META Discovery And Role-based Matchmaking (MEDIROMA)

Thesis Report

Submitted by

Aleo Aninda

Supervisor

Dr. Brett Wilkinson

Submission date

11, November 2019

Submitted to the *College of Science and Engineering* in partial fulfilment of the requirements for the degree of *Master of Science (Computer Science)* at Flinders University – Adelaide, Australia.

Declaration of originality

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signature: 

Date: 11/11/2019

Abstract

Player engagement and satisfaction has been one of the top priorities for game developers of competitive multiplayer games. One of the challenges for the developers is to ensure the balance between fairness and satisfaction for every match that the player may experience. However, ensuring satisfaction can be difficult since many factors play into how the player experiences the game. For example, a player may encounter team mates who would get into a conflict for having an overlapping choice of roles within the game. This would result in either a compromise or disagreement that would lead to decreasing the team's chances of winning. However, this can be avoided if the matchmaking solution takes into account of the player's role preference before placing them into the teams. This would avoid any arguments and give the players more satisfaction with their games. To achieve such a solution, the developer must identify the roles that fit into the structure of the game that guarantees a moderate to high chance of winning a game. Such a role composition is defined by the Most Effective Tactical Advantage (META) of the game, which can be mined from existing dataset of match statistics for that game. The research presented here proposes a model which utilises the META of the game and allocates players based on their role preference. This solution has been achieved by using feature space exploration routine, such as Mean Shift clustering algorithm and identifying the cluster that represents the META of the game. The META information from that cluster is then passed to the matchmaking solution, which then leverages that information to allocate players by looking at their past matches to predict their preferred role. This overall model would ensure player satisfaction along with fairness of the game. The model was applied on League of Legends, using a match statistics dataset of over six thousand matches. It produced a META cluster, which has a composition of roles that has an 87.65% chance of winning a game. Afterwards, a positive feedback was received from League of Legends players who attempted to follow the role composition given by the META cluster. Based on this feedback, it can be postulated that the model was able to deliver on its promised goals of improving player satisfaction, and this can be emulated in the live build of a competitive game like League of Legends.

Acknowledgements

For the past 12 months, this has been a passion project that helped me gain a significant amount of knowledge in data science and video games service components. My enthusiasm for working on this project has been greatly motivated by my supervisor Dr. Brett Wilkinson, who helped me to explore various researches in the video games industry. His valuable time invested into my thesis has helped me keep on track of my deliverables and reach my goals.

I am also grateful to Oracle Elixir for providing with very detailed datasets of match statistics that has been used for the evaluation of this thesis. Their feature rich datasets helped me to investigate further into the different possibilities of extracting the desired results.

Lastly, I would like to thank my friends Anson Mahindra, Hugo Chang, Andres Siblez, Bryan Roel Elivera, Juni Rivera, Gabriella Fraser, Alexia Rossignol, Sunny Kang and Marcus Ma for inspiring me to work on this project and helping me learn about the game League of Legends. Their expertise of the game has helped me to understand the player's perspective of the game and shape the model to improve player satisfaction.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Motivation

A matchmaking system is a service within a multiplayer video game that allows players to find other players based on one or more criterion such as skill rating, game settings preference, and network latency. The service itself could be divided into several components, where each unit serves a different purpose based on the requirements of the game developers. Due to the wide variety of online games available, matchmaking solutions may differ in games and are subjected to the genre, player and developer expectations.

A proper implementation of a matchmaking service improves engagement of players in the game and creates the financial opportunity for the publisher and developer to further invest into the game. If the solution is inadequate in providing a fair environment for all players, then the player population would fall, and less concurrent player number leads to the slow death of the game. As a result of this, queue times (waiting to be placed into a game) would become high and existing players may not be able to find enough players in their own regions, forcing them to region hop and create undesirable high latency impact on the other players. This domino effect can be avoided if the developer is proactive in creating a matchmaking solution that would satisfy the needs of the players.

Competitive video games like the popular League of Legends rely on their matchmaking service to ensure a fair placement for its players and create an enjoyable experience. In this game, a team of 5 players competes against another team of 5 players, where each of them picks an in-game avatar called a ‘champion’, each of which possesses unique skills. Each team starts the game on opposite sides of the map, designated as Dire and Ancient. Both of their objectives are to defend their strategic towers and protect their core building in their base called ‘Nexus’. The first team to destroy this Nexus of the enemy team wins the game. Aside from destroying enemy towers and the core building, the players must eliminate the opponents to earn in game credits called ‘gold’ that can be used to buy and craft in game items which gives additional boost to their abilities or adds new skills. After every elimination, the players respawn back to their base after a certain amount of time.

To achieve their objective, the players must carefully choose a champion that will complement each other’s strategies and help in creating tactics that will give them an advantage over their enemy team. Even though every team may have a unique approach to game strategies, there is always a general set of tactics and composition followed by majority of teams to achieve a victory. For example, in the game of Cricket, a team may choose to follow a composition of six batsmen, three spinners and two medium fast bowlers for a pitch that has been affected by rainfall, where their strategy is to use the slow field to their advantage in their bowling inning. However, this

composition would not be suitable in a dry flat wicket, where fast pace bowlers would have an advantage. Furthermore, the type of game mode such as Test match or One Day match would also change the strategies followed by the teams, where they could opt for a strategy that would have more emphasis on endurance for a Test match, or more aggression for a One Day match. Similarly, in League of Legends, the players use different composition of champions that is catered towards a specific strategy, which increases their chance of winning. For example, the players may opt to choose a champion composition of three AD Carry, one Support and one Jungle for an aggressive playstyle. The set of actions, process, and tactics that increases the chance of winning a game is commonly known as Most Effective Tactical Advantage (META). The META adopted by the players keep changing, as they learn new strategies that combines the special abilities of the champions within a set of five member team. With every new update and features that gets introduced into the game, the META of the game also changes, which forces the players to learn and adapt to the new META.

1.2 Purpose

One of the major obstacles in maintaining a consistent META is the hurdle that solo queue players (players who start a matchmaking request without joining a squad) face upon coming into a squad that may have overlapping role preference. For example, a solo queue player may want to pick a champion that another player in that same squad is proficient at playing. Since the same champion cannot be picked by more than one player, the players with overlapping preference may argue, which can either end in a compromise, or end in a toxic outcome that results in most likely defeat. In either case, it is not a comfortable experience for either players and leads to frustration over the game. Furthermore, since one of the players must compromise from their preferred role, it is possible that they may need to play a META that they are not accustomed to, which also leads to diminishing their probability of winning the game. Such incidents could have been avoided if the matchmaking solution would take into account of the player's role preference before placing them into the game, in order to avoid competition within the team for the same roles.

Competitive games like League of Legends have developed professionally curated sports events, commonly known as esports, surrounding the player community. For the past 10 years, the League of Legends esports scene has grown significantly, with the highest-level tournaments granting over US$6 million to the winning team. Even in Australia, the esports scene has seen a massive increase, with 41% of the players claimed that they are viewers of these tournament, as shown in [1]. Due to such high stakes in lucrative tournaments, it has become important for the developer to maintain a steady player base for the game, while ensuring that they can introduce new refreshing content without disrupting the pro league scene. Keeping the game balanced while also introducing new features can be very tricky if the developer doesn't have the META information of their game. Therefore, discovering the META of the game is a crucial process that game designers and developers must be aware of and actively pursue.

