Analyzing the Differences between Professional and Amateur Esports through Win Probability

Peter Xenopoulos
xenopoulos@nyu.edu
New York University
New York, NY, United States

William Robert Freeman
Bill@PureSkill.gg
PureSkill.gg
Charlotte, NC, United States

Claudio Silva
csilva@nyu.edu
New York University
New York, NY, United States

ABSTRACT

Estimating a team's win probability at any given point of a game is a common task for any sport, including esports, and is important for valuing player actions, assessing profitable bets, and engaging fans with interesting metrics. Past studies of win probability in esports have relied on data extracted from matches held in well-structured and organized professional tournaments. In these tournaments, players play on set teams, oftentimes where players are well acquainted with all participants. However, there has been little study of win probability modeling in casual gaming environments – those where players are randomly matched – even though these environments form the bulk of gaming hours played. Using interpretable win probability models trained on large CSGO data sets, we improve upon the current state of the art in Counter-Strike: Global Offensive (CSGO) win probability prediction. We identify important features, such as team HP and equipment value, across different skill levels. We also find a small benefit to using Elo-based player skill estimates in predicting win probability. Furthermore, we discuss how our win probability models can be used to investigate the problem of player-leaving in competitive matchmaking.

CCS CONCEPTS

• **Applied computing** → **Computer games**; • **Information systems** → *Data mining*.

KEYWORDS

esports, online gaming, win probability, first person shooter

ACM Reference Format:
Peter Xenopoulos, William Robert Freeman, and Claudio Silva. 2022. Analyzing the Differences between Professional and Amateur Esports through Win Probability. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512277

1 INTRODUCTION

Esports, or competitive video gaming, is not only rapidly growing but is also being embraced by traditional sports organizations due to esports' high viewership numbers. In fact, some esports competitions draw more viewers than conventional sporting events [4]. Additionally, there is significant academic and analytical interest in esports [33]. Concerning esports analytics, one of the primary objectives is developing accurate win probability models. Estimating win probability in esports is crucial for a variety of stakeholders and applications, such as teams evaluating players, sports bettors assessing their bets, or media organizations creating statistics-informed broadcasts. For example, there is a growing industry that aims to analyze a player's in-game actions and provide suggestions for improvement. Likewise, media broadcasts, along with video games themselves, are utilizing win probability to highlight important moments of a match. Thus, developing and understanding win probability models under different conditions is a fundamental task in esports analytics. In conventional sports, such as basketball, soccer and ice hockey, win probability is well-researched. However, due to a lack of data, there has been a dearth of analytical work concerning win probability in esports, particularly for first-person shooter (FPS) games, such as Counter-Strike: Global Offensive (CSGO) or Valorant.

Although win probability is useful to estimate, due to a general lack of accessible esports data, win probability models have only been trained on data coming from organized, top-tier competitive play. However, we know that in other sports, win probability models can differ for different playing environments. For example, Bransen and Davis [2] found differences between men and women's soccer expected goals models in terms of what features were important and how the various models valued different scenarios. In games with a strong esports following, casual gaming environments still form the majority of playing hours. However, it is unclear how current win probability models, trained on professional player data, extend to lower skill environments. Furthermore, there has been little investigation towards interpreting esports win probability models and comparing feature effects across skill levels.

We perform a detailed study of the differences between professional, semi-professional, and casual gaming environments in CSGO, a popular first person shooter game that is currently the most played game on Steam, a popular gaming platform. We assemble a data set of thousands of CSGO matches spanning professional tournaments, third-party matchmaking for high-skill players, and amateur matchmaking. Then, we train interpretable models for each skill environment and interpret the models to explore feature effects across skill levels. Furthermore, we explore the effect of player skill priors on win probability. We also use our trained win probability to provide suggestions to the problem of player-leaving in matchmaking (whereby a player permanently leaves a match). We find that while many features effects are common determinants

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512277

of winning across skill levels, some features have heterogenous effects among skill levels. Concerning player skill priors, we find small effects on win probability, and the effect is more pronounced for models trained on amateur data. Finally, we find evidence that the current solution to the player-leaving problem (i.e., a player permanently leaves a competitive game) in matchmaking has little effect on equalizing win probability between teams, and we provide suggestions to improve player experience.

