. Riot Games. http://www.riotgames.com/our-games.

[20] François Rioult, Jean-Philippe Métivier, Boris Helleu, Nicolas Scelles, and Christophe Durand. 2014. Mining tracks of competitive video games. *AASRI Procedia* 8 (2014), 82–87.

[21] Matthias Schubert, Anders Drachen, and Tobias Mahlmann. 2016. Esports analytics through encounter detection. In *Proceedings of the MIT Sloan Sports Analytics Conference*.

[22] Superdata 2016. *Worldwide digital games market: 2015 total*. Superdata. http://www.superdataresearch.com.

[23] Chengwei Xiao, Jiaqi Ye, Rui Máximo Esteves, and Chunming Rong. 2015. Using Spearman's correlation coefficients for exploratory data analysis on big dataset. *Concurrency and Computation: Practice and Experience* (2015).

[24] Pu Yang, Brent E Harrison, and David L Roberts. 2014. Identifying patterns in combat that are predictive of success in MOBA games.. In *FDG*.

[25] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. 2014. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press.