1.3 Outline

Based on the scope of this problem in League of Legends, this research proposes META Discovery and Role-base Matchmaking (MEDIROMA) model as a solution. In the following sections, the concepts behind META discovery and matchmaking solution will be discussed. In Chapter 2, related work based on matchmaking solutions have been described, analysed and critiqued in detail. In Chapter 3, the research proposal for the project and its methodology has been laid out. Chapter 4 explores the results obtained from the implementations of the solution and whether it produced satisfactory results. In Chapter 5, the drawbacks and limitations of this solution are discussed. Finally, in Chapter 6, this paper is concluded with proposals for future expansion and improvements.

Chapter 2: Related Work

The majority of multiplayer games have focused on the estimation of skill rating as the primary criterion for balancing games with multiple players, which has thus generated many implementations of matchmaking solutions based on Bayesian estimation and factor graph. Other solutions have investigated the latent features of the games based on their genre, such as identifying team contribution of the individuals, impact of certain roles played by the individuals, and improving engagement of the players by dynamically adjusting the placements of the players. In this chapter, the discussion starts with the more mainstream implementations of matchmaking solutions, which heavily emphasizes the priority of skill rating above everything else. Afterwards, research papers which highlights the importance of latent features such as player retention, role preference, satisfaction and engagement are discussed. Some of the less traditional solutions for matchmaking solution are discussed later, which focuses on limiting network latency between players using various approaches such as utilising cloud platforms, peer to peer network and artificial latency adjustments for fairness between the players. Finally, studies on current matchmaking implementations for popular games like League of Legends are discussed.

One of the most popular skill rating system for online multiplayer games is called TrueSkill™, which heavily relies on skill ratings of players, that updates after every match [2]. This new skill rating is a heavily modified version of the Elo ranking system created for Chess, where instead of matching two players, the TrueSkill matches two teams consisting of multiple players competing against each other. The TrueSkill uses a factor graph, which has defined parameters for individual and team performance and skill, as well as a message passing algorithm that calculates the probability of one team wining, losing and drawing against the other. The formula behind the message passing algorithm is based on Bayesian mathematical formula and helps to update the skill ratings of each player after every game. Dangauthier et al. further extended their existing solution of TrueSkill™ from [2] in [3] to include the time varying attribute that determines the skill rating of an agent. They proposed a solution that takes the skill rating of an agent from a date range and creates a time series of skill change for that agent. They applied their new solution to the skill rating chess dataset of players, and presented the skill change of the chess players over the years, and the overall change in strength of the players.

Other Bayesian solutions proposed a matchmaking solution which emphasizes the utilisation of score values from results of games to determine the skill rating of players [4]. Their objective was to create a system for games where results are heavily dependent on the numerical scores of the game rather than just the three possible outcomes (win, loss, or draw) of the game. They argue that this would lead to a more balanced matchmaking, since the players would be allocated in such a way that would ensure their contributions into each game has been quantified into their skill rating.

They modified the popular TrueSkill™ matchmaking solution to consider the offense and defence scores of teams and use factoring graphs to update the skill rating of the individual players. Skill rating has also been applied in physical sports such as tennis, which uses a variant of Expectation Propagation to estimate player strength as shown in [5]. Birlutiu et al. in [5] used the Expectation Propagation-Correlated variance to compute a posterior distribution over player's strength, from which the mean is used to make an inference of the player's strength. According to the author, using Bayesian techniques to estimate player strength is hard to control even with a small number of players. S. Nikolenko et al. have improved upon the existing model of TrueSkill™ in [6], which was made using the factor graph structure, that dramatically improved upon the estimation accuracy of the original TrueSkill™ algorithm. According to the author, their solution in [6] solves the limitations of TrueSkill™ in the areas of multiway ties and with variable team sizes. Furthermore, their new algorithm on the factor graph structure significantly improves the performance compared to the original model.

While many matchmaking algorithms focuses on achieving fairness based on the skill level of players in a video game, Chen et.al. proposed a solution in [7], that increases the probability of player retention and engagement for a multiplayer PvP (player-versus-player) game. They focused their attention on statistics such as churn data, which is the proportion of players who leave the game for a period, and the status of the players such as winning streak, losing streak or a combination of both. Their Engagement Optimisation Matchmaking (EOMM) model is focused on solving matchmaking as an optimisation problem, rather than a regression problem on other skill-based solutions. Furthermore, it is asserted that skill-based matchmaking is a subset of the solution provided by EOMM and can also provide tuning for other properties such as total time and/or money spent in game. Another novel solution for matchmaking where skill level of players is not the sole attribute to determine the placement of matches, was the use of playstyle, roles and preferences within a team to determine their matchmaking, as proposed by Stroh-Maraun et al. in [8]. This paper effectively delves into the psychological behaviour of the players and tries to model it into a quantifiable solution. They narrowed down their research into the following concepts: round level characteristics, individual habits, skill levels and matchmaking algorithms that can impact on player retention. These attributes help them create a matchmaking solution that will assemble a balanced team with players from a diverse range of playstyle and preference, while also ensuring an opponent team is selected closely matching their skill level. Delalleau et. al. discusses in [9] the drawbacks of simple skill-based matchmaking where a single metric determines their ranks, and instead proposes a solution that considers the skills like reflex, planning and teamwork that is being utilised by the player. They used machine learning algorithms, specifically neural networks, to predict the match winner and measures the enjoyment of the player. Various statistics from the match and the player's performance are used to derive these values that can be used to create a broader picture about the player rather than a simple

rating. The authors heavily emphasised that fun factor is as important as balanced gameplay, since equally matched players may find their games together boring.

Various solutions utilising factor graph to identify latent features of the game and contributions of the agents themselves have been published in several papers. Zhang et al. proposed a context-based skill rating solution in [10], which uses a factor-based model to calculate skill ratings of the players. They argued that player skill should not be generalised based on a single outcome, but rather calculated based on the context of the events. According to the authors, this problem can be solved by assuming the individual player skills as a product of the metrics of context factor and the agent factor. Furthermore, they applied collapsed Gibbs sampling algorithm to make estimations in their model, which allowed them to predict the outcome of any given sample based on the context sensitive data. Huang et al. proposed in [11] a generalized version of the Bradley Terry model, which compares and identifies individual ratings from the team ratings. Their new algorithm extracts the individual skill from the estimations of the team ratings. With this generalised approach to estimating the skill ratings, the authors believe that it can be applied to applications which consists of more than two agents on opposing ends, as well as team compositions that can have multiple agents.

Some papers have presented a new approach to measure team performance, as proposed by DeLong et al. in [12], where the performance of the individual players impacted upon the performance of other players within a team is quantified and measured. They argued that simply factoring skill ratings of the individual players to determine the overall skill of the team does not produce the correct estimation of skill level of the teams. They defined the term 'team chemistry' as the measure that needs to be quantified for proper estimation of team skill levels. Other skill rating systems such as Elo, Glicko, and TrueSkill™ were used as base learners, while they leveraged four different techniques to measure the subset performances of the players to determine the 'team-chemistry' of the teams. Furthermore, DeLong et al. introduced a new variant of TeamSkill which takes into account game-specific performance measures as features into its existing skill prediction rating model for teams in [13]. According to the authors, this new variant called TeamSkill-EV Mixed outperforms all prior approaches of estimation models on team competitions. However, this variant only works if the teams are evenly matched, which first needs to be classified based on the threshold of probability of a team winning over the other. DeLong et al. further extended their model of TeamSkill from [12] to analyse and estimate skill of players from the data obtained from National Basketball Association on basketball matches [14]. They applied their model to find the 'team chemistry' within the basketball teams and identify the contributions of the individuals within the teams towards the outcome of the game. Menke et al presents a model in [15] that evaluates the contribution of individual players within a team and how much influence they played in the outcome of the game. Furthermore, their model can estimate properties about the game itself. Using these estimates, the authors claim that more engaging games can be made in the future, which will help developers to create games that can retain their player base. Huang et. al.

presents an exponential model to rank individuals from group competitions in [11]. In this paper, the authors argued that even though some form of average point system would indicate the performance of an agent, it still does not reflect their true ability since it doesn't take into account of their opponent's abilities. While other games do not keep track of individual ratings at all and the raw score of the teams are used to assert the outcome. Thus, they proposed a model that uses two convex minimization formulas to estimate individual ratings and applied this solution to the card game of bridge.

