Aalto University
School of Science

Master's Programme in Mathematics and Operations Research

Win probability estimation for strategic decision-making in esports

Perttu Jalovaara

Master's Thesis
2024

Aalto University
School of Science

Author Perttu Jalovaara

Title Win probability estimation for strategic decision-making in esports

Degree programme Mathematics and Operations Research

Major Systems and Operations Research

Supervisor Prof. Ahti Salo

Advisor D.Sc. (Tech.) Oliver Struckmeier

Date 26 August 2024 **Number of pages** 38 **Language** English

Abstract

Esports, i.e., the competitive practice of video games, has grown significantly during the past decade, giving rise to esports analytics, a subfield of sports analytics. Due to the digital nature of esports, esports analytics benefits from easier data collection compared to its physical predecessor. However, strategy optimization, one of the focal points of sports analytics, remains relatively unexplored in esports. In traditional sports analytics, win probability estimation has been used for decades to evaluate players and support strategic decision-making.

This thesis explores the use of win probability estimation in esports, focusing specifically on League of Legends (LoL), one of the most popular esports games in the world. The objective of this thesis is to formalize win probability added, i.e., the change in win probability associated with a certain action, as a contextualized measure of value for strategic decision-making, using mathematical notation appropriate for contemporary esports. The proposed method is elaborated by applying it to the evaluation of items, a strategic problem in LoL. To this end, we train a deep neural network to estimate the win probability at any given LoL game state. This in-game win probability model is then benchmarked against similar models.

Keywords Esports analytics, League of Legends, win probability estimation, match outcome prediction, in-game win probability, win probability added

Preface

This thesis is the culmination of an adolescent obsession with optimizing video game strategies. I want to thank Professor Ahti Salo and my advisor, Oliver Struckmeier, for enabling me to work on a topic I am incredibly passionate about. I also want to thank my family and friends for their endless support and the joy they bring into my life. Lastly, I want to thank my partner for her unyielding patience and love.

Helsinki, 26 August 2024

Perttu Jalovaara

Contents

Symbols and abbreviations

Symbols

$\mathcal{A}$	action space
$\mathcal{X}$	state space
D	data set $D = \{d^{(1)}, \ldots, d^{(N)}\}$
$d^{(i)}$	i-th data point $d^{(i)} = (a^{(i)}, x^{(i)}, z^{(i)})$ in D
$a^{(i)}$	i-th action in D
$x^{(i)}$	i-th initial state in D
$z^{(i)}$	i-th final state in D
y	match outcome (1 = win, 0 = loss)
w	estimated win probability
W	initial win probability
ΔW	win probability added

Abbreviations

AGV	added goal value
AI	artificial intelligence
API	application programming interface
DNN	deep neural network
ECE	expected calibration error
ETM	end-of-game tactics metric
LoL	League of Legends
MOBA	multiplayer online battle arena
RNN	recurrent neural network

1 Introduction

Esports, i.e., the competitive practice of video games, has grown significantly in popularity during the past decade [52]. The growing body of research on esports is highly interdisciplinary [18]. *Esports analytics* is defined as the use of esports-related data to assist with decision-making processes arising both in-game and outside of it [42]. Its physical predecessor, sports analytics, has produced insights that extend beyond the domain of sports [3, 58]. Due to their digital nature, esports benefit from easier data collection, a core challenge of sports analytics in the big-data era [30]. Esports analytics has clear potential, but the field is still in its infancy [18, 42].

By definition, the optimal strategy in sports, esports, or any game where two teams compete against each other, is to maximize the probability of your team winning [35]. Although difficult to quantify, this *win probability* is implicitly estimated by players, coaches, and fans during a match [26]. Computational win probability estimation is one of the cornerstones of modern sports analytics, with research beginning in the 1960s [24, 36]. Win probability estimates are now regularly used in sports to guide decision-making [26, 28, 40]. In esports, win probability models have recently started to gain traction through their use in broadcasts of major events such as the 2023 League of Legends (LoL) World Championship [37]. Despite their prevalence [17, 27], esports win probability models have not yet been widely utilized in strategy optimization, a focal point of sports analytics [1].

In this thesis, we study a recurring decision-making problem in esports: *Given a game state and a finite set of actions, which action should a player take to maximize their team's win probability?* This allows us to consider both high-level decisions (e.g., in Counter-Strike: *Should we take site A or B this round?*) and low-level decisions (e.g., in LoL: *Which item should I buy in this situation?*). In both examples, the set of decision alternatives (actions) is finite; there are two sites in the former and roughly two hundred items in the latter. The problem definition excludes continuous gameplay decisions such as movement in real-time games or aiming in shooters.

As any LoL player would tell you, the choice of items cannot be made without game-specific details and is thus context-dependent. This is often the case with strategic decisions in esports, which makes it interesting yet difficult to evaluate and compare the decision alternatives. Moreover, some actions are only taken by players to secure wins when they are already ahead or as a last resort in desperate situations. Such biases cloud esports data, which makes simple aggregate statistics such as *win rate* unreliable for comparing actions. Thus, a debiased, i.e., contextualized [53], method of evaluating actions is required.

This thesis explores the use of win probability estimation in esports and proposes the *mean win probability added* as a contextualized metric for evaluating actions. Win probability added, i.e., the change in win probability associated with a certain action or player, is a commonly used metric in sports analytics for evaluating players and decisions [28, 40, 53]. The objective of this thesis is to formalize the use of win probability estimation to support esports decision-making in a general, game-agnostic manner. Additionally, this thesis aims to elaborate the proposed method by applying it to the evaluation of items, a strategic problem in LoL, one of the most popular esports

games in the world [56]. To this end, we train a deep neural network to estimate the win probability at any given LoL game state. This *in-game win probability model* is then benchmarked against similar existing models.

This thesis is structured as follows. Chapter 2 provides an overview of the core concepts and terminology of LoL. Chapter 3 establishes a background for the thesis by reviewing the relevant literature on sports and esports analytics. Chapter 4 describes the method of evaluating decision alternatives using win probability estimates. In Chapter 5, this method is applied to real esports data through a LoL case study. Finally, Chapter 6 concludes the thesis with a summary of the thesis and an outline of future research avenues.

2 League of Legends

League of Legends (LoL) is a competitive computer game developed by Riot Games. This chapter describes the game to the extent necessary to understand the rest of this thesis. All elements related to LoL in the thesis (e.g., characters, items, graphics) are the exclusive property of, and provided courtesy of, Riot Games. Because LoL is updated periodically, every two to three weeks, some of the details will eventually become outdated. Nevertheless, the core game concepts provided here remain relatively reliable. Current and precise game information can be found on the LoL Wiki [23]. At the time of writing, the most recent game update is Patch 14.15.

LoL can be played on different *maps*, i.e., virtual battlegrounds, such as *Summoner's Rift* and *The Howling Abyss*. Furthermore, LoL includes multiple game modes, each with its own set of rules. This thesis, however, focuses solely on the standard competitive map and mode combination, Classic Summoner's Rift 5v5. Throughout the rest of this thesis, the term LoL refers specifically to this game format.

Figure 1: A simplified visualization (minimap) of Summoner's Rift, the standard League of Legends map. The gray areas in the bottom-left and top-right corners are the Blue Base and Red Base, respectively.

In LoL, two teams of five players compete against each other with the objective of destroying the opposing team's central structure, the *Nexus*. The teams are assigned—either by a matchmaking system or according to tournament rules—a side of the map to play on. These sides are represented by the colors blue and red. The teams are thus referred to as Blue Team and Red Team. As seen in Figure 1, each team has its own Base, which contains the team's Nexus. Blue Team wins by destroying the Red Nexus, and vice versa. Players can perform a multitude of actions in LoL, but the outcome of a match is ultimately determined by the destruction of a Nexus.

In order to reach the enemy Nexus, the Blue Team must first destroy the *turrets* protecting the Red Base. Naturally, the Red Team tries to prevent this from happening,

while simultaneously attacking the turrets and base of the Blue Team. In standard play, each player serves a specific role in a team. The five roles are called *top*, *jungle*, *mid*, *bot*, and *support*. These names stem from the allocation of players to specific parts of Summoner's Rift (see Figure 1). There are three *lanes* (top, middle, bottom) that lead from one base to the other. The area between the lanes is the jungle. The support usually begins the game in the bottom lane with their team's bot but eventually transitions into roaming around the map. This standard lane allocation ensures that all turrets are protected from the beginning of a match.

Before the game begins, each player must select a virtual character, known as a *champion*, to play. This pre-game process is called *champion selection*, or simply *draft*, and it sets the groundwork for the match. With 10 players and 168 unique champions to choose from, the number of possible draft permutations is $168!/158! \approx 1.3 \cdot 10^{22}$. In practice, some champions are picked more frequently than others, resulting in a distribution of drafts that is concentrated on a relatively small subset of these permutations. Nonetheless, the draft produces inherent pre-game variation, making every LoL match unique.

In the game, each player controls their champion—typically using a keyboard and a mouse—until a Nexus is destroyed or a team forfeits the match. The player is synonymous with their champion; champions cannot be drafted twice in the same match. In addition to the five human-controlled champions, each team has an army of *minions*, which are comparatively weak, computer-controlled units that automatically attack any opponents they encounter.

Figure 2: A screenshot from LoL demonstrating champion movement. The blue circle on the left indicates a movement command for the champion in play, Shen.

The most elementary control mechanism in LoL is a right-click of the mouse, which causes the champion to move or perform a *basic attack*, depending on the location of the cursor. This fundamental pattern of LoL is illustrated in Figure 2. In

addition to moving and attacking, players *cast* their champion's *abilities* to interact with their opponents, their teammates, and other components of the game. Most champions have five abilities in total: one passive ability, three basic abilities (typically assigned to the Q, W, and E keys), and one ultimate ability (typically assigned to R). Unlike basic attacks and movement commands, which can be made virtually continuously, most abilities have a *cooldown*, a post-cast timer that limits the use of the ability. In addition to unique abilities and a distinct appearance, every champion has characteristic *statistics*, which dictate, e.g., how fast they move and how powerful their attacks and abilities are.