2 RELATED WORK

Win probability and outcome prediction is an established line of work in conventional sports, particularly as it pertains to player evaluation. For example, Sicilia et al. [28] propose DeepHoops, an end-to-end deep learning architecture that ingests basketball tracking data and predicts the expected points to be scored in a possession. In ice hockey, Liu and Schulte [15] apply reinforcement learning techniques to value player actions based on how an action changes the expected return of a team. For soccer, Power et al. [26] value passes by how a pass changes a team's chance of shooting the ball. Additionally, Decroos et al. [3] introduce a framework to value soccer players through a model which measures their team's chance of scoring. The aforementioned works rely on models that either predict a team's chance of scoring or of winning – two very similar tasks. Bransen and Davis [2] investigate the differences between xG models trained on men and women's soccer event data. They find that although, on average, the xG models performed the same across the two environments, there were differences in feature importances, as well as several in-game situations with expected goal differences.

There are also similar efforts to estimate win probability in esports. One of the initial efforts is *TrueSkill* [7, 22], a Bayesian skill rating system that infers player skill from team results. These player skill estimates influence pre-match win probabilities, but do not provide live, in-game win probabilities. Much of the later work in esports has revolved around massive online battle arena (MOBA) games. Yang et al. [38] predict both pre-match and live win probability for high-level Defense of the Ancients 2 (Dota 2) matches using in-game player features, as well as pre-match player rankings. They find that real-time features were especially important for predicting live win probability. Hodge et al. [8] also train a win probability model for Dota 2, however, they consider professional Dota 2 matches rather than casual matches. Lee et al. [13] present a win probability model based on player log data from another similarly popular MOBA, League of Legends (LoL), and use casual games from top-level players. Kim et al. [11] propose a confidence calibration method for a win probability models in LoL. Recently, Yang et al. [39] describe a spatio-temporal neural network to predict win probability in Honor of Kings, another MOBA game, using matches from high-level players. Win probability can be useful for a variety of tasks in esports, such as player valuation or highlight detection. For example, Kang and Lee [9] use win probability models to find moments in League of Legends when there are large win probability changes. They use these changes to identify and generate highlights.

For first person shooter (FPS) games, there have been some, albeit much fewer, works directed towards win probability modeling. Makarov et al. [20] use game replays from professional games to model after-plant scenarios in CSGO. Xenopoulos et al. [36] propose a CSGO player valuation framework based on how a player's action changes their team's chance of winning a round. Additionally, Xenopoulos et al. [35] use professional CSGO replays to create a game-level win probability model based on team scores and in-game economic decisions. While the aforementioned works in MOBA and FPS games have focused on professional or high-level casual gaming, it is unknown how win probability models trained in high-skill environments translate into lower-skill environments.

Previous research has shown that there are differences in video gaming performance between various skill levels. For example, Khromov et al. [10] observe that high-skill CSGO players focus their gaze in the center of the screen, whereas low-skill players have a much higher gaze variance. Koposov et al. [12] find that professional players have, on average, faster reactions times and respond to visual stimuli faster than non-professionals. There are also other factors that may cause differences in performance. For example, in professional matches, players mostly disengage from any streaming platforms, such as Twitch. However, streaming is often found in casual gaming environments. Matsui et al. [21] find that streaming decreases a player's in-game performance. Likewise, casual environments typically have a higher latency than professional gaming environments. Liu et al. [16] find that for CSGO, lower latency results in better in-game performance, on average. Additionally, in casual environments where players are assigned randomly to teams, it is possible to have malicious, known as "toxic", players on your team. These players can frustrate a player, resulting in performance lapses [34]. Smerdov et al. [30] find that stress and concentration levels for professional players are less correlated with playing performance. They also find that the absence of team communication does not affect professional players as much as amateur ones. Additionally, Smerdov et al. [29] find that professional CSGO athletes can be identified by their behavior on their gaming chairs. We build upon previous win probability and esports literature to quantify the differences between professional and casual gaming environments through analyzing win probability models trained on data from various skill levels.