Some of the solutions have investigated difficulty adjustment, where Sarkar et al implemented a difficulty optimization solution for Human Computational Games (HCG) using matchmaking algorithms like Glicko 2 in [16]. Their goal was to increase engagement from volunteers participating in HCGs, while also ensuring greater number of difficult tasks are performed. Their solution is based on the proposal of Cooper et al. in [17], which proposed the solution but did not give empirical evidence of the solution. Sarkar et. al. expanded the solution in [16] to include the concept of improving engagement from volunteers and measured its performance against randomly generated difficulty levels and ordered difficulty levels. They found that both matchmaking and sorted ordered difficulty tasks improved engagement from players, while matchmaking and randomly ordered difficulty tasks increased the number of tasks that were completed. Baldwin et. al. proposed a dynamic difficulty adjustment for multiplayer games to increase engagement of lower skilled players in an online environment in [18]. The authors argued that compared to single player games, where difficulty can be dynamically adjusted to suit the skill level of the player, the same cannot be applied on multiplayer games since all the agents are human controlled. Therefore, they proposed a solution where lower skilled players will be given more opportunities and abilities than their higher skilled counterparts.

Proposals have been made for a role-based matchmaking solution by Myślak et al. in [19] which heavily emphasizes on the most effective tactics available of the video game itself. They implemented a solution where solo queuing players are placed in a such a way that the team is distributed proportionately according to the role they play. This ensures more coordination among the team members and greater engagement of players. Furthermore, it allows a solo queuing team a better chance at winning the game against a full squad. The authors used k-means clustering algorithm to identify the distinct roles played within a game of League of Legends and used it to predict how likely a team is going to win based on the team composition. Suznjevic et. al. proposes a matchmaking solution that takes into account individual player performance and based on that rewards the appropriate ratings to the player instead of adjusting the ratings of all players based on the outcome of the matches in [20]. Their solution, known as ACARI (**A**pplication **C**ontext **A**ware **R**ating algor**I**thm) calculates player rating based on their performance and calculates their rating separately based on the role they played. This allows the system to determine the real skill of the player based on the role performed by the player.

Matchmaking solutions focusing solely on the network and latency have also been made, proposed by Yitong et al. in [21], where they used the cloud platform as the game server as well as the rendering server for the clients. This matchmaking solution only deals with allocating players from a player pool within the game servers and rendering servers and does not take into account the player's skill level, playstyle, or any other attributes of the players themselves. The scope of their solution is limited to ensuring each player is connected to only one game server and one rendering server, while complying with minimum latency standards as required by the game itself. Agarwal et al. have presented a matchmaking solution that allocates players in a multiplayer match based on their latency to each other on a peer to peer system in [22]. The authors created a system that estimates the latencies between the players before being placed into a cluster of similarly matched latency. They named their system Htrae, which uses geolocation of players using network coordinates, that allows the system to predict whether the players would have lower latency when placed together in a match. Zander et. al. proposed a new solution to latency imbalance between the participants in a multiplayer video game in [23]. They described how unfair a match can become for a player when they are further away from the server, causing them to experience delays in sending and receiving data from the game server. This leads to giving advantage to players with closer proximity to the server to gain an advantage over the other players and creates frustration for those who experience delays. Thus, the authors argued that the best way to create a fair environment would be adjusting the latency 'artificially' for the players in close proximity to servers, so that all participants have a fair chance of competing in the game. Manweiler proposed a matchmaking solution for multiplayer mobile games which considers the latency of the cellular network and adapts to the pre-sets of the game as required by the players in [24]. According to the authors, the experience in a multiplayer game on cellular network can be hampered due to disparity of latency on mobile internet and create an unfair environment for some of the players. Therefore, they suggested a solution which estimates the network performance between the competing players and making sure that this estimation can be calculated as quickly as possible. Once the estimation has been calculated, the system then allocates the players into groups according to their network performance to create a fair environment for everyone. Boroń et. al. proposed a peer to peer matchmaking solution that utilizes a shared resource management architecture between the players in [25]. The authors argued that due to high cost of server rentals, many developers are discouraged from creating and hosting dedicated servers. However, if the players contribute their unused computational and network resources towards the P2P matchmaking solution of the game, then the cost of running a dedicated server won't be incurred and the developers would be encouraged to create more multiplayer games. Thus, the authors introduced a platform called SelfAid, which allows the developers to take advantage of the unused resources, while they can define the matchmaking parameters on their end.

Some papers have attempted to study the existing matchmaking solutions, where one Claypool et al presents a detailed study on the current matchmaking system of the

popular MOBA (multiplayer online battle arena) game League of Legends in [26]. In this paper, the authors conducted a survey on the players and asked them their opinions on the matchmaking solution provided by League of Legends. The study's aim was to investigate the satisfaction of the player base and the expectation from the matchmaking solution of the game. In their study, interesting data emerged where even though games are balanced according to the rank of the players, the players still believe it was unbalanced. On the other hand, actual unbalanced games have been reported to have been more enjoyable, only if they have won those games. Véron et. al. presented a paper in [27] that studied the matchmaking service offered by the game League of Legends and gave a detailed analysis on the matchmaking system, along with suggesting improvements. They obtained their data from the publicly available dataset of League of Legends, and categorised them into three categories: avatar information, company handlers, and matchmaking data. Then they used these data to evaluate the quality of the matchmaking system in terms of waiting time, match evenness and response time.

Matchmaking solutions aimed towards other applications have been proposed by various authors. Shafran et. al. proposed a new matchmaking solution for a general Multi Agent System in [28], which can accommodate a dynamic pool of agents which continuously looks for other agents based on the predetermined requirements. They claim that their solution would help agents find other agents within a very short time constraint due to its use of cache memory that allows the system to lookup other agents quickly. Ogston et. al. performed a simulation on their proposed matchmaking system for Multi Agent System in [29], which allows the agents to find other agents within their nearest neighbours, and they try to match with at least 10 to 100 other agents to create a cluster. Furthermore, no predefined parameters need to be set for the matchmaking to proceed, while the cluster size can be changed as per required. Sycara et. al. proposed a matchmaking solution for software agents in a web environment in [30], which allows various services to communicate with other relevant components more efficiently. Their proposed matchmaking solution uses their proprietary language called LARKS, which allows software agents to broadcast and make request to other agents for their services more efficiently. It performs and applies several filters to its matchmaking solutions such as context matching, profile comparison, similarity matching, signature matching and constraint matching.