The standard lane allocation combined with opposing motives inevitably leads to *combat* between the teams. Since LoL is a real-time game, player combat is fast-paced and requires precise keyboard and mouse control. At the core of combat are two gameplay elements: *health* and *damage*. Basic attacks and most abilities deal damage, causing their targets to lose health. Once a unit's (e.g. champion, minion, turret) health reaches zero, the unit *dies*. The champion dealing the final instance of damage to a dying unit is rewarded a *kill*. Any other champions participating in the kill are rewarded an *assist*. Dead champions are temporarily suspended from the game; they have a *death timer* ranging from 10 to 60 seconds, increasing in duration as the game proceeds. During this timer, the dead champion cannot affect the game, creating a window of opportunity for their opponents to destroy turrets or even the Nexus.

To prevail in combat, players strive to increase their relative strength and gain an advantage over their opponents. The term *strength* does not refer to any particular attribute in LoL; rather, it encompasses the overall power of a champion. A player can increase the strength of their champion through two types of fundamental resources: *experience* and *gold*, which are gained by, e.g., killing minions, participating in champion combat, and destroying turrets. The benefits of these resources are indirect; they translate into increases in champion strength only when certain thresholds are met. Collecting enough experience will cause a champion to *level up*. Every level amplifies a champion's statistics and abilities. Collecting enough gold allows a player to purchase *items* that increase specific statistics and grant unique *effects*. Some items provide an *active* effect, an additional ability for the player to use. Deciding which item to purchase is difficult because the provided effects are often impossible to compare directly.

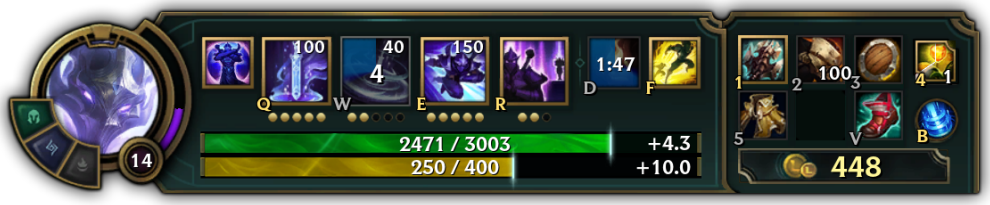

Figure 3: A screenshot of the LoL HUD. Information appearing from left to right and top-down: champion, level, experience, abilities, summoner spells, current and maximum health, current and maximum energy (usually mana), items, and gold.

Figure 3 presents a screenshot of the in-game heads-up display (HUD), which contains critical information about the state of the player as well as the actions available to them. The following explanation provides an example of LoL terminology in action. In the case of Figure 3, the played champion is Shen, he is level 14 and has roughly a third of the required experience for the next level-up. Most of Shen's abilities are available (Passive, Q, E, R), except for Spirit's Refuge (W), which is on cooldown for 4 seconds. Shen has 2471 health out of his maximum of 3003. He has 250 energy out of his maximum of 400. Shen's summoner spells are Ignite (D), which is on cooldown, and Flash (F), which is available. Shen has five items and a Stealth Ward (4) in his inventory. One of the items, Titanic Hydra (1), grants an active effect that is available. The item Deadman's Plate (2) is granting additional effects with 100 stacks. Shen can also cast Recall (B) and has 448 unspent gold.

In addition to kills and deaths, numerous other events can occur in a LoL match. The most impactful events and their consequences are listed in Table 1.

Table 1: Overview of the most impactful events in LoL.

Event	**Consequence**
Kill (champion)	Killer receives gold and experience.
Assist	Assistant receives a share of kill gold and experience.
Death	Dead champion is temporarily suspended from the game.
Voidgrub (kill)	Killing team deals more damage to turrets.
Rift Herald	Killer can summon the Herald to destroy a turret.
Dragon	Killing team becomes slightly stronger.
Baron Nashor	Killing team becomes stronger for 3 minutes.
Elder Dragon	Killing team becomes extremely strong for 3 minutes.
Turret (destruction)	Destroyer receives gold and the lane opens up.
Inhibitor	Destroying team gets more minions for 5 minutes.
Nexus	Destroying team wins the match.

Like most modern esports, LoL has a steep learning curve. For a novice player, the first step is to learn the abilities and nuances of one champion. However, playing the game at any decent level requires internalizing the abilities of all 168 champions, the numerous champion-specific interactions, and the essential game mechanics. Unlike perfect information games [59] such as chess, the game state in LoL is partially observable [4]; a team can only see the parts of Summoner's Rift occupied by their units, and the rest is obscured by the *fog of war*. Moreover, a player's in-game view, the *camera*, can only focus on a small, specific area of the map at any given time. The amount of observable information is thus limited by the player's ability to move the camera while performing hundreds of gameplay actions every minute.

While not comprehensive, the above overview covers: 1) the essential concepts and terminology of LoL, and 2) the complexity and motivation behind analyzing decision-making in esports. The next chapter reviews past analytical work, covering studies specific to LoL, as well as broader research on esports and traditional sports.

3 Background

In order to understand and appreciate the potential of win probability estimation in esports, one must acquaint oneself with the existing applications of win probability estimation in traditional sports. Therefore, this chapter begins with an overview of the relevant sports analytics literature. Then, we look at recent developments in esports analytics, focusing on win probability estimation, specifically for League of Legends. Accurate win probability estimation requires the ability to draw conclusions from the state of a game. This ability is also crucial in developing artificial agents that can play games. While not the subject of this thesis, the research on such game-playing systems provides valuable findings that can be applied to win probability estimation. Thus, this chapter concludes with a discussion on the parallels between game-playing artificial intelligence and esports analytics.

3.1 Win probability estimation in traditional sports

Sports analytics is a multidisciplinary field comprising the collection, analysis, and interpretation of sports-related data [1, 29]. The field is also commonly referred to as statistics in sports [44]. The primary objective of sports analytics is to gain insights that provide a competitive advantage on the field of play [1]. On-field sports analytics include, e.g., the assessment of player performance [41] and the comparison of coaching tactics [28]. Sports analytics also serves to inform off-field decision-making in various business-oriented aspects of the sports industry, ranging from optimal ticket pricing of sports matches to economic assessment of large-scale sports events like the Olympic Games [30].

The impact of sports analytics can be seen in phenomena such as the *three-point revolution* in basketball, where analysis of shot effectiveness resulted in a dramatic strategic shift at the professional level [35]. The findings in sports analytics also extend outside the domain of sports [30]. For example, sports data has been used to exhibit human biases [3, 15] and to support game theoretical predictions of human behavior [8, 58]. The advent of the big-data era has led to promising advancements in sports analytics [30]. However, the collection and standardization of data remain core challenges [41].

Estimating the win probabilities of the competitors is one of the central ideas in sports analytics. Win probability estimation probably emanates from baseball analytics, with research dating back to the early 1960s [24]. In addition to baseball, win probability estimation has been applied to most major sports, with examples from basketball [28, 49], hockey [6, 34], American football [26, 33], and soccer [11, 40]. Recently, the advancements in sports data collection [41] and an increase in the popularity of sports betting [48] have motivated research on win probability estimation. Applications of win probability estimation vary from descriptive (e.g. performance evaluation and prediction) to prescriptive (e.g. the recommendation and optimization of strategies). The shift from descriptive to prescriptive analytics is a recent trend in sports analytics [41].

Lindsey [24] used statistical methods to model the progression of the score of

a baseball game based on a data set of roughly two thousand professional baseball matches from 1958 and 1959. By analyzing the score difference data, Lindsey [24] presented estimates for the teams' win probabilities based on the score of previous innings. This analysis aims to support baseball managers' decision-making during a match. By having estimates of their win probability, a winning team could rest their star players to avoid injuries when the comeback probability of the opposing team is at a satisfactorily low level [24].

In a more recent book on baseball analytics, Tango et al. [53] employed a Markov chain approach to analyze the game. Markov chain modeling assumes that the future state of a stochastic process only depends on its current state and is otherwise independent of its history [32]. Tango et al. [53] described baseball game states as permutations of the half-inning (unit of play), score difference, occupied bases, and the number of outs. The state-to-state transition probabilities can then be estimated from historical data to obtain a discrete-time Markov chain describing the game of baseball. Tango et al. [53] used this stochastic model to compute the win probability of the home team in each game state. The resulting win expectancy matrix enables the estimation of *win values* for each basic event in baseball (single, out, home run, etc.) [53]. These win values describe the average change in win probability when an event takes place, quantifying the value of each event in terms of win probability. Tango et al. [53] used these win probability estimates to analyze coaching decisions, debunk common myths, and generate strategic guidelines for baseball.

A similar Markov chain approach was taken by McFarlane [28] to evaluate end-of-game decision-making in the National Basketball Association (NBA). McFarlane [28] used logistic regression to estimate win probabilities during the last three minutes of a basketball game. These estimates were used to construct an *end-of-game tactics metric* (ETM) for comparing tactical decisions such as choosing between a two-point and a three-point field goal attempt. ETM is defined as the difference between the win probability of the actual decision and that of the theoretically optimal decision for the game state [28]. McFarlane [28] evaluated the end-of-game decisions made by NBA teams during the 2015–2016 season and showed that the mean ETM difference of the teams correlated well with their actual winning percentage in close games.

Win probability estimates have been used to evaluate tactical decisions in other sports as well. Lock and Nettleton [26] used random forests to estimate the play-by-play win probability in National Football League (NFL) games. Their random forest model included variables describing the past performance of the competing teams, allowing accurate win probability estimation even in the case of unevenly matched teams [26]. Lock and Nettleton [26] provided examples in American football where the estimated change in win probability could support the decision-making process of a coach. This focus on changes in win probability is similar to the win value analysis in baseball by Tango et al. [53]. Lock and Nettleton [26] experimented with multiple adjustments to their win probability model, including an attempt to account for the effects of momentum in American football. The momentum adjustment resulted in added complexity, but no improvement in performance [26], in line with the findings of other NFL studies [13, 19]. In addition to their in-depth coverage of win probability estimation in American football, Lock and Nettleton [26] proposed a general binning

method for measuring and visualizing the quality of win probability estimates. While not stated in the original paper, this binning method is a special case of reliability diagrams [31], which are used to evaluate the calibration of machine learning models [16]. Moreover, Lock and Nettleton [26] also describe the assessment of variable importance in a win probability model.