3 COUNTER-STRIKE

3.1 Game Mechanics

First released in 2000, Counter-Strike is a video game series with a long history. Since its initial release, Counter-Strike's fundamental game mechanics have been mostly stable. In CSGO, which is the latest version of Counter-Strike, two teams of five battle to complete a variety of objectives on a *map*, which is a distinct virtual world. Typically, professional matches are structured as either a best of one or best of three maps. A team has to win the match by winning a certain number of *rounds*, where one team plays as the attacking side and the other as the defending side. The round win conditions depend on what side a team plays. If a team plays as the "CT" side (defenders), they can win a round by preventing the other team from planting a bomb at one of the bombsites, or by defusing a planted bomb. The side that attempts to plant and explode the bomb is denoted the "T" side (attackers). Both sides can win a round by eliminating the other side's opponents. Players earn in-game money

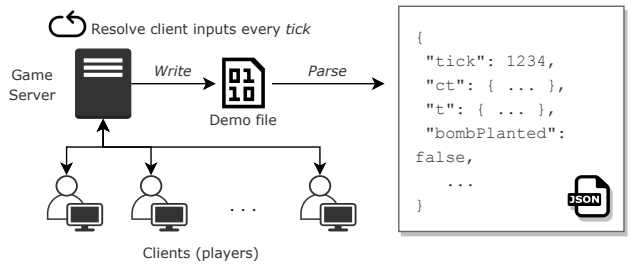

Figure 1: Players' game clients send updates to the game server, which reconciles inputs and returns a unified global game state back to the clients. The game server records the updates to the global game state in the *demo file*.

as they complete objectives, such as planting or defusing the bomb, or by eliminating members of the opposing side. At the beginning of each round, players can use their earned in-game money to buy equipment, such as guns, grenades and armor. The starting side for each team is determined before the match begins, either randomly or is agreed upon beforehand. The two teams swap sides at halftime, which is after the 15th round. A team needs to win 16 rounds to win a map.

3.2 Demo Files

CSGO game servers reconcile the client (player) inputs and sends back a unified (ground-truth) game state to each client. The updates to the clients occur at a defined *tick* rate. For professional matches, the tick rate is typically 128, which means the game server updates clients 128 times a second. On the game server, it is common practice to record the events of a match using *demo files*. Demo files, also called demos, contain the updates from the game server to the clients that produce the ground-truth global game state. Effectively, one can reconstruct the match from the demo file. Demo files are used for verifying match results in tournaments, analyzing past matches, or for identifying cheating players. Accordingly, demo files are ubiquitous at all levels of CSGO play, and are easy to obtain from most matchmaking services. We show our demo file parsing pipeline in Figure 1. We provide more implementation details on our demo file parsing procedure in Appendix A.

3.3 Matchmaking

To play a CSGO match, players and teams undergo a *matchmaking* process which connects players looking to play. CSGO matches are organized across a wide array of tournament providers, third-party matchmaking services, and the official matchmaking service through the game itself. We consider CSGO demos from three main sources which cover a broad range of gameplay levels: (1) HLTV, (2) FACEIT, a popular third-party matchmaking service, and (3) Official Matchmaking (MM). HLTV is a popular CSGO fan site that hosts demos from top-tier professional competitions. Matches covered by HLTV are from structured tournaments, usually with brackets or schedules, and involve well-established teams playing for cash prizes. FACEIT and Official Matchmaking are public matchmaking services, where players mostly play on informal, or "pick-up", teams. In this form of matchmaking, players, or groups of players, join a queue, which arranges games for randomly chosen teams of five players each. Many professional and high-level players play in invite-only pick-up leagues in FACEIT, referred to as FPL and FPL Challenger, where they play with other skilled players. MM is the matchmaking system run by the CSGO game developers. In this matchmaking system, competing teams are chosen in a way such that the two teams are close to equally ranked.