The literature review can be summarised in three key points: a) fairness and satisfaction is the primary goal of any matchmaking solution, b) depending on the genre of the game, the matchmaking solution needs to be tailored, and c) existing matchmaking solutions needs to be upgraded to allow for expansion with new features of competitive games. Most of these solutions heavily relied on the utilisation of factor graphs and Bayesian Distribution for determining the skill rating of players and teams, while some looked at improving player retention and engagement. Research works which prioritises on finding out latent features of multiplayer games motivated this research to further expand investigation into the problems that simply cannot be labelled using predictive models and statistical distributions. One of these problems that caught

the attention of this research was the solution of Myślak et al. from [19], which proposed a solution for solving the overlapping role preference issue in League of Legends using a role-based matchmaking solution. Even though they had a good grasp of the problem, their approach to solving it was based on the assumption of a false META, and a poorer choice of a data mining routine with K-means clustering. In Chapter 3, [19] is further broken down to identify its issues and explain the proposal of this research to solve those problems.

Chapter 3: Research Proposal

3.1 Inspiration

The research presented here proposes to expand and improve the concept and approach demonstrated in [19]., where they presented the idea of allocating players via a matchmaking solution based on their role preference. Their motivation for the research came from the idea that conflicts between the players regarding role composition can be preemtively avoided if their role preference is taken into account during matchmaking. They used the popular multiplayer video game League of Legends as an example to show its weakness in managing conflicts when it comes to players who solo queue for a match. The conflict arises when the role preference of players overlap with each other and causes compromises within the team to adapt to a non-standard META that reduces their chances of winning.

In [19], the authors proposed the concept of taking into account the player's role preference during matchmaking and allocating them in such a way that their preference is not overlapped by others. For example, in League of Legends, the standard META of the game consists of the following roles: Top, Mid, Jungle, AD Carry and Support. Therefore, their matchmaking proposal would ensure that players with a preferred role preference such as Top would not be placed in a group with players who also have their role preference as Top. They based their result on the match statistics obtained from the developer of League of Legends and postulated that teams that play standard META had a better chance of winning than teams that play non-standard META.

The critical flaw in their model is the assumption that the presumed 'standard META' represents the actual META of the game. The standard META is the de facto META that is typically agreed upon by the community of players after they have invested a significant amount of time into the game. Their perception of the META relies on their own experience and is judged differently at different skill levels of the community. However, this standard META has no hard evidence to prove whether it is the actual META of the game and whether it can guarantee a winnable result in the majority of games. Without proper investigation into discovering the META of the game, the role-based matchmaking could be allocating players based on a misguided role specification.

3.2 Proposal

The research presented here proposes a solution to find the actual META of the game instead of making assumptions about the META. One of the empirical approaches to discovering the META is to analyse match statistics that consists of feature information such as the composition of the teams, contributions made by each of those roles, and outcome of the matches. To ensure an automated and unbiased analysis of the statistical data, a data mining solution can be applied to the given dataset, that will extract relevant information regarding the key factors that influences the outcome of the

game. For example, in League of Legends, a data mining routine would look into the team compositions and identify which sets of ‘champions’ (in-game characters with unique abilities) impacts the results of the games played. Discovering these sets of champions would help in finding the actual META of the game, that can be used by the matchmaking solution for allocating the players to create a more balanced game.

3.3 Methodology

The model for META Discovery and Role-based Matchmaking (MEDIROMA) consists of two components: i) Discovering the META of the game through feature space analysis of an existing dataset of match statistics, and ii) Allocating the players through a matchmaking service that takes into account of the composition of roles defined by the META obtained from feature extraction. The two components are complemented by various sub-components that allow for a robust solution. Figure 1 shows the overall workflow of the MEDIROMA model.

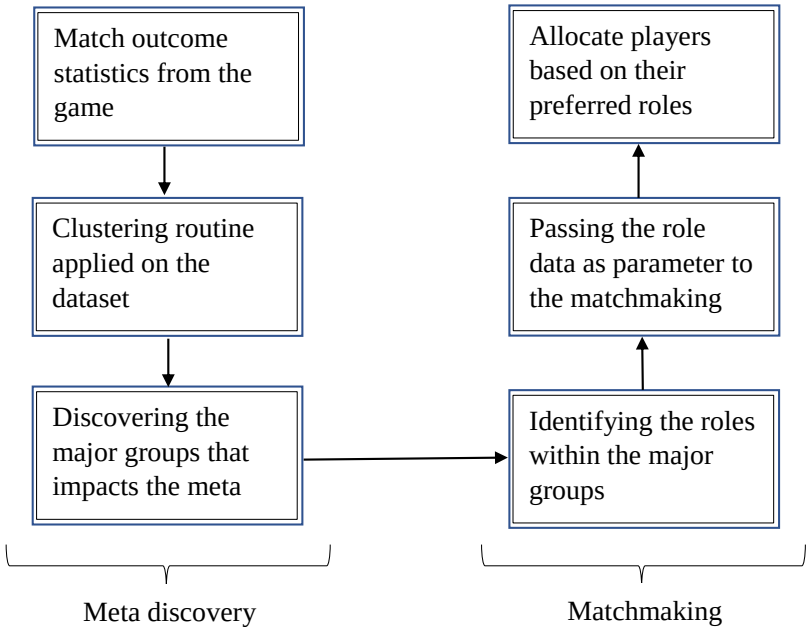

Figure 1 MEDIROMA process workflow

In the following sections, the sub-components and their implementations will be discussed in detail.

3.3.1 Data Mining Routine

The data mining solution adopted for the model of MEDIROMA is a non-parametric feature space analysis created by Comaniciu et. al in [31], called Mean Shift Clustering. This solution allows for very limited intervention and is a valuable tool to identify clusters within a given dataset without the need of passing the expected number of clustering groups. Myślak et. al in [19] alternatively used K-means clustering, albeit their reason for applying a clustering routine was to verify their initial hypothesis for the existence of the 5 roles within the game. However, since K-means is a flat clustering solution that requires parameters to be passed for the expected number of clusters, this makes it unreliable in extracting the correct number of features which might produce

incorrect result, as explained by Ren et al. in [32]. For the case of Mean Shift clustering, the only parameter that is required from the user is the bandwidth size (which is the size of the kernel function) for the exploration of the feature space and has very limited impact on the outcome of the feature extraction.

3.3.2 Data Pre-processing and Feature Selection

The dataset for the match outcome statistics for League of Legends was obtained from Oracle Elixir[1], an organisation that collects and manages League of Legends match statistics at the highest level of competitive tournaments. The matches represented in the dataset have been played during the year 2018, which consists of over six thousand matches played over two seasons and the world championship. Since any competitive game should be balanced at the highest skill level of the game, the choice of dataset would ensure that the feature space extracted would reflect the meta for that skill ceiling.

There are 97 features within the dataset that represents the attributes, actions and outcome for all the 6,000 feature sets. Since the goal of the feature space extraction is to discover the meta that allows a team to win, a feature selection routine is applied on the dataset to determine the ranking of the features based on their impact on the outcome of the game. For feature selection, Univariate Feature Selection was applied on the dataset, that produced a ranking of all the available features. Only the top 50 of these features are then used for the feature space analysis.

Using the open source library of the Pandas DataFrame from Python, most of the data pre-processing was conducted to truncate empty cells, remove undesired features, and insert additional features when required. Furthermore, all non-numerical data were converted to numerical format to allow the feature space exploration routine to easily manage and calculate the nearest distance for each of the feature set. However, a copy of the original dataset is also preserved, which can then be used as a baseline for meta feature extraction in the later stages.