Table 2: Evaluation methods for win probability models.

Method	**Description**
Accuracy [16]	Proportion of correctly predicted outcomes.
ECE [16, 40]	Average difference between estimated win probability and true proportion of wins.
Reliability diagrams [31, 40]	Visualization of estimated win probability versus true proportion of wins.

In their research on in-game win probability estimation in soccer, Robberechts et al. [40] developed a Bayesian win probability model with an innate measure of prediction uncertainty. The frequent occurrence of ties and the low-scoring nature of soccer pose modeling challenges that are not present in the sports discussed above [40]. Robberechts et al. [40] overcame these challenges by modeling the future goal-scoring probabilities of both teams as independent Poisson distributions. These goal-scoring estimates are then mapped to corresponding win-draw-loss probabilities. The win probability model included both pre-game features and in-game features. These features are dynamically weighted, so that the in-game features rise in importance as the game progresses. This approach allowed the model to produce more accurate win probability estimates during the closing moments of a game [40].

Robberechts et al. [40] used reliability diagrams and expected calibration error (ECE) [16] to evaluate the calibration of their win probability model. Table 2 summarizes these methods of evaluating win probability models. Robberechts et al. [40] also analyzed feature importance in their model, following the footsteps of Lock and Nettleton [26]. Robberechts et al. [40] defined an *added goal value* (AGV) metric for evaluating soccer players based on their average contribution to their team's win probability. AGV relies on estimated changes in win probability, similar to win values [52] and ETM [28]. These win probability metrics are summarized in Table 3.

The analytical work mentioned in this section is merely an overview of the vast body of research on win probability estimation. Novel win probability models are being developed for previously relatively unexplored sports, such as cricket [2]. Based on the research described in this section, one can conclude that win probability estimates are a commonly used basis and globally accepted for evaluating players and decisions in sports. In the next section, win probability estimation is shown to have similar potential in the realm of esports.

Table 3: Win probability metrics used in traditional sports.

Metric	Description	Use
Win values [53]	Average change in win probability from game events.	Evaluating the impact of basic events in baseball.
ETM [28]	Win probability distance to an optimal decision.	Comparing tactical decisions at the end of a basketball game.
AGV [40]	A player's contribution to the team's win probability.	Evaluating player performance in soccer.

3.2 Win probability estimation in esports

Despite the popularity of esports, the field of esports analytics is still in its early stages [18, 42]. In contrast to traditional sports analytics [30], esports analytics does not face the same data collection challenges due to the inherently digital environment of esports [42]. Multiplayer online battle arena (MOBA) is among the most popular esports genres, with LoL being its biggest title [56]. This section, along with the rest of the thesis, focuses primarily on LoL due to the popularity of the game and the author's experience with it. Nevertheless, notable analytical work has been published for other major esports titles, such as Dota 2 [10, 17, 21] and PlayerUnknown's Battlegrounds [14, 25]. Due to major differences between the core game mechanics, it is pertinent to focus on the literature specific to LoL [12, 22, 27, 43, 60].

Win probability estimation is a relatively novel topic in LoL analytics, with few research papers dedicated specifically to it [22, 27]. However, extensive work has been done under the term *match outcome prediction*, which is typically treated as a binary classification task [5]. A binary match outcome classifier determines which outcome (win or loss) is more likely, without necessarily assigning explicit probabilities to the outcomes [20]. However, most binary classifiers include some numerical measure of confidence that can be mapped to a probability [20]. Thus, win probability estimation can be seen as an extension of match outcome prediction.

LoL match outcome prediction can be divided into two interesting subproblems: pre-game prediction [12, 60] and in-game prediction [22, 43]. Pre-game predictions are made after the draft phase of a match, using only information available before the actual gameplay commences. This includes information on the players' past performances and their chosen champions. In-game predictions are made using all the available information up to time $t \in [0, T)$, where 0 and T denote the start and end of a match, respectively. Note that LoL matches vary in duration, with typical matches lasting anywhere from 15 to 40 minutes. End-of-game prediction, i.e., outcome classification using all information available at time T is a comparatively uninteresting task, equivalent to asking which basketball team won when you know the final score.

Do et al. [12] trained and compared various machine learning models for pre-game match outcome prediction. Their novel contribution was to use measures of

player-champion experience as features in the models. The final experience measures—player-champion win rate and champion mastery points—were chosen using Pearson's correlation test. Do et al. [12] collected and used a data set of 5 000 ranked LoL matches from various skill levels. They achieved upwards of 72% pre-game prediction accuracy with multiple machine learning methods, including support vector machines, k-nearest neighbors, decision trees, and deep neural networks (DNN). Their DNN was throned the best due to its high validation accuracy (75.1%) and comparatively low standard error (0.6%) [12]. Significantly, these high pre-game prediction accuracies were achieved for matches governed by a fair matchmaking system, aimed at giving equal odds for both teams [38]. These results indicate that the draft phase, where each player selects a champion to play, is paramount for the outcome of a match. Furthermore, to maximize their win probability, players should only pick champions they are well-practiced on [12].

White and Romano [60] approached the task of pre-game prediction using logistic regression and a data set of 87 743 ranked LoL matches. Their research focused on the effects of psychological momentum in multi-match play sessions. The logistic regression model achieved 72.1% accuracy in pre-game prediction. The momentum effects of the players' previous matches were slight, increasing the pre-game prediction accuracy by 0.1–0.3% compared to a baseline model [60].

Silva et al. [43] published one of the first research papers on in-game outcome prediction for LoL matches. They approached the problem using recurrent neural networks (RNN) due to their suitability for prediction tasks involving time series data [43]. The choice of RNNs for match outcome prediction implies an assumption of momentum effects [13] in LoL. This is in contrast to the previously discussed Markov Chain modeling [28, 53], which relies on an assumption of independence of the game states [32]. Silva et al. [43] did not explicitly consider these assumptions in their paper, citing the sequential nature of the data as the primary reason for using RNNs. They used a data set consisting of 7 621 professional LoL matches played between 2015 and 2018. This match data is multimodal, including both categorical pre-game information and numerical in-game time series. The best RNN trained by Silva et al. [43] achieved in-game prediction accuracies ranging from 63.9% at the start of a match (t = 5 min) to 83.5% at the later stages of a match (t = 25 min).

The models discussed above were evaluated solely based on prediction accuracy. However, in order to obtain meaningful win probability estimates, a match outcome prediction model must be *well-calibrated* [16] in addition to being accurate [9]. Kim et al. [22] were the first to address the problem of confidence calibration in LoL match outcome prediction. They trained a DNN for in-game prediction and then calibrated the model using a novel method designed specifically for this task. The proposed method, *data uncertainty loss*, is a loss function [20] that aims to minimize calibration error by accounting for the inherent uncertainty in the matches [22]. The calibrated in-game prediction model achieved an accuracy of 73.8% (aggregated over all game times) and an ECE of 0.57%. This marks a significant improvement over the baseline; a similar uncalibrated model had an accuracy of 73.0% and an ECE of 4.47%. Using reliability diagrams and ECE, Kim et al. [22] showed that the data uncertainty method outperformed other, commonly used calibration methods. Temperature scaling [16]

was the second-best calibration method, yielding only slightly worse results than the data uncertainty method [22]. The details of all the prediction models discussed above are outlined in Table 4.

Table 4: Summary of LoL match outcome prediction models.

Prediction task	Method	Matches [#]	Accuracy [%]
Pre-game [12]	DNN	5 000	75.1
Pre-game [60]	Logistic regression	87 743	72.1
In-game [43]	RNN	7 621	63.9–83.5
In-game [22]	DNN	83 875	73.8

Maymin [27] developed an in-game win probability model to refine existing LoL gameplay metrics and introduce new ones. These advanced gameplay metrics were incorporated into a novel player-evaluation framework, which is designed to aid player improvement [27]. Riot Games provides public access to LoL match data through an application programming interface (API) [39]. However, this public data is quite limited in granularity, especially for advanced in-game analytics. Maymin [27] implemented custom software to extract more granular and comprehensive match data. The in-game win probability model used logistic regression and was trained on millions of matches [27]. To prevent highly correlated game states in the training data, Maymin [27] included only a random minute of data from each match. Maymin [27] also investigated the correlation between an individual's performance and their team's win probability. To this end, an end-of-game outcome prediction decision tree was constructed using the advanced gameplay metrics of an individual player. The decision tree achieved 80% validation accuracy, indicating that the advanced metrics succeed in measuring player performance [27]. The research paper did not include any measure of prediction accuracy or calibration error [16] for the in-game win probability model.

Although still limited in quantity, the research on win probability estimation in esports is promising. Recent papers [9, 22] have highlighted the importance of confidence calibration in providing useful win probability estimates. This is crucial because modern deep learning models—while attractive due to the ease of data collection in esports—are inherently poorly calibrated [16]. In theory, esports games make it possible to collect perfect information. In practice, however, the quality of the available data is often constrained by the game APIs. To overcome this issue and obtain higher-quality data, custom data collection methods can be developed, as demonstrated by Maymin [27]. Despite these advancements, strategy optimization, one of the focal points of sports analytics [1], remains relatively unexplored in esports. Fortunately, a shift to such prescriptive analytics can be expected as the field of esports research matures [41]. To conclude the background of this thesis, the following section discusses how esports analytics serves to benefit from the research on game-playing artificial intelligence.