For FACEIT, we only collect demos from the FPL Circuit, which reflects a player base with professional players. A large proportion of FPL players also play in matches available on HLTV. To collect MM demos, we use demos collected by PureSkill.gg, a private subscription service through which players receive automated coaching and feedback generated from their demos. In order for a demo to be collected by PureSkill.gg, at least one of the players in the game needs to be registered on the site. We consider demo files from June 1, 2021 to September 1, 2021 for both HLTV, FACEIT and MM. We choose this time period as there were no updates to competitive maps nor weapons. To maintain computational feasibility, we use a random subset of demo files from PureSkill.gg, and the complete set from HLTV and FACEIT. We do not consider demo files that contained errors in the parsing process. In total, we use 3,551 demo files from HLTV, 3,492 from FACEIT, and 8,199 from PureSkill.gg.

4 MODELING WIN PROBABILITY

4.1 Task Description

A common task, not only in conventional sports but also esports, is predicting who will win. Usually, this task concerns the outcome of either a subset of a match, such as a round or fighting engagement, or the entire match itself. For our purposes, we are concerned with predicting *round* win probability in CSGO. Round win probability is a fundamental quantity used in player valuation and game broadcasts [36]. We denote the outcome of round r in match m by $Y_{m,r}$, which is equal to 1 if the CT side wins, and 0 if the T side wins. We want to estimate the win probability at each timestamp t in round r, which we denote by $\widehat{Y}_{m,r,t}$. The goal is to estimate a function $f(\boldsymbol{x}_{m,r,t})$ to predict the win probability, where $\boldsymbol{x}_{m,r,t}$ is a *game state vector*, which describes the game in match m at round r at time t.

A game state can contain a variety of features. For example, Xenopoulos et al. [36] use aggregate team equipment value, health, and bomb plant information to predict CSGO round win probability. Likewise, in Makarov et al. [20], they use the average equipment value, number of healthy players, smoke cover on the bomb, and total grenades remaining. To represent $\boldsymbol{x}_{m,r,t}$, we consider not only the global game state, such as the time remaining in the round, the map, or if the bomb is planted, but also side-specific information, such as aggregate team equipment values, health points (hp), armor, and grenades remaining. Specifically, we use the side-specific sums of equipment values, players remaining, HP, armor, helmets, grenades, and bomb defusal kits. For grenades, we take the sum of the remaining amount of flash, smoke, explosive, and incendiary grenades for each side. These features closely resemble those in previous literature, and are easy to obtain from demo files [36].

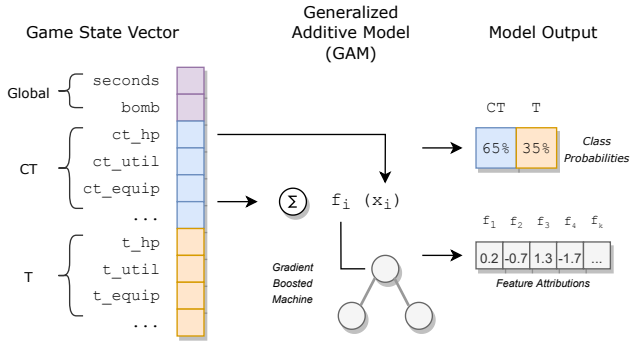

Figure 2: Game states, comprising of global and side-specific information, are passed through a generalized additive model, which estimates feature attributions.

4.2 Model Design

Past works on win probability prediction have used a variety of models such as logistic regression [20, 35, 38], tree-based methods [3, 36], and neural networks [37, 39, 40]. Due to strong performance on previous win probability tasks in other sports, as well as a desire for native interpretability characteristics, we use explainable boosting machines [24]. Explainable boosting machines (EBMs) are generalized additive models of the form

$$g(E[y]) = \beta_0 + \sum f_j(x_j) + \sum f_{ij}(x_i, x_j) \tag{1}$$

where g is the link function, f_j is the feature function for feature j and x_j is the j-th feature of an arbitrary input. f_j, also called the *shape function* is a learned univariate function for each feature, and bivariate function for feature pair f_{ij}. We use the implementation from [24], where f_j is a gradient boosted tree and g is a logistic function. The implementation contains a computationally efficient method to rank feature pairs for inclusion in the model. We provide more details on EBMs in Appendix B.