3.3.3 Process of Cluster Analysis

Mean Shift clustering is used as the feature space exploration routine to discover the clusters from the dataset. As a hierarchical clustering routine, Mean Shift does not require any parameters and can independently discover the clusters. The open source library of Scikit-Learn from Python is used to import the Mean Shift routine and apply on the modified dataframe produced by Pandas. Only the features selected from the Univariate Feature Selection are chosen to be mined from the original dataframe. Once the routine produces clusters, each of the clusters are then passed through a predictor that calculates the win rate based on their defined outcome in the feature set. The cluster with the highest win rate is then chosen as the representation of the META cluster for the game, and then its composition of champions is extracted. The set of these champions are then used to find the role mapping for each of these champions, which

[1] https://oracleselixir.com/

is then sent to the matchmaking service as a parameter for ensuring role-based matchmaking. Figure 2 shows the workflow of the META discovery process.

Table 1 Role mapping to champions. From the set of roles, each role Ri can be mapped to one more champion Hj

Roles	Champions
R_1	H_4
R_2	H_3
R_1	H_7
…	…
…	…
R_i	H_j

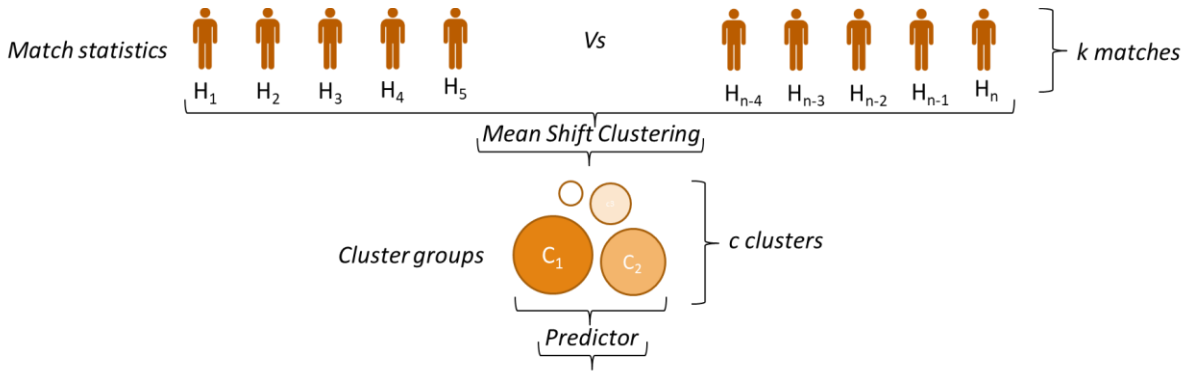

Figure 2 META discovery process of MEDIROMA

3.3.4 Matchmaking Process

Based on the information of the META group, the matchmaking solution allocates the players based on their preferred role. It looks at each player's records on the amount of time spent on each champion and estimates which champion they are likely to pick. The matchmaker would look at the list of champions played in the last 10 matches and assign the role of that champion that has been picked the most to that player. Afterwards, the matchmaker would put that player in a pool of players belonging to that role, from where they will be picked again by the matchmaker to create a team that needs the role to be fulfilled. The matchmaker would ensure that team allocation aligns with the META group. The role-based matchmaking process has been visualised in Figure 3.

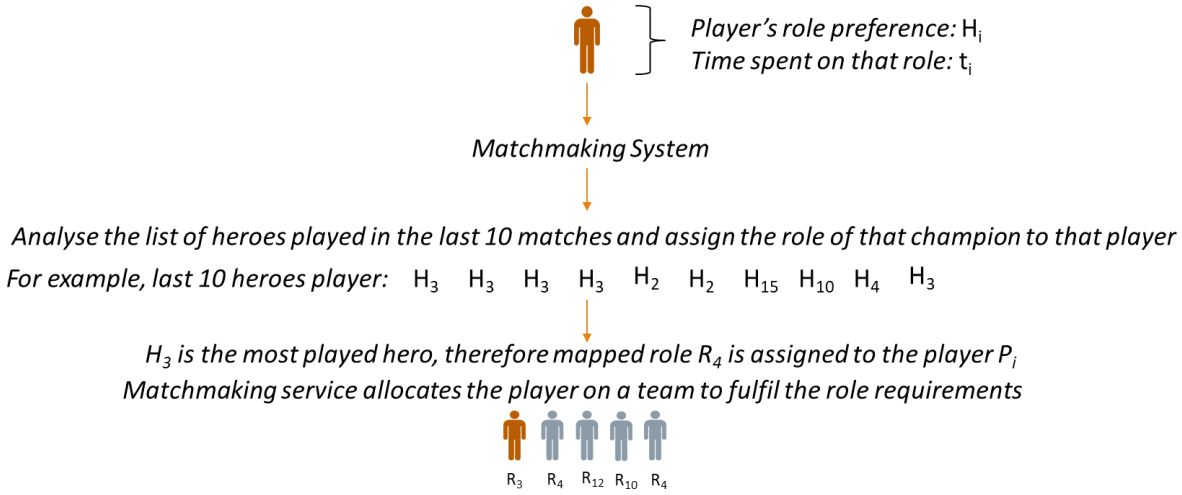

Figure 3 Role-based matchmaking process of MEDIROMA

Chapter 4: Results

4.1 Features and Cluster Extraction

From the initial list of 97 features, the top 50 features are extracted by using Univariate Feature Selection. These 50 features are ranked by a score that determines how much impact it has on the singular feature of match outcome. Furthermore, these features reveal META information about the key factors that allows teams to win. For example, the feature with the highest score is 'teamtowerkills', which is the number of enemy towers destroyed by a team. This is the most important feature of the game, since destroying enemy towers in League of Legends weakens the belligerents and awards the team with boosts and rewards in the form of boosted minions, in game credits, and bonus damage. Although this information is not utilised by the matchmaking process, they can be valuable for any players looking to improve their playstyle and increase their chance of winning.

In the following table (Table 2) the top 50 features are listed. Appendix A contains the data dictionary on the definitions of these features.

Table 2 Top 50 features selected by Univariate Feature Selection

Features	Score
teamtowerkills	43396.7
opptowerkills	43396.7
teamkills	9194.546
teamdeaths	9167.249
teamdragkills	6367.049
oppdragkills	6367.049
fbaron	5620.69
teambaronkills	4321.919
oppbaronkills	4321.919
gspd	3775.746
firsttothreetowers	2925.379
firstmidouter	1779.879
a	1332.917
d	1214.244
gdat15	1174.776
k	1105.744
ft	1098.262
gdat10	612.5116
monsterkillsenemyjungle	600.2472
elementals	524.8811
oppelementals	524.8811
xpdat10	461.6448

airdrakes	292.193
herald	277.4979
kpm	242.3003
okpm	242.2983
csdat10	206.7166
firedrakes	202.2845
waterdrakes	197.2118
earnedgpm	146.3859
earthdrakes	143.1604
fbassist	98.89533
side	95.43443
dmgtochampsperminute	86.50107
dmgtochamps	66.68718
totalgold	57.68362
monsterkills	53.57122
elders	40.07142
oppelders	40.07142
wcpm	26.28438
wardkills	21.28475
fd	20.56287
ban4	19.22078
ban5	18.13891
goldspent	17.05847
doubles	16.22088
team	11.9018
fb	8.423843
fbvictim	8.423843

After the completion of feature space exploration, the Mean Shift Clustering discovered four clusters from the dataset. Each of these cluster is then simultaneously passed through a predictor for calculating the win rate based on their outcome for that game. To verify the stability of the output, the dataset has been mined several times with varying bandwidth size for consistency. In most cases, the mining routine produced four clusters with minor changes in composition.

Figure 4 shows the scatter plot for the clusters, with the clusters being boldened and highlighted.