3.3 Artificial intelligence and esports analytics

Games have long been a playground for artificial intelligence (AI) research. Game-playing AI systems were first developed for classic games like backgammon [54] and chess [7]. Recently, AI systems have matched and even exceeded the skill of top human players in video games such as Gran Turismo [61], StarCraft [57], and Dota 2 [4]. These video games pose challenges that mimic the complexity of the real world: high-dimensional environments, partial observability, and long time horizons [4, 57]. Increasingly complex video games help bridge the gap from the study of abstract games to useful applications in real-world domains [4].

Google DeepMind's AlphaGo algorithm [46] achieved notoriety when it defeated Lee Sedol, a world champion in the game of Go [47]. The first versions of AlphaGo used a combination of value and policy networks to evaluate and select moves [47]. The policy networks were initially trained using supervised learning on a data set of Go matches played by human experts [46]. AlphaGo's successor, AlphaGo Zero, completely abandoned the dependence on human knowledge, relying solely on reinforcement learning through self-play yet vastly surpassing its predecessors in performance [47]. Among many other technical challenges, Silver et al. [46] highlight the problem of successive game states being strongly correlated, which leads to overfitting of the value network if not properly addressed. This problem was mitigated by training the value network on millions of independent game states, each sampled from a unique match of self-play [46]. The same problem and solution appeared five years later in esports analytics [27].

The AlphaGo Zero algorithm was generalized and applied to chess and shogi (Japanese chess) under the name AlphaZero [45]. AlphaZero achieved superhuman performance [7] in both games within 24 hours of *tabula rasa* reinforcement learning through self-play [45]. A few years later, DeepMind developed a multi-agent reinforcement learning algorithm for StarCraft, a notoriously difficult real-time strategy game with a large, combinatorial action space [57]. The algorithm, named AlphaStar, reached Grandmaster level in StarCraft II, ranking above 99.8% of competitive human players [57]. This marked a significant milestone for AI research, as mastering the complex domain of StarCraft can be seen as a stepping stone towards even more difficult real-world applications [57].

There have been no notable published attempts at developing a superhuman AI agent for LoL, perhaps due to the absence of an official interface to the game engine that would facilitate the training of reinforcement learning models. Dota 2, however, includes an official scripting API designed for building game-playing programs [55]. This API was used by Berner et al. [4] in developing OpenAI Five, the first AI system to defeat the world champions at an esports game. Like LoL, Dota 2 is a five-on-five MOBA game where team coordination is vital for performance [51]. The OpenAI Five model consisted of five near-identical DNNs, each controlling one of the five heroes (the Dota 2 equivalent of champions) on the team [4]. These networks demonstrated collaborative behavior by concentrating the team's resources in the hands of its strongest members, a strategy seen in expert human play [4]. The observation embedding system of OpenAI Five [4] was hand-designed for the nuances of Dota 2,

similar to the StarCraft-specific architecture of AlphaStar [57].

As game-playing AI systems conquer more esports, we should expect a rise in the level of human play, similar to the historical effect of chess engines [7, 45]. The presence of these AI agents opens up a unique opportunity to learn from near-perfect players. This opportunity is specific to esports, at least for now, as traditional sports are physically constrained and require advanced robotics to mimic human play. Moreover, esports analytics serves to benefit from the vast body of game-playing AI research, as both fields face similar technical challenges and opportunities [27, 46].

4 Methodology

This chapter addresses the following decision-making problem: *Given a game state and a finite set of actions, which action should a player take to maximize their team's win probability?* The proposed method relies on having access to a large data set of matches, which is used to train a win probability estimation model and then compute statistics to evaluate the decision alternatives. The chapter attempts to formalize *win probability added* as a contextualized measure of value for strategic decision-making, using mathematical notation appropriate for contemporary esports.

4.1 Win probability in a zero-sum game

Consider a zero-sum game with two competing teams. Furthermore, assume that the game has only two possible outcomes (win and loss); there are no draws. Since the game is zero-sum, the teams have opposite goals [59]. Therefore, an increase in win probability for one team results in an equal decrease for the other. The ground-truth win probability cannot be determined for complex games like modern esports [9]; the win probability must be estimated.

Due to the symmetric nature of zero-sum games, it suffices to only consider the win probability from the perspective of one of the competing teams. Let $w(x)$ denote a win probability estimate at game state $x \in \mathcal{X}$, where $\mathcal{X}$ is the set of all possible states of the zero-sum game. For every game state x, there exists a mirrored state $x' \in \mathcal{X}$ such that $w(x') = 1 - w(x)$. The game states x and x' represent the same situation from the perspectives of each team. Let $\mathcal{T}_x$ denote the team from whose perspective x is given. Now, let $y \in \{0, 1\}$ be a dependent random variable representing the outcome of a match. From the perspective of team $\mathcal{T}_x$, $y = 1$ denotes a win and $y = 0$ a loss. Formally, $w \colon \mathcal{X} \to [0, 1]$ estimates the conditional probability of team $\mathcal{T}_x$ winning the match given a game state x, i.e.,

$$w(x) \approx \Pr(y = 1 \mid x). \tag{1}$$

By symmetry, the estimated win probability of the other team is $1 - w(x)$. From now on, $w(x)$ is referred to as the win probability at state x, remembering that the $w(x)$ is, firstly, an estimate, and secondly, given from the perspective of team $\mathcal{T}_x$.

The abundance of data makes machine learning models attractive for estimating win probabilities in esports. The technical implementation of a win probability model is always game-dependent; Bayesian modeling might be suitable for one game [40] and deep neural networks for another [22]. Win probability models are typically evaluated based on their accuracy [9], the proportion of correctly predicted outcomes. However, in addition to being accurate, a win probability should also be well-calibrated [16], as discussed in Chapter 3. A well-calibrated model produces unbiased win probability estimates that accurately reflect the true, unknown win probability (right side of Equation 1). Calibration can be determined empirically using, e.g., expected calibration error (ECE) [16] and reliability diagrams [31].

To evaluate the calibration of a win probability model, we need a data set of game states and corresponding win probability estimates. For the computation of ECE and

reliability diagrams, these win probability estimates are distributed into M bins. The number of bins should be chosen so that each bin contains sufficiently many samples; typically $M \in [5, 20]$ provides reliable results [9], but this depends on the number of samples N. For each bin B_m, we compute the mean win probability estimate $\overline{w}(B_m)$ and the expected outcome $\overline{y}(B_m)$, i.e., the proportion of wins in the bin. ECE is then the mean absolute difference between $\overline{y}(B_m)$ and $\overline{w}(B_m)$, weighted by the number of samples in each bin, i.e., the cardinality $|B_m|$;

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\overline{y}(B_m) - \overline{w}(B_m)|. \tag{2}$$

A well-calibrated model has a low ECE; ECE = 0 indicates perfect calibration. Reliability diagrams are generated by plotting the mean win probability estimate $\overline{w}(B_m)$ against the expected outcome $\overline{y}(B_m)$ for each bin B_m. Examples of reliability diagrams can be seen in Section 5.1.

4.2 Problem definition and challenges

Having formalized the concept of win probability, let us now elaborate the decision-making problem at hand:

> Given a game state $x^* \in \mathcal{X}$ and a finite set of actions $A = \{a_1, a_2, \ldots a_L\} \subseteq \mathcal{A}$, which action $a^* \in A$ should a player take to maximize their team's win probability, i.e., $a^* = \arg\max_{a \in A} \Pr(y = 1 \mid x^*, a)$?

The state space $\mathcal{X}$ and action space $\mathcal{A}$ are dependent on the zero-sum game being analyzed. The state space $\mathcal{X}$ can be continuous or discrete, but the action space $\mathcal{A}$ is assumed to be discrete to rule out continuous gameplay actions like movement in LoL. The size of the finite action set A is denoted by L. Using examples from LoL, the actions can be e.g. item purchases or events such as dragon kills and turret destructions. Similarly, $\mathcal{X}$ can be a high-dimensional vector space representing information about the in-game time, champion positions, destroyed towers, etc.

We are primarily interested in studying this problem from a strategic decision-making perspective. Here, strategic refers to the premeditated or retrospective nature of the analysis. We are not interested in automating player decision-making during a game, but in supporting it before and after the game. There are several challenges associated with the task:

1. Game state representation. In many contemporary esports, the game state is only partially observable during play [4]. This might be the case even post-game due to constraints of the game API [27]. Moreover, the state spaces of modern esports are often high-dimensional and multimodal [4], which makes choosing the representation $\mathcal{X}$ difficult. Simplified game state representations are warranted as long as they contain sufficient information for reliable win probability estimates.

2. Complexity of actions. The complexity of actions can vary significantly depending on the specific decision being analyzed. We are particularly interested in complex actions that are difficult or impossible to model using the state-to-state

transitions typical of Markov decision processes [50]. Due to the complicated rules and long time horizons of modern esports [4], the effects of actions can often be uncertain. Some actions offer minimal immediate benefit, with their value appearing later at unpredictable times. For example, Mejai's Soulstealer, an item in LoL, provides only minor benefits at the time of purchase. However, due to its *Glory* effect, it can quickly transform into one of the most powerful items in the game.

3. Variance and external factors. In multiplayer games, the choices of one player have a limited impact on the overall win probability of a team. External factors, such as the performance of teammates and opponents, can obscure the true impact of individual actions. The effects of individual actions are further diminished and obscured as the number of players increases. Moreover, some esports (e.g. Hearthstone) even incorporate randomness as a fundamental game mechanic. Thus, assessing the impact of individual actions is often difficult.

4. Action selection bias. Esports games make it possible to collect massive data sets of publicly played matches, providing an opportunity to evaluate actions through aggregate statistics. However, this data is significantly affected by selection bias, since the players are using their biased judgment to select which action to take. Some actions are only taken by players to secure wins when they are already ahead or as a last resort in desperate situations, inflating or deflating their win rates, respectively. Moreover, esports communities often share popular strategies, which can result in the overuse or misuse of some popularly recommended strategies, making the associated actions seem worse than they truly are.

The next section addresses these challenges by introducing a contextualized method of evaluating actions.