We can think of the output of each shape function as the attribution of a particular feature towards a prediction. Thus, for a given example input, the prediction is the sum of the feature scores. One benefit of EBMs is that in addition to class probability predictions for an input, we also calculate the feature attribution of each feature in an input. We can then use this information to investigate both global and local explainability characteristics. Furthermore, we can plot the x_i's against the f_i's to observe how the attribution of a feature changes over its domain. In addition to analyzing the scores of each feature, we can easily apply standard interpretability methods to EBMs, like LIME [27] or SHAP [19].

We train three models, each using one of HLTV, FACEIT, or MM data. To train each model, we split our data using 80% of demos for training and 20% for testing. For validation and early stopping, we use 15% of the train set. Typically, win probability models are evaluated using log loss, defined as

$$LL = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\widehat{y_i}) + (1 - y_i)\log(1 - \widehat{y_i}) \tag{2}$$

where y_i is either 0 or 1 (representing one of two classes), and $\widehat{y_i}$ is the probabilistic prediction from a binary classifier.

While we evaluate our models using log loss, it is also important to assess how well *calibrated* our models are. Well-calibrated models will have predicted probabilities that align well with the distribution of outcomes. Previous studies have shown that modern machine learning models can often produce uncalibrated predictions [6, 23]. To measure model calibration quantitatively, we use expected calibration error (ECE), defined as

$$ECE = \sum_{b=1}^{B} \frac{n^b}{N} |acc(b) - conf(b)| \tag{3}$$

where B is the number of bins, n^b is the total number of samples in bin b, N is the entire evaluation set, acc is the accuracy (number of correct predictions in the bin), and $conf$ is the confidence of the bin (mean of predicted probabilities in the bin). We set $B = 10$ such that all n^b are equal (equal sized bins).

5 RESULTS

5.1 Evaluating Win Probability Models

We present the log loss and ECE for each model in Table 1. We compare our trained models to the current state-of-the-art in CSGO win probability prediction from [36], which attains a log loss of 0.535. We find that our models not only exhibit superior performance to the current state-of-the-art, showing about a 15% decrease in log loss compared to previous models, but are also well-calibrated in their respective CSGO gaming environments. Interestingly, we observe slightly worse performance in predicting the outcomes of MM game states than in HLTV or FACEIT outcomes. One explanation could be that, since MM matches typically do not contain structured teams or high-level players like HLTV and FACEIT, there may be a higher degree of randomness due to a wider variety of skill level matchups and strategies.

Oftentimes in esports win probability prediction, researchers and practitioners only have access to a specific skill-level's data, such as games from amateur matchmaking or games from organized tournaments with professionals. Thus, it is important to highlight how a model trained in one environment performs in another. Doing so has implications for applications like player valuation [36], where win probability models are fundamental. In Table 1, we show a matrix of each model's test set performance in each environment. While we find that HLTV and FACEIT models are generally applicable to each other, we observe that the MM model had both higher log loss in HLTV and FACEIT environments, and was also less calibrated on the HLTV and FACEIT data. Thus, it is unclear if a model trained in an amateur environment can extend to a professional environment without a performance loss.

5.2 Feature Analysis

One benefit of using EBMs as our model architecture is that we can explore each feature's effect on prediction via the feature contribution terms (f_i's). We can calculate the global attribution of a feature i by averaging the value of f_i across all samples in the training set. In Figure 3, we show the global feature importance for each feature across our three models. Generally, we see that a

Model	Test: *HLTV*	Test: *FACEIT*	Test: *MM*
HLTV	0.449 (0.009)	0.443 (0.012)	0.467 (0.023)
FACEIT	0.451 (0.0101)	0.441 (0.008)	0.464 (0.020)
MM	0.470 (0.033)	0.461 (0.028)	0.456 (0.004)

Table 1: Log loss and ECE (in parenthesis) performance by model (trained using one of HLTV, FACEIT or MM data) on test set (one of HLTV, FACEIT or MM).