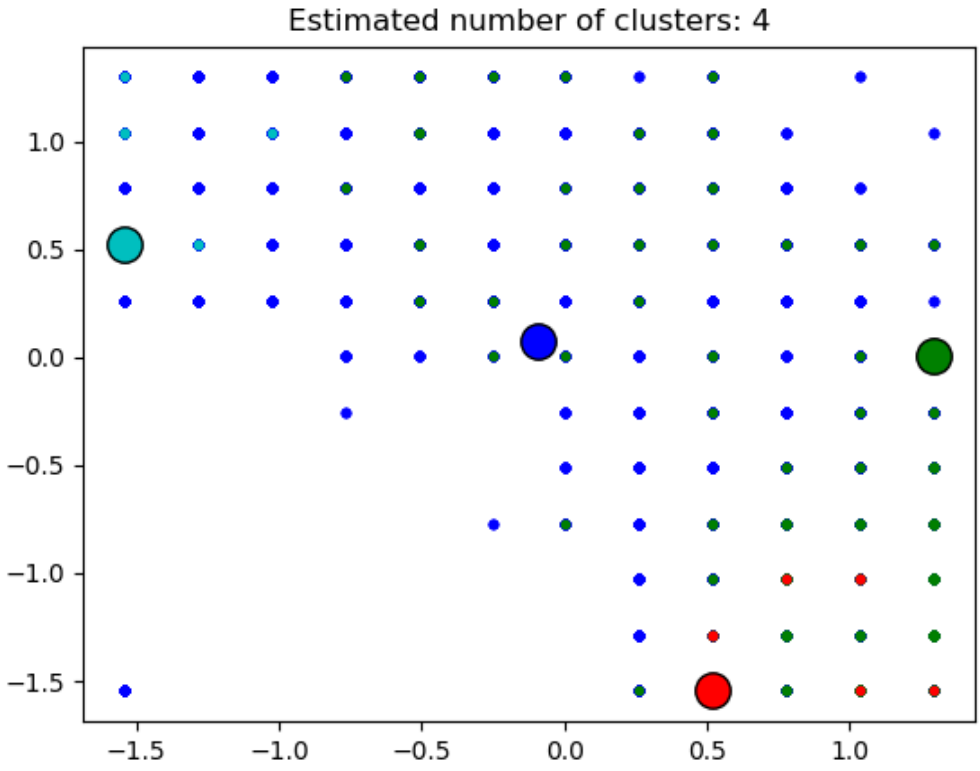

Figure 4 Cluster diagram from the dataset

4.2 META Cluster Identification

The clusters produced by feature space exploration routine are passed to a predictor that calculates the win rate based on each cluster's win loss ratio. Each cluster produced different results in terms of their win rate when passed through a predictor. Table 3 shows the win rate for each of the clusters.

Table 3 Cluster groups win rate

Cluster Group	Win rate (%)
1	49.39
2	87.65
3	100
4	0

Cluster groups 3 and 4 are outliers in this outcome, because no composition of champions exist that can always guarantee a win or a loss. Therefore, the only valid clusters from this outcome are cluster groups 1 and 2. Since cluster group 2 (87.65%) has a higher win rate than cluster group 1 (49.39%), we deduce that cluster group 2 is the META cluster that represents the META of the game. Therefore, cluster group 2 is further explored to investigate its champion composition.

4.3 Evaluation

The validation and verification of the output has been conducted by means of extracting the roles and champions from the META cluster that was identified in the previous section. The goal of the evaluation is to find out whether the composition of the champions recommended by the META cluster yields better experience and more satisfaction for the players. Since experience and satisfaction is subjective with individuals, the evaluation is limited in its scope to determine the true scale of applying this META composition on the video game on a large scale. However, even without running a live simulation on an existing build of the game using the extracted composition, the viability of running such a matchmaking solution can still be determined by creating an environment where players are tasked to play only those certain champions to simulate the experience of playing in a solo queue environment. The limitation of this evaluation approach is discussed further in Chapter 5.

The cluster group 2 revealed 36 champions that spans across all the different standard roles of League of Legends such as Top, Mid, Jungle, AD Carry and Support. According to the predictor, picking any five of these champions should result in a 87.65% chance of winning for that respective team. The following is the list of 36 champions: Jax, Vayne, Vladimir, Skarner, Taliyah, Ryze, Kindred, Trundle, Gragas, Urgot, Kha'zix, Hecarim, Olaf, Nocturne, Jarvan IV, Ezreal, Kalista, Kayle, Cassiopeia, Zac, Aatrox, Sivir, Sejuani, Tristana, Rek'Sai, Kai'Sa, Evelynn, Vi, Karthus, Elise, Rakan, Xin Zhao, Lee Sin, Kayn, Camille and Nidalee. Based on the information extracted from this cluster, it can be postulated that these champions dictate the META of League of Legends, which is represented by the top 1% player population of the game. However, the composition of champions may be different for lower skilled player base, since the META tends to shift between skill levels of the players. This has been discussed in detail in Chapter 5.

Evaluation of this composition was conducted with the help of existing players whose insight was sourced from public online discussion boards where they discussed the role of teams based on their own experience and knowledge of the game. The consensus from the posts on this discussion board was agreed upon that the composition consisted mainly of the standard META composition known in the community, along with a few surprises that they considered not to be detrimental to the outcome of a match. However, since the meta discovered is only applicable to the highest level of play, their own skill level might have obscured them from experiencing a subset of this META composition. Some of these players attempted to mix match these compositions to various degree and reported a moderate to high level of success in their games in terms of the outcome and their own satisfaction. Based on this evaluation, it can be postulated that the player engagement can be improved if the matchmaking solution of League of Legends takes into account this meta cluster.

Chapter 5: Discussion

This chapter examines the possibilities and drawbacks of the proposed solution MEDIROMA. Since this research had a short time frame for completion, several avenues of investigation have been left unchecked. The time frame has also limited the ability to evaluate the model in a live environment, which could have yielded more quantifiable results for the experiment. The possibilities for the research in META discovery in other domain of competitive applications, along with other genres of video games, are also laid out in this chapter for discussion. This serves the purpose of examining the potential for future improvements and expansions that can be applied to produce better results, as well as accommodating the solution for other applications. Furthermore, the scope of this research has been discussed, which explains the factors that needs to be considered for implementing a complete and robust matchmaking solution that solve majority of the problems of a matchmaking service and competitive multiplayer games.

The major drawback of this research has been a definitive approach to evaluate the model of MEDIROMA through a practical and live experiment. Since a live version of this matchmaking solution could not be tested against the current matchmaking solution offered by League of Legends against a specific control group of players, it is difficult to estimate the success of this solution. However, due to the lack of any support or implementation of a role-based matchmaking in League of Legends, it can be hypothesized that player experience will improve if the developers took the approach of using a META discovery solution to allocate the players based on their preferred role. Furthermore, it can be postulated that solo queue players will have a better chance of winning against a full 5 stack team with this solution implemented, since the solo queue team is now less likely to get into a conflict over choosing their roles and champions.

It must be noted that this research has attempted to solve one of many other problems associated with matchmaking solutions in multiplayer video games. Implementing a role-based matchmaking alone will not solve many of the issues that affects the experience of competitive players. For example, this research does not incorporate any of the elements of skill rating or network latency into its solution. Therefore, this research cannot be used by itself for any multiplayer games and must be incorporated with existing matchmaking services that provides both a skill rating and latency management component. However, integrating these components alone will not be enough to address the issues of matchmaking if the feedback from the player community is not taken into account regarding the latent features of the game, such as the dynamics of the player expectation Vs actual experience, frustration score, champion win rate, and many more factors. Depending on the requirements of the game, these factors may be adjusted within the matchmaking service to allow for a more robust solution that can satisfy the players.