4.3 Contextualized evaluation of actions

The proposed method of evaluating actions requires a data set of actions with their contexts extracted from historical match data. We denote this data set of actions with $D = \{d^{(1)}, d^{(2)}, \ldots, d^{(N)}\}$, where every data point $d^{(i)} = (a^{(i)}, x^{(i)}, z^{(i)}) \in \mathcal{A} \times \mathcal{X} \times \mathcal{X}$ is a triple consisting of an action a, an initial state x, and a final state z. In the i-th data point, the initial state $x^{(i)}$ is the game state in which action $a^{(i)}$ first took effect. The final state $z^{(i)}$ is the last game state affected by $a^{(i)}$. Depending on the game and actions in consideration, determining z can be difficult or impossible. The effects of an action can also last until the end of a match. In these cases, z should by default be set to the terminal game state, i.e., the last known state of the match.

Continuing with examples from LoL, let us consider an action space consisting of all purchasable items in LoL. In this example, the action data set D is a list of item purchases from matches collected through the LoL API [39]. A data point $d = (a, x, z) \in D$ describes an item purchase made by a player in one of the collected matches. a is the item purchased, x is the game state at the time of purchase, and z is the game state when the item was sold, destroyed, or upgraded. If the item remains in the player's inventory until the end of the match, we use the terminal game state as z. For every data point $d^{(i)} = (a^{(i)}, x^{(i)}, z^{(i)}) \in D$, we define two quantities:

1. $W(d^{(i)})$, the initial win probability in $d^{(i)}$;

$$W(d^{(i)}) = w(x^{(i)}). \tag{3}$$

2. $\Delta W(d^{(i)})$, the win probability added in $d^{(i)}$;

$$\Delta W(d^{(i)}) = w(z^{(i)}) - w(x^{(i)}). \tag{4}$$

Here, $w\colon \mathcal{X} \to [0,1]$ is a win probability estimate as defined in Section 4.1. The initial win probability $W\colon \mathcal{A} \times \mathcal{X} \times \mathcal{X} \to [0,1]$ and the win probability added $\Delta W\colon \mathcal{A} \times \mathcal{X} \times \mathcal{X} \to [-1,1]$ form the basis for our contextualized evaluation method. $W(d^{(i)})$ encapsulates the initial state $x^{(i)}$ in which action $a^{(i)}$ was taken into one win probability. $\Delta W(d^{(i)})$ measures the change in win probability during the effective window of $a^{(i)}$. These quantities are illustrated in Figure 4.

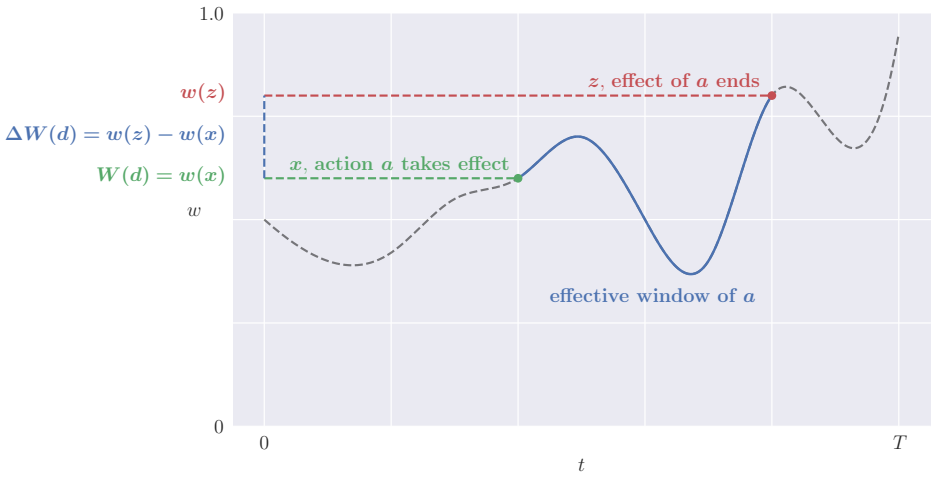

Figure 4: A hypothetical match progression illustrating the initial win probability $W(d)$ and the win probability added $\Delta W(d)$. The horizontal axis represents the in-game time $t \in [0,T]$, and the vertical axis the estimated win probability w.

The initial win probability $W(d^{(i)})$ reflects the context of an individual action $a^{(i)}$ but it does not contain information about the impact of $a^{(i)}$. Moreover, the win probability added $\Delta W(d^{(i)})$ is not a reliable estimate for the impact of $a^{(i)}$; $\Delta W(d^{(i)})$ is affected by variance and external factors, as discussed in Section 4.2. In order to mitigate these adverse effects and generate meaningful insights, these quantities must be averaged over many similar data points in $D = \{d^{(1)}, \ldots, d^{(N)}\}$. Thus, we define two statistics for any action $a \in \mathcal{A}$ and for any set of game states $X \subseteq \mathcal{X}$:

1. $\overline{W_X}(a)$, the mean initial win probability of a in X;

$$\overline{W_X}(a) = \frac{1}{|I_X(a)|} \sum_{i \in I_X(a)} W(d^{(i)}). \tag{5}$$

2. $\overline{\Delta W_X}(a)$, the mean win probability added by a in X;

$$\overline{\Delta W_X}(a) = \frac{1}{|I_X(a)|} \sum_{i \in I_X(a)} \Delta W(d^{(i)}). \tag{6}$$

Here, $I_X(a) = \{i \in \{1, \ldots, N\} \mid a^{(i)} = a \wedge x^{(i)} \in X\}$ is a set of all the indices i for data points $d^{(i)} \in D$ where the action $a^{(i)}$ is a and the initial state $x^{(i)}$ is an element of the state set X. The cardinality $|I_X(a)|$ is the associated sample size. While the action space $\mathcal{A}$ is assumed to be discrete, the state space $\mathcal{X}$ can be continuous, necessitating the use of the state set X to group similar states. For the original decision-making problem, we can define a state set X^* consisting of states similar to x^*. The definition of X^* naturally depends on the state space representation $\mathcal{X}$ of the game in consideration.

The mean initial win probability $\overline{W_X}(a)$ measures the systemic bias in data points where action a is selected in specific game states $x \in X$. $\overline{W_X}(a) > 0.5$ indicates that action a is selected more often in winning situations. Conversely, $\overline{W_X}(a) < 0.5$ indicates that a is selected more often in losing situations. The mean win probability added $\overline{\Delta W_X}(a)$ measures the average impact of taking action a in the context of X. $\overline{\Delta W_X}(a) > 0$ indicates that taking action a in game states $x \in X$ is beneficial on average and $\overline{\Delta W_X}(a) < 0$ the opposite. According to the law of large numbers, $\overline{\Delta W_X}(a)$ approaches the true win probability added by taking action a in a game state $x \in X$ as the associated sample size $K = |I_X(a)|$ approaches infinity and X approaches the singleton set $\{x\}$ i.e.,

$$\lim_{K \to \infty} \lim_{X \to \{x\}} \overline{\Delta W_X}(a) = \Pr(y = 1 \mid x, a) - \Pr(y = 1 \mid x). \tag{7}$$

Thus, given a sufficiently large sample size K and a sufficiently specific (similar to x^*) state set X^*, we can approximate the original decision-making problem by replacing the true win probability $\Pr(y = 1 \mid x^*, a)$ with the mean win probability added $\overline{\Delta W_{X^*}}(a)$, i.e.,

$$\max_{a \in A} \Pr(y = 1 \mid x^*, a) \tag{8}$$

$$\equiv \max_{a \in A} \Pr(y = 1 \mid x^*, a) - \Pr(y = 1 \mid x^*) \tag{9}$$

$$\cong \max_{a \in A} \overline{\Delta W_{X^*}}(a). \tag{10}$$

Under these assumptions, given a game state $x^* \in \mathcal{X}$ and a finite set of actions $A = \{a_1, a_2, \ldots a_L\} \subseteq \mathcal{A}$, the player should select the action with the highest mean win probability added in states similar to x^*, i.e., $\arg\max_{a \in A} \overline{\Delta W_{X^*}}(a)$, to maximize their team's win probability.

5 Case study

In this chapter, we describe the data set and model used to estimate LoL win probabilities. The performance of the model is evaluated based on accuracy, ECE, and reliability diagrams. Then, we apply the contextualized evaluation method described in Chapter 4 to find the best items in different situations for one champion, Shen.

5.1 In-game win probability model

Data and model description

In Section 3.2, we reviewed two notable papers addressing the task of in-game match outcome prediction for LoL [22, 43]. We follow the approach of Kim et al. [22] and train a deep neural network (DNN) to estimate the win probability given a game state. In this approach, the win probability is estimated based on a single game state; the previous game states are not considered, aligning with the independence assumption of Markov chain modeling [26]. Thus, we do not account for any in-game momentum effects, but previous research has shown that these effects are negligible, at least in traditional sports [13, 19].

For this case study, we used the LoL API [39] to collect a data set of 400 000 Ranked matches played on Patch 14.15. We use 350 000 matches to train the DNN and reserve 50 000 matches for testing. Because of the limitations of the LoL API [39], our match data contains only one game state per minute by default. Additionally, we have the game state at every important game event (kill, turret, dragon, etc.). The frequency of these events increases as the game progresses. This results in a non-uniform time distribution of game states, which is further skewed by LoL's varying match duration. The average match duration in our data set is roughly 25 minutes. Instead of training the DNN using every state in every match, we divide matches into 5-minute intervals and sample one game state uniformly per interval. In addition to removing the non-uniformity caused by event timing, the uniform random sampling reduces overfitting due to the correlation of successive game states [27, 46]. However, this sampling method does not affect the skewness caused by the varying match duration; the sample sizes are significantly smaller in the *late game*, i.e., 30 min and onwards.