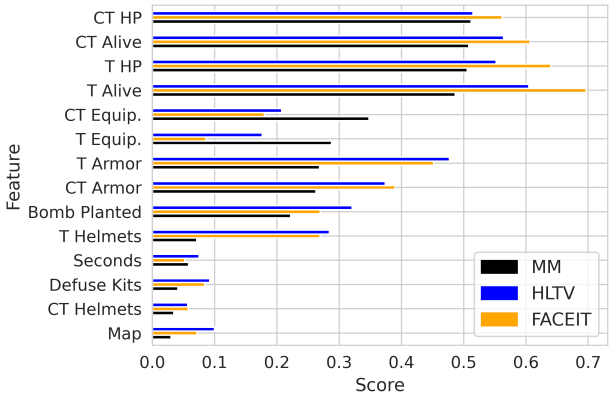

Figure 3: Global feature scores by train data set. Players remaining, HP and equipment are the most important team features across all models.

side's HP, armor, equipment, and players alive contribute the most to our predictions. This finding is consistent with past work [36], as well as conventional game knowledge.

Since f_i is calculated through a univariate function taking x_i as input, we can plot the value of f_i against the domain of x_i. Effectively, each f_i is "modular", and contributes independently to the final prediction. The reported feature scores are in log odds. With these plots, we can investigate the effect of a feature at different levels of that feature. In Figure 4 we show the feature contributions for the CT side equipment value, total hp and total armor, which consistently attain high global feature contributions across both professional and amateur data. We generally observe monotonically increasing benefits of these features for CT win probability. Likewise, for the T-sided features, we see a similar benefit for T win probability.

CSGO game states can be divided into two scenarios: (1) states where the is bomb planted, and (2) states where the bomb not planted. A CSGO round starts with a one minute and fifty-five second timer and no bomb planted. If the time runs out and the bomb has *not* been planted, then the CT side wins. If the bomb is planted, the bomb plant phase begins, and the game clock begins to count down from forty seconds. Defusing the bomb takes ten seconds without a defuse kit, and five seconds with a defuse kit.

Map	HLTV	FACEIT	MM
de_mirage	18% (0.01)	35% (0.03)	41% (0.00)
de_dust2	17% (-0.08)	20% (-0.06)	21% (-0.05)
de_inferno	19% (-0.18)	17% (-0.13)	20% (-0.02)
de_nuke	20% (0.10)	13% (0.11)	8% (0.06)
de_overpass	14% (0.15)	7% (0.06)	7% (0.10)
de_vertigo	12% (0.07)	8% (0.08)	4% (0.09)

Table 2: Map selection rates by CSGO matchmaking system. The feature score for each map is shown underneath the selection rate – positive indicates a map that is beneficial for the CT side.

In Figure 5, we plot the feature contribution for two models – one trained on game states where the bomb is planted and the other trained on game states where the bomb was not planted. We see strong feature contributions when time is about to expire for both phases. The directions of the effects are intuitive as 40 seconds after the bomb is planted results in a T win if the bomb has not been defused. Likewise, if 115 seconds pass, the bomb has not been planted, and players from each side are still alive, the CT side wins the round.

The map on which a CSGO match is played is also a key component of any match. Previous work has shown that professional teams can be inefficient with their map selections [25]. In HLTV and FACEIT, the map is chosen ahead of time through a draft system, but for MM, players select which maps they want to play, and the matchmaking system creates matches accordingly. Generally, map selections vary drastically across competitive environments. We show the map selection rate by gaming environment in Table 2, along with the estimated map feature attribution for each of the three models. We see that map selections are much more uniform in top-tier professional matches compared to MM, where roughly 60% of MM matches take place on de_mirage or de_dust2. We see that while the map compositions highly favor de_mirage, de_dust2, and de_inferno in casual environments, maps like de_nuke, de_overpass, and de_vertigo are more common at more skilled levels. Furthermore, these maps also exhibit clear advantages for the CT side, as their feature attributions are positive.

6 DISCUSSION

6.1 Interpreting Win Probability Models

While some of the results identified in Section 5 are intuitive, there exist many interesting relationships which require deeper discussion provided in this section. We organize this discussion into broadly related feature groups: (1) equipment, hp and armor, (2) time and bomb plants, (3) maps.