There is room for improving the model of MEDIROMA in terms of meta discovery at different skill levels and different build releases. Due to limited access to match statistics, the meta discovery process is limited only at the highest level of gameplay, which consists of pro league matches played by professional players. Although the META discovered has been applied to the highest skill level, it does not reflect the META for the lower skilled players of the game. One approach could be to identify the META at different skill levels such as Bronze, Silver, Gold and Platinum, and allocate the players by combining skill rating and their preference. However, this would increase the queue time for matchmaking and may result in frustration for players. Another approach is to investigate the meta for all skill levels and assuming only one general meta exists for the game. However, this would be detrimental for the players, since the meta may vary at different skill levels and create unexpected outcomes for the players.

The means of META discovery has limitations since MEDIROMA relies solely on the Mean Shift clustering for feature space exploration. As a non-parametric clustering algorithm, Mean Shift clustering is a valuable tool to identify clusters within a given dataset without the need of passing the expected number of clustering groups. However, it has drawbacks when clustering datasets that have multiple modes within the continuous dataset. To overcome this limitation, Ren et. al combined the Parallel Spatial Boosting Machine Learner and DBSCAN along with Mean Shift Clustering algorithm to allow it to avoid classifying multiple modes within a dense cluster region, and thus proposed Boosted Mean Shift Clustering in [32]. Therefore, by using the solution for boosted Mean Shift clustering, the META discovery process may improve and avoid situation where it produces cluster groups that are outliers in the feature space.

This research work needs to be evaluated in other video games, along with other applications which has a competitive environment such as sports or military tactics. This is will help identify issues with the methodology of the proposed solution, and whether changes need to be made to accommodate different applications. For example, a META discovery for the best composition of a team of Football team would require different sets of feature selection compared to that of a META discovery for the composition of a heavy armoured division within a military unit. It would be suitable to have a model that can be applied over any applications without much adjustment, but this would require further research that can iterate through various permutations of the existing model, along with minor adjustments. Thus, an investigation needs to be conducted to examine if a general purpose and portable model can be established that can be applied over any competitive applications.

Chapter 6: Conclusion

Although most of the research papers have attempted to solve only one of the singular problems within the domain of matchmaking, it is important to note that most video games would necessitate the requirement of having multiple problems to be addressed. However, most of the papers presented in the literature review were too focused on skill ratings of the players, and very few of them considered looking at the solving issues which persists in many competitive video games. Since video games are a complex interactive media, developers need to look beyond the scope of simple match outcomes to improve the player experience. The lack of attention on improving player experience has motivated this research into focusing on the problem of player conflicts that can arise from overlapping role preference. However, as with all research papers, this research has a very limited scope when trying to solve the problems associated with matchmaking solutions.

Given the limitations, the model of MEDIROMA has room for improvements in terms of its implementation and evaluation. Investigation into integrating the META features discovered at the early phase with the matchmaking procedure could result in taking advantage of the latent features of the game, that would improve the player experience of the game. Along with implementations of a better META discovery solution, the matchmaking solution needs to be integrated with a skill rating module and a network latency module. This will help in making MEDIROMA a fully-fledged matchmaking solution that can be used for most modern multiplayer video games. Aside from video games, MEDIROMA can also be applied to other applications where competition is a vital component such as sports and military. Discovering and understanding the most effective strategies for any competitive assessment requires a significantly large dataset with an ample number of features that describes the attributes of the feature sets. Therefore, for any applications that attempts to investigate its META, investigators must ensure that they have collected sufficient data regarding the competitive environment. Given the successful evaluation based on the positive feedback from the players, it can be postulated that other competitive assignments can be examined for its META and a winnable composition can be formed.

Bibliography

1 Brand, J.E., Jervis, J., Huggins, P.M., and Wilson, T.W.: 'Digital Australia 2020', in Editor (Ed.)^(Eds.): 'Book Digital Australia 2020' (Faculty of Society & Design, Bond University, 2019, edn.), pp.

2 Herbrich, R., Minka, T., and Graepel, T.: 'TrueSkill ™: A Bayesian skill rating system', in Editor (Ed.)^(Eds.): 'Book TrueSkill ™: A Bayesian skill rating system' (2007, edn.), pp. 569-576

3 Dangauthier, P., Herbrich, R., Minka, T.: 'TrueSkill Through Time: Revisiting the History of Chess', In Advances in neural information processing systems 2008, pp. 337-344

4 Guo, S., Sanner, S., Graepel, T., and Buntine, W.: 'Score-based Bayesian skill learning', in Editor (Ed.)^(Eds.): 'Book Score-based Bayesian skill learning' (2012, edn.), pp. 106-121

5 Birlutiu, A., and Heskes, T.: 'Expectation Propagation for Rating Players in Sports Competitions', 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), 2007, (PKDD), pp. 374-381

6 S. Nikolenko, A.S.: 'A new Bayesian rating system for team competitions', Proceedings of the 28th International Conference on Machine Learning (ICML-II), 2011, pp. 601-608

7 Chen, Z., Aghdaie, N., Xue, S., Zaman, K.A., El-Nasr, M.S., Kolen, J., and Sun, Y.: 'EOMM: An engagement optimized matchmaking framework', in Editor (Ed.)^(Eds.): 'Book EOMM: An engagement optimized matchmaking framework' (2017, edn.), pp. 1143-1150

8 Stroh-Maraun, N., Kaimann, D., and Cox, J.: 'More than skills: A novel matching proposal for multiplayer video games', Entertainment Computing, 2018, 25, pp. 26-36

9 Delalleau, O., Contal, E., Thibodeau-Laufer, E., Ferrari, R.C., Bengio, Y., and Zhang, F.: 'Beyond Skill Rating: Advanced Matchmaking in Ghost Recon Online', IEEE Transactions on Computational Intelligence and AI in Games, 2012, 4, (3), pp. 167-177

10 Lei, Z., Jun, W., Zhong-Cun, W., and Chong-Jun, W.: 'A Factor-Based Model for Context-Sensitive Skill Rating Systems', in Editor (Ed.)^(Eds.): 'Book A Factor-Based Model for Context-Sensitive Skill Rating Systems' (2010, edn.), pp. 249-255

11 Huang, T.K., Lin, C.J., and Weng, R.C.: 'Ranking individuals by group comparisons', Journal of Machine Learning Research, 2008, 9, pp. 2187-2216

12 DeLong, C., Pathak, N., Erickson, K., Perrino, E., Shim, K., and Srivastava, J.: 'TeamSkill: Modeling team chemistry in online multi-player games', in Editor (Ed.)^(Eds.): 'Book TeamSkill: Modeling team chemistry in online multi-player games' (2011, edn.), pp. 519-531

13 Delong, C., and Srivastava, J.: 'TeamSkill Evolved: Mixed Classification Schemes for Team-Based Multi-player Games' (Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. 2012)

14 Delong, C., Terveen, L., and Srivastava, J.: 'TeamSkill and the NBA: applying lessons from virtual worlds to the real-world', in Editor (Ed.)^(Eds.): 'Book TeamSkill and the NBA: applying lessons from virtual worlds to the real-world' (2013, edn.), pp. 156-161