The primary focus of this thesis is on the application of win probability estimation and not its game-specific technical implementation. Thus, the precise DNN architecture is not described here. Kim et al. [22] describe the implementation and calibration of a similar in-game win probability model for LoL in detail. Nevertheless, Table 5 presents the most important explanatory variables (features) of the DNN. The model relies primarily on in-game features, with *Rank* being the only pre-game feature. Notably, we do not use items as explanatory variables due to their categorical nature. The added complexity of 60 categorical variables (six item slots per champion) is deemed imprudent despite the potential increase in predictive power. The exclusion of items does not affect their analysis using win probability added, as the action space $\mathcal{A}$ is distinct from the state space $\mathcal{X}$, i.e., the explanatory variables of the model.

Table 5: Overview of the most important features of the win probability model. Player-specific variables are repeated ten times, once for each player.

Variable	Description
Position	Current 2D map coordinates of each player.
Level	Current champion level per player.
Gold	Cumulative amount of gold acquired per player.
Damage Dealt	Cumulative amount of damage dealt per player.
Damage Taken	Cumulative amount of damage taken per player.
Kills	Cumulative number of champion kills per player.
Assists	Cumulative number of champion assists per player.
Deaths	Cumulative number of deaths per player.
Voidgrubs	Cumulative number of Voidgrubs killed per team.
Dragons	Cumulative number of Dragons killed per team.
Barons	Cumulative number of Baron Nashors killed per team.
Turrets	Cumulative number of turrets destroyed per team.
Inhibitors	Cumulative number of inhibitors destroyed per team.
Time	Current in-game time.
Rank	Average rank (skill level) of the players.

Model performance

The performance of the model is evaluated on the test set consisting of 50 000 matches with 5 653 687 game states in total. On this set, the model achieves an aggregate accuracy of 75.9% with an ECE of 0.90%. The ECE is computed with $M = 20$ bins (see Equation 2). Table 6 presents these performance metrics alongside those of similar in-game prediction models from the literature. The implemented model surpasses the previous models in accuracy but has a slightly worse ECE than the calibrated DNN by Kim et al. [22].

Table 6: In-game prediction model performance comparison.

Model	Matches [#]	Accuracy [%]	ECE [%]
RNN [43]	7 621	63.9–83.5	-
Baseline DNN [22]	83 875	73.0	4.47
Calibrated DNN [22]	83 875	73.8	0.57
Presented DNN [This thesis]	350 000	75.9	0.90

In order to investigate the effect of in-game time on the accuracy of the model, we split the test states into 4-minute intervals and compute the accuracy in each interval until 40 minutes, visualized in Figure 5. The accuracy starts at 55.8% and increases with time, peaking at 84.8% in the $[24, 28]$-minute interval. The accuracy decreases in the last three time intervals, likely due to the model having less training data in these match stages. Alternatively, the dynamics of LoL might cause the game to become less predictable in the late game; the Elder Dragon combined with long death timers create *comeback* opportunities for the losing teams, possibly increasing the match outcome variance.

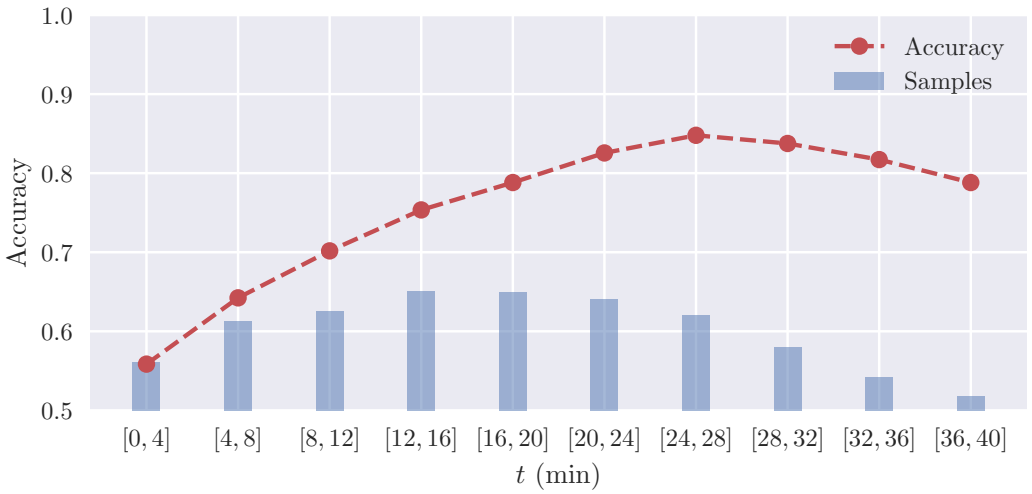

Figure 5: A plot of the accuracy of the model in 4-minute in-game time intervals. The histogram representing the number of samples in each time interval is normalized so that the bar lengths add up to 1.

Figure 6 presents four reliability diagrams illustrating the calibration of the win probability model. The test data is now divided into four time intervals, spanning 10 minutes each. The reliability diagrams are generated by plotting the mean win probability estimate $\overline{w}$ against the expected outcome $\overline{y}$, i.e., the proportion of wins, for each win probability bin. We use $M = 20$ bins to align with the ECE computation. The calibration declines slightly with time, but the model is overall well-calibrated.

The reliability diagrams indicate that the model provides reliable win probability estimates, especially in the early game ($t \in [0, 10]$ min), where the win probability distribution is light-tailed. The histograms in Figure 6 show how the distribution of win probabilities changes with time. The distributions are similar to those observed by Choi et al. [9], who showed that LoL win probabilities can be modeled as symmetric beta distributions with time-dependent parameters. As the game progresses, the tails of the win probability distribution become heavier and the win probability estimates become less reliable. This finding coincides with the previously postulated effects of late-game outcome variance.

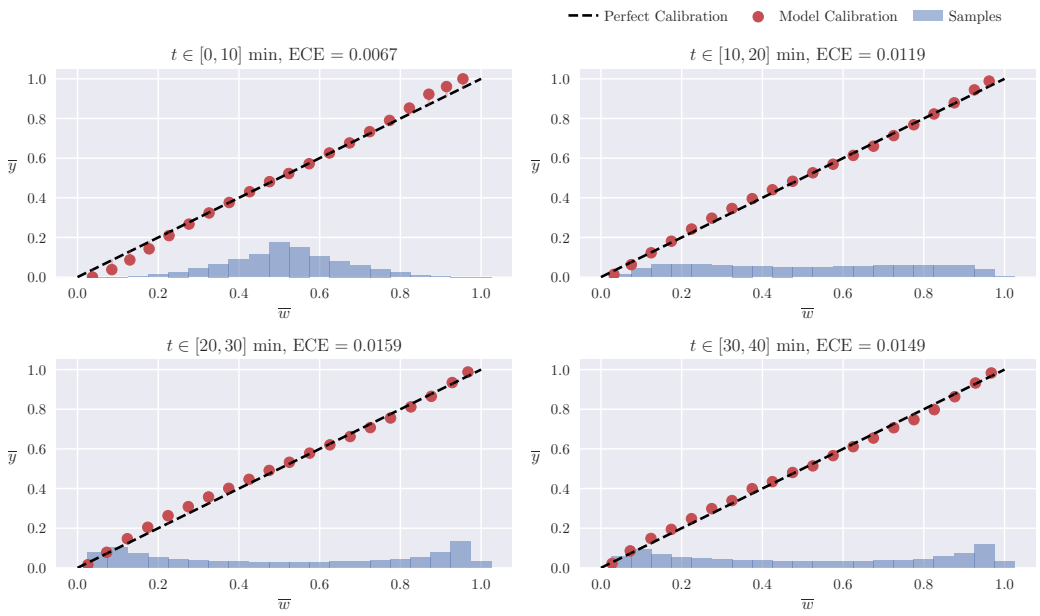

Figure 6: Four reliability diagrams illustrating the calibration of the model. The titles indicate the in-game time interval and the associated ECE value ($M = 20$).

5.2 Contextualized evaluation of items

Item system and problem definition

As mentioned in Chapter 2, *items* are a core gameplay system in LoL. Items can be purchased using gold, increasing champion statistics and granting unique effects. There are roughly two hundred purchasable items in LoL, divided into three main categories: basic items, epic items, and legendary items. These categories are sorted in increasing order of power and cost. Basic items are cheap and relatively weak, but they can be *upgraded* into epic items, which can in turn be upgraded into legendary items. Upgrading is the process of transforming items into more powerful ones using gold; basic and epic items are *components*, i.e., intermediate parts, of legendary items. Most legendary items have two to three required components. However, a player's inventory can only hold six items at a time, which makes it difficult to buy the components for multiple legendary items simultaneously. Thus, players typically choose which legendary item they want to obtain next and then purchase the components leading up to it. Due to these reasons, we are primarily interested in analyzing the 107 legendary items currently in the game.

Items generally provide the same benefits, no matter the champion in play. However, every champion interacts with these benefits in different ways; an item effect can be useless for one champion and extremely valuable for another. Thus, it is reasonable to evaluate items in the context of a specific champion. The teammates and opponents also affect the value of items. In the early stages of a game, it is common to build an item to counter your lane opponent, the enemy champion that you are fighting most often. In the following analysis, we focus on items specifically for Shen, a champion

primarily played in the top lane. This choice is motivated by the author's extensive experience playing Shen, allowing for a more thorough qualitative analysis of the results.

The effects of items are not independent; some items have synergistic effects with other items, while purchasing multiple items with similar effects leads to diminishing returns. When a player is choosing their next item, they thus have to consider the items already in their inventory. From a combinatorial perspective, while there are only 107 legendary items, the number of six-item permutations is $107!/101! \approx 1.3 \cdot 10^{12}$. To avoid this complexity, we will only consider the choice of the first legendary item in the following analysis. Here is the final problem description with all its simplifications:

> Given the opposing top lane champion, which legendary item should Shen purchase first to maximize his team's win probability?