15 Menke, J.E., Reese, C. S., & Martinez, T. R. : ‘Hierarchical Models for Estimating Individual Ratings from Group Competitions’, American Statistical Association, 2007
16 Sarkar, A., Williams, M., Deterding, S., and Cooper, S.: ‘Engagement effects of player rating system-based matchmaking for level ordering in human computation games’, in Editor (Ed.)^(Eds.): ‘Book Engagement effects of player rating system-based matchmaking for level ordering in human computation games’ (2017, edn.), pp. 1-10
17 Cooper, S., Deterding, C.S., and Tsapakos, T.: ‘Player rating systems for balancing human computation games: testing the effect of bipartiteness’, in Editor (Ed.)^(Eds.): ‘Book Player rating systems for balancing human computation games: testing the effect of bipartiteness’ (DIGRA Digital Games and Research Association, 2016, edn.), pp.
18 Baldwin, A., Johnson, D., Wyeth, P., and Sweetser, P.: ‘A framework of dynamic difficulty adjustment in competitive multiplayer video games’, in Editor (Ed.)^(Eds.): ‘Book A framework of dynamic difficulty adjustment in competitive multiplayer video games’ (IEEE, 2013, edn.), pp. 16-19
19 Myślak, M., and Deja, D.: ‘Developing Game-Structure Sensitive Matchmaking System for Massive-Multiplayer Online Games’, in Editor (Ed.)^(Eds.): ‘Book Developing Game-Structure Sensitive Matchmaking System for Massive-Multiplayer Online Games’ (2015, edn.), pp. 200-208
20 Suznjevic, M., Matijasevic, M., and Konfic, J.: ‘Application context based algorithm for player skill evaluation in MOBA games’, in Editor (Ed.)^(Eds.): ‘Book Application context based algorithm for player skill evaluation in MOBA games’ (2015, edn.), pp. 1-6
21 Yitong, G., Yunhua, D., and Xueyan, T.: ‘On matchmaking for multiplayer cloud gaming’, in Editor (Ed.)^(Eds.): ‘Book On matchmaking for multiplayer cloud gaming’ (2017, edn.), pp. 1-3
22 Agarwal, S., and Lorch, J.: ‘Matchmaking for online games and other latency-sensitive P2P systems’, in Editor (Ed.)^(Eds.): ‘Book Matchmaking for online games and other latency-sensitive P2P systems’ (2009, edn.), pp. 315-326
23 Zander, S., Leeder, I., and Armitage, G.: ‘Achieving fairness in multiplayer network games through automated latency balancing’, in Editor (Ed.)^(Eds.): ‘Book Achieving fairness in multiplayer network games through automated latency balancing’ (2005, edn.), pp. 117-124
24 Manweiler, J., Agarwal, S., Zhang, M., Roy Choudhury, R., & Bahl, P: ‘Switchboard A matchmaking system for multiplayer mobile games’, Proceedings of the 9th international conference on Mobile systems, applications, and services 2011, pp. 71-84
25 Boroń, M., Brzeziński, J., and Kobusińska, A.: ‘P2P matchmaking solution for online games’, Peer-To-Peer Networking and Applications, 2019, pp. 1-14
26 Claypool, M., Decelle, J., Hall, G., and O'Donnell, L.: ‘Surrender at 20? Matchmaking in league of legends’, in Editor (Ed.)^(Eds.): ‘Book Surrender at 20? Matchmaking in league of legends’ (2015, edn.), pp. 1-4
27 Véron, M., #233, ron, Marin, O., #233, and Monnet, b.: ‘Matchmaking in multi-player on-line games: studying user traces to improve the user experience’. Proc.

Proceedings of Network and Operating System Support on Digital Audio and Video Workshop, Singapore, Singapore2014 pp. Pages
28 Shafran, V., Kaminka, G., Kraus, S., and Goldman, C.V.: 'Towards bidirectional distributed matchmaking', in Editor (Ed.)^(Eds.): 'Book Towards bidirectional distributed matchmaking' (2008, edn.), pp. 1405-1408
29 Ogston, E., and Vassiliadis, S.: 'Local distributed agent matchmaking', in Editor (Ed.)^(Eds.): 'Book Local distributed agent matchmaking' (2001, edn.), pp. 67-79
30 Sycara, K., Widoff, S., Klusch, M., and Lu, J.: 'Larks: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace', Autonomous Agents and Multi-Agent Systems, 2002, 5, (2), pp. 173-203
31 Comaniciu, D., and Meer, P.: 'Mean shift: a robust approach toward feature space analysis', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24, (5), pp. 603-619
32 Ren, Y., Kamath, U., Domeniconi, C., and Zhang, G.: 'Boosted mean shift clustering', in Editor (Ed.)^(Eds.): 'Book Boosted mean shift clustering' (2014, edn.), pp. 646-661

Appendix A

Below is the data dictionary for the features that impacts the outcome of the game in League of Legends.

Variable	Description
side	Map side.
champion	Champion name.
k	Total kills.
d	Total deaths.
a	Total assists.
teamkills	Total kills by team.
teamdeaths	Total deaths by team.
fb	First blood kill (1 yes, 0 no).
fbassist	First blood assist (1 yes, 0 no).
fbvictim	First blood victim (1 yes, 0 no).
kpm	Kills per minute (individuals and teams reported separately).
okpm	Opponent kills per minute (for players, reflects opponent in same position).
fd	First dragon of game killed (1 yes, 0 no).
teamdragkills	Total dragons killed by team.
oppdragkills	Total dragons killed by opposing team.
elementals	Total elemental drakes killed by team.
oppelementals	Total elemental drakes killed by opposing team.
firedrakes	Total infernal drakes killed by team.
waterdrakes	Total ocean drakes killed by team.
earthdrakes	Total mountain drakes killed by team.
airdrakes	Total cloud drakes killed by team.

elders	Total elder dragons killed by team.
oppelders	Total elder dragons killed by opposing team.
herald	Rift herald taken (1 yes, 0 opponent took it, blank herald not killed).
ft	First tower of game killed (1 yes, 0 no).
fttime	First tower kill time, in minutes. (Seconds measured in hundredths of a minute.)
firstmidouter	First team to kill mid lane outer tower (1 yes, 0 no).
firsttothreetowers	First team to kill three towers (1 yes, 0 no).
teamtowerkills	Total towers killed by team.
opptowerkills	Total towers killed by opposing team.
fbaron	First baron of game killed (1 yes, 0 no).
fbarontime	First baron time, in minutes. (Seconds measured in hundredths of a minute.)
teambaronkills	Total barons killed by team.
oppbaronkills	Total barons killed by opposing team.
dmgtochamps	Total damage dealt to champions.
dmgtochampsperminute	Total damage dealt to champions per minute.
earnedgoldshare	Share of team's total gold, with starting gold and inherent gold generation removed.
wardkills	Total wards cleared/killed (of all types).
wcpm	Total wards cleared/killed per minute (of all types).
totalgold	Total gold earned from all sources.
earnedgpm	Earned gold per minute, with starting gold and inherent gold generation removed.
goldspent	Total gold spent.
gspd	Gold spent percentage difference. For more information, see http://oracleselixir.com/2015/07/measuring-

	the-margins-gold-spent-percentage-difference/.
monsterkills	Neutral monsters killed.
monsterkillsownjungle	Neutral monsters killed in own team's jungle.
monsterkillsenemyjungle	Neutral monsters killed in opposing team's jungle.
csdat10	Creep score difference at 10:00.
gdat10	Gold difference at 10:00.
gdat15	Gold difference at 15:00.
xpdat10	Experience difference at 10:00.