Evaluation of Shen's first legendary items

Let us begin by evaluating the first legendary items for Shen without the added complexity of a lane opponent. We will consider the four most popular legendary items for Shen: Sunfire Aegis, Titanic Hydra, Hollow Radiance, and Heartsteel. Using the in-game win probability model, we compute the mean initial win probabilities $\overline{W}$ and mean win probabilities added $\overline{\Delta W}$ for these items. Here, $\overline{W}$ and $\overline{\Delta W}$ are shorthand notation for $\overline{W_X}(a)$ and $\overline{\Delta W_X}(a)$, where $a \in A = \{\text{Sunfire Aegis, Titanic Hydra, Hollow Radiance, Heartsteel}\}$ and $X = \{\text{all game states where Shen purchases his first legendary item}\}$. Additionally, the win rate $\overline{y}$ and the sample size $K = |I_X(a)|$ are provided for context. These statistics are computed using a data set with 8 206 item purchases matching our criteria. Table 7 presents the results of the initial evaluation with no lane opponent information.

Table 7: Statistics of the most popular items for Shen in the top lane. The items are sorted by the mean win probability added $\overline{\Delta W}$. The mean initial win probability $\overline{W}$, the win rate $\overline{y}$, and the sample size K are provided for context.

Item	$\overline{\Delta W}$ [%]	$\overline{W}$ [%]	$\overline{y}$ [%]	K [#]
Titanic Hydra	1.46	51.41	51.94	2189
Heartsteel	0.66	53.66	54.15	1411
Sunfire Aegis	-0.51	50.53	49.10	3004
Hollow Radiance	-0.84	52.76	51.25	1602

Let us first consider the average impact of each item by looking at the mean win probability added $\overline{\Delta W}$. Titanic Hydra seems significantly better than the other items, with a $\overline{\Delta W}$ of 1.46%. Heartsteel comes in second with a $\overline{\Delta W}$ of 0.66%. The mean initial win probability $\overline{W}$ of Heartsteel (53.66%) is significantly higher than that of

Titanic Hydra (51.41%), indicating that Heartsteel is purchased in more winning situations than Titanic Hydra. This is to be expected as Heartsteel's effect, *Colossal Consumption*, is better the earlier it is purchased, making it an attractive item when ahead. Greedily considering only the win rate $\overline{y}$ of an item could lead a player to believe that Heartsteel is the best alternative because it wins the most amount of matches (54.15%). However, the win rate is biased; it fails to account for the context in which Heartsteel is selected. The remaining two items, Sunfire Aegis and Hollow Radiance, have negative $\overline{\Delta W}$ values, -0.51% and -0.84%, respectively. It is possible to argue for purchasing Sunfire Aegis in specific situations because it has the lowest $\overline{W}$ and highest sample size K; it is the default option for Shen players and is thus incorrectly being bought in every scenario, possibly deflating its $\overline{\Delta W}$. On the other hand, Hollow Radiance has a higher $\overline{W}$ (52.76%) and a much lower $\overline{\Delta W}$ than Titanic Hydra. Thus, Hollow Radiance is dominated by Titanic Hydra.

Let us now increase the specificity by narrowing the state set X based on the lane opponent of Shen. Top lane champions can be divided into two categories based on their primary damage type, which is either *physical damage* or *magic damage*. Two state sets are defined accordingly: X_{phys} and X_{magic}. The total sample sizes for these sets are $|I_{X_{\text{phys}}}| = 5994$ and $|I_{X_{\text{magic}}}| = 2212$; physical damage champions are more common in the top lane. Tables 8 and 9 present the results of the evaluation based on the opposing laner's damage type.

Table 8: Statistics of the most popular items for Shen vs. physical damage top laners. The bolded values differ noticeably from the general values of Table 7.

Item	$\overline{\Delta W}$ **[%]**	$\overline{W}$ **[%]**	$\overline{y}$ **[%]**	K **[#]**
Titanic Hydra	**1.94**	51.49	52.36	1696
Heartsteel	**0.23**	53.49	53.39	1107
Sunfire Aegis	-0.68	50.34	48.69	2791
Hollow Radiance	-0.71	53.01	51.25	**400**

Table 9: Statistics of the most popular items for Shen vs. magic damage top laners.

Item	$\overline{\Delta W}$ **[%]**	$\overline{W}$ **[%]**	$\overline{y}$ **[%]**	K **[#]**
Heartsteel	2.24	54.26	56.91	304
Sunfire Aegis	1.75	53.04	54.46	213
Titanic Hydra	-0.21	51.16	50.51	493
Hollow Radiance	-0.88	52.67	51.25	1202

The item statistics versus physical damage are almost identical to the general item statistics in Table 7. This is not unexpected, as X_{phys} comprises 73% of all the data. Nevertheless, three differences stand out in Table 8: the mean win probability added $\overline{\Delta W}$ of Titanic Hydra (1.46% → 1.94%) and Heartsteel (0.66% → 0.23%), and the sample size K of Hollow Radiance (1602 → 400). Based on $\overline{\Delta W}$, Titanic Hydra is the predominant legendary item for Shen versus physical damage champions. This result is not surprising to most LoL players; Titanic Hydra covers Shen's main weakness (wave clear) and synergizes with Shen's Q Ability, Twilight Assault. The margin by which Titanic Hydra surpasses the alternatives is nonetheless noteworthy.

The relative sample size of Hollow Radiance decreased from 19.5% to 6.7% as we specified the opponent to be a physical damage champion. This change is credited to the *magic resistance* provided by Hollow Radiance, which reduces magic damage taken by the purchaser. Magic resistance does not affect physical damage, making it ineffective versus physical damage champions. For comparison, the relative sample size of Hollow Radiance is 54.5% in Table 9, indicating its popularity versus magic damage top laners. Despite its popularity, Hollow Radiance, with a low $\overline{\Delta W}$ of -0.88%, is outperformed by the alternatives. Moreover, Sunfire Aegis, an item that provides *physical resistance*, is evidently better versus magic damage than physical damage. This result is surprising and cannot be fully explained without more data. The high $\overline{W}$ (53.04) and the low sample size (213) of Sunfire Aegis indicate that the item is only purchased versus magic damage when Shen's team is winning. In these scenarios, Shen can itemize against the opposing team—most likely consisting of physical damage champions—instead of his lane opponent, making Sunfire Aegis a viable alternative.

In Table 9, the order of Heartsteel and Titanic Hydra is reversed compared to the previous tables; Heartsteel has the highest $\overline{\Delta W}$ value yet (2.24%) while Titanic Hydra falls into the negatives for the first time (-0.21%). Heartsteel is still systemically bought when Shen's team has a lead ($\overline{W}$ = 54.26%), but the high $\overline{\Delta W}$ indicates that Heartsteel succeeds in furthering that lead versus magic damage top laners. Nevertheless, the results of Table 9 are tentative, and more data should be gathered to provide recommendations for Shen's itemization versus magic damage champions.

5.3 Summary of the results

To summarize the results of this LoL case study, our in-game win probability model achieved 75.9% accuracy and 0.90% ECE on unseen test data, rivaling the performance of similar models found in the literature. Applying the contextualized evaluation method presented in Chapter 4, we studied the choice of the first legendary item for Shen. Based on the mean win probability added, Titanic Hydra was deemed the best item for Shen against physical damage lane opponents. Surprisingly, the most popular item against magic damage lane opponents, Hollow Radiance, was discovered to also be the worst alternative.

6 Conclusions

At the start of this thesis, we set out to explore the use of win probability estimation in esports with the objective of applying it to strategy optimization. This focal point of sports analytics remains relatively unexplored for esports, as discussed in Chapter 3. In Chapter 4, we formalized win probability as a metric for strategic decision-making, using mathematical notation appropriate for contemporary esports. We defined two statistics for the contextualized evaluation of decision alternatives: the mean initial win probability and the mean win probability added. These statistics were applied to real esports data in Chapter 5, where we studied the choice of the first legendary item for Shen, a champion in LoL, using a custom in-game win probability model that rivaled the performance of similar models from the literature. This case study, while niche, showed that win probability estimation can be used to generate novel insights that can hopefully lead to better strategic decision-making in esports.

The abundance of data makes esports an attractive field for deep learning. In addition to esports analytics, deep learning models have been used in game-playing systems, such as OpenAI Five [4]. The OpenAI Five model famously beat the Dota 2 world champions in 2019, achieving superhuman performance [7] in a complex five-on-five game. Interestingly, the OpenAI Five team completely ignored the itemization problem studied in this thesis; the AI agents were designed to choose the most popular items by default [4]. However, this does not mean that the choice of items is irrelevant. Rather, it highlights the different objectives of esports analytics and game-playing AI research; esports analytics focuses on evaluating and improving human performance while game-playing AI systems are designed to solve problems that have been historically difficult for machines. Nevertheless, the two fields share similar technical challenges and will continue to grow intertwined.

Recently, the emphasis in sports analytics has shifted from descriptive (e.g. performance evaluation) to prescriptive analytics (e.g. strategy optimization) [41]. This thesis is a step toward prescriptive analytics in esports, hopefully inspiring others to analyze strategic decisions in their favorite games. The proposed win probability statistics provide more insights than their biased predecessors, e.g., win rate. Nevertheless, they suffer from a fundamental trade-off between specificity and sample size. Future research could study the optimal choice of the state set X, balancing the specificity and reliability of the statistics. Despite their value, athletes and coaches should not follow statistics blindly; domain knowledge and intuition are required to generate meaningful insights and make the best decisions.

References

[1] B. Alamar. *Sports Analytics: A Guide for Coaches, Managers, and Other Decision Makers*. Columbia University Press, 2013.

[2] M. Asif and I. G. McHale. "In-play forecasting of win probability in one-day international cricket: A dynamic logistic regression model". *International Journal of Forecasting* 32.1 (2016), pp. 34–43.

[3] M. Bar-Eli, O. H. Azar, I. Ritov, Y. Keidar-Levin, and G. Schein. "Action bias among elite soccer goalkeepers: The case of penalty kicks". *Journal of Economic Psychology* 28.5 (2007), pp. 606–621.

[4] C. Berner et al. "Dota 2 with large scale deep reinforcement learning". *arXiv preprint* arXiv:1912.06680 (2019).

[5] K. U. Birant and D. Birant. "Multi-objective multi-instance learning: A new approach to machine learning for esports". *Entropy* 25.1 (2023), article 28.

[6] S. E. Buttrey, A. R. Washburn, and W. L. Price. "Estimating NHL scoring rates". *Journal of Quantitative Analysis in Sports* 7.3 (2011), pp. 1–18.

[7] M. Campbell, A. J. Hoane, and F.-h. Hsu. "Deep Blue". *Artificial Intelligence* 134.1 (2002), pp. 57–83.

[8] P.-A. Chiappori, S. Levitt, and T. Groseclose. "Testing mixed-strategy equilibria when players are heterogeneous: The case of penalty kicks in soccer". *American Economic Review* 92.4 (2002), pp. 1138–1151.

[9] E. Choi, J. Kim, and W. Lee. "Rethinking evaluation metric for probability estimation models using esports data". *arXiv preprint* arXiv:2309.06248 (2023).

[10] N. Clark, B. Macdonald, and I. Kloo. "A Bayesian adjusted plus-minus analysis for the esport Dota 2". *Journal of Quantitative Analysis in Sports* 16.4 (2020), pp. 325–341.

[11] T. Decroos, V. Dzyuba, J. V. Haaren, and J. Davis. "Predicting soccer highlights from spatio-semporal match event streams". *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. AAAI, 2017, pp. 1302–1308.

[12] T. D. Do, S. I. Wang, D. S. Yu, M. G. McMillian, and R. P. McMahan. "Using machine learning to predict game outcomes based on player-champion experience in League of Legends". *Proceedings of the 16th International Conference on the Foundations of Digital Games*. ACM, 2021, pp. 1–5.

[13] M. J. Fry and F. A. Shukairy. "Searching for momentum in the NFL". *Journal of Quantitative Analysis in Sports* 8.1 (2012), pp. 1–20.

[14] N. F. Ghazali, N. Sanat, and M. A. As'ari. "Esports analytics on PlayerUnknown's Battlegrounds player placement prediction using machine learning approach". *International Journal of Human and Technology Interaction* 5.1 (2021), pp. 17–28.

[15] T. Gilovich, R. Vallone, and A. Tversky. "The hot hand in basketball: On the misperception of random sequences". *Cognitive Psychology* 17.3 (1985), pp. 295–314.

[16] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. "On calibration of modern neural networks". *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. PMLR, 2017, pp. 1321–1330.

[17] V. J. Hodge, S. Devlin, N. Sephton, F. Block, P. I. Cowling, and A. Drachen. "Win prediction in multiplayer esports: Live professional match prediction". *IEEE Transactions on Games* 13.4 (2021), pp. 368–379.

[18] D. Jeong and S. Youk. "Refining esports: A quantitative cartography of esports literature". *Entertainment Computing* 47 (2023), article 100597.

[19] A. W. Johnson, A. J. Stimpson, and T. K. Clark. "Turning the tide: Big plays and psychological momentum in the NFL". *Proceedings of the 6th Annual MIT Sloan Sports Analytics Conference*. MIT Sloan, 2012.

[20] A. Jung. *Machine Learning: The Basics*. Springer Nature, 2022.

[21] C. H. Ke et al. "DOTA 2 match prediction through deep learning team fight models". *2022 IEEE Conference on Games*. IEEE, 2022, pp. 96–103.

[22] D.-H. Kim, C. Lee, and K.-S. Chung. "A confidence-calibrated MOBA game winner predictor". *2020 IEEE Conference on Games*. IEEE, 2020, pp. 622–625.

[23] League of Legends Wiki Community. *League of Legends Wiki*. 2024. URL: https://leagueoflegends.fandom.com/wiki/League_of_Legends_Wiki (visited on 07/23/2024).

[24] G. R. Lindsey. "The progress of the score during a baseball game". *Journal of the American Statistical Association* 56.295 (1961), pp. 703–728.

[25] J. Liu, J. Huang, R. Chen, T. Liu, and L. Zhou. "A two-stage real-time prediction method for multiplayer shooting e-sports". *Proceedings of the 20th International Conference on Electronic Business*. ICEB, 2020, pp. 9–18.

[26] D. Lock and D. Nettleton. "Using random forests to estimate win probability before each play of an NFL game". *Journal of Quantitative Analysis in Sports* 10.2 (2014), pp. 197–205.

[27] P. Z. Maymin. "Smart kills and worthless deaths: esports analytics for League of Legends". *Journal of Quantitative Analysis in Sports* 17.1 (2021), pp. 11–27.

[28] P. McFarlane. "Evaluating NBA end-of-game decision-making". *Journal of Sports Analytics* 5.1 (2019), pp. 17–22.

[29] T. W. Miller. *Sports Analytics and Data Science: Winning the Game with Methods and Models*. FT Press, 2015.

[30] E. Morgulev, O. H. Azar, and R. Lidor. "Sports analytics and the big-data era". *International Journal of Data Science and Analytics* 5.4 (2018), pp. 213–222.

[31] A. Niculescu-Mizil and R. Caruana. "Predicting good probabilities with supervised learning". *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 625–632.

[32] J. R. Norris. *Markov Chains*. Cambridge University Press, 1998.

[33] K. Pelechrinis. "iWinRNFL: A simple, interpretable & well-calibrated in-game win probability model for NFL". *arXiv preprint* arXiv:1704.00197 (2018).

[34] S. Pettigrew. "Assessing the offensive productivity of NHL players using in-game win probabilities". *Proceedings of the 9th Annual MIT Sloan Sports Analytics Conference*. Vol. 2. MIT Sloan, 2015, article 8.

[35] J. Poropudas and T. Halme. "Dean Oliver's four factors revisited". *arXiv preprint* arXiv:2305.13032 (2023).

[36] C. Reep and B. Benjamin. "Skill and chance in association football". *Journal of the Royal Statistical Society, Series A* 131.4 (1968), pp. 581–585.

[37] Riot Games. *Dev Diary: Win Probability Powered by AWS at Worlds*. 2023. URL: https://lolesports.com/en-GB/news/dev-diary-win-probability-powered-by-aws-at-worlds (visited on 06/29/2024).

[38] Riot Games. *League of Legends Support: Matchmaking and Autofill*. 2023. URL: https://support-leagueoflegends.riotgames.com/hc/en-us/articles/201752954-Matchmaking-and-Autofill (visited on 06/27/2024).

[39] Riot Games. *Riot Developer Portal: League of Legends API*. 2024. URL: https://developer.riotgames.com/docs/lol (visited on 06/27/2024).

[40] P. Robberechts, J. Van Haaren, and J. Davis. "A Bayesian approach to in-game win probability in soccer". *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, 2021, pp. 3512–3521.

[41] V. Sarlis and C. Tjortjis. "Sports analytics — Evaluation of basketball players and team performance". *Information Systems* 93 (2020), article 101562.

[42] M. Schubert, A. Drachen, and T. Mahlmann. "Esports analytics through encounter detection". *Proceedings of the 10th Annual MIT Sloan Sports Analytics Conference*. MIT Sloan, 2016.

[43] A. L. C. Silva, G. L. Pappa, and L. Chaimowicz. "Continuous outcome prediction of League of Legends competitive matches using recurrent neural networks". *Proceedings of SBGames 2018*. SBC, 2018.

[44] R. M. Silva. "Sports analytics". Ph.D. dissertation. Simon Fraser University, 2016.

[45] D. Silver et al. "Mastering chess and shogi by self-play with a general reinforcement learning algorithm". *arXiv preprint* arXiv:1712.01815 (2017).

[46] D. Silver et al. "Mastering the game of Go with deep neural networks and tree search". *Nature* 529.7587 (2016), pp. 484–489.

[47] D. Silver et al. “Mastering the game of Go without human knowledge”. *Nature* 550.7676 (2017), pp. 354–359.

[48] H. Stekler, D. Sendor, and R. Verlander. “Issues in sports forecasting”. *International Journal of Forecasting* 26.3 (2010), pp. 606–621.

[49] H. S. Stern. “A Brownian motion model for the progress of sports scores”. *Journal of the American Statistical Association* 89.427 (1994), pp. 1128–1134.

[50] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[51] E. T. S. Tan, K. Rogers, L. E. Nacke, A. Drachen, and A. Wade. “Communication sequences indicate team cohesion: A mixed-methods study of ad hoc League of Legends teams”. *Proceedings of the ACM on Human-Computer Interaction*. Vol. 6. ACM, 2022, article 225.

[52] D. Tang, R. K.-w. Sum, M. Li, R. Ma, P. Chung, and R. W.-k. Ho. “What is esports? A systematic scoping review and concept analysis of esports”. *Heliyon* 9.12 (2023), article e23248.

[53] T. M. Tango, M. G. Lichtman, and A. E. Dolphin. *The Book: Playing the Percentages in Baseball*. Potomac Books, 2007.

[54] G. Tesauro. “TD-Gammon, a self-teaching backgammon program, achieves master-level play”. *Neural Computation* 6.2 (1994), pp. 215–219.

[55] Valve Developer Community. *Dota Bot Scripting*. 2023. URL: https://developer.valvesoftware.com/wiki/Dota_Bot_Scripting (visited on 06/29/2024).

[56] J. A. C. Vera, J. M. A. Terrón, and S. G. García. “Following the trail of esports: The multidisciplinary boom of research on the competitive practice of video games”. *International Journal of Gaming and Computer-Mediated Simulations* 10.4 (2018), pp. 42–61.

[57] O. Vinyals et al. “Grandmaster level in StarCraft II using multi-agent reinforcement learning”. *Nature* 575.7782 (2019), pp. 350–354.

[58] M. Walker and J. Wooders. “Minimax play at Wimbledon”. *American Economic Review* 91.5 (2001), pp. 1521–1538.

[59] A. Washburn. *Two-Person Zero-Sum Games*. Springer US, 2014.

[60] A. White and D. M. Romano. “Scalable psychological momentum forecasting in esports”. *arXiv preprint* arXiv:2001.11274 (2020).

[61] P. R. Wurman et al. “Outracing champion Gran Turismo drivers with deep reinforcement learning”. *Nature* 602.7896 (2022), pp. 223–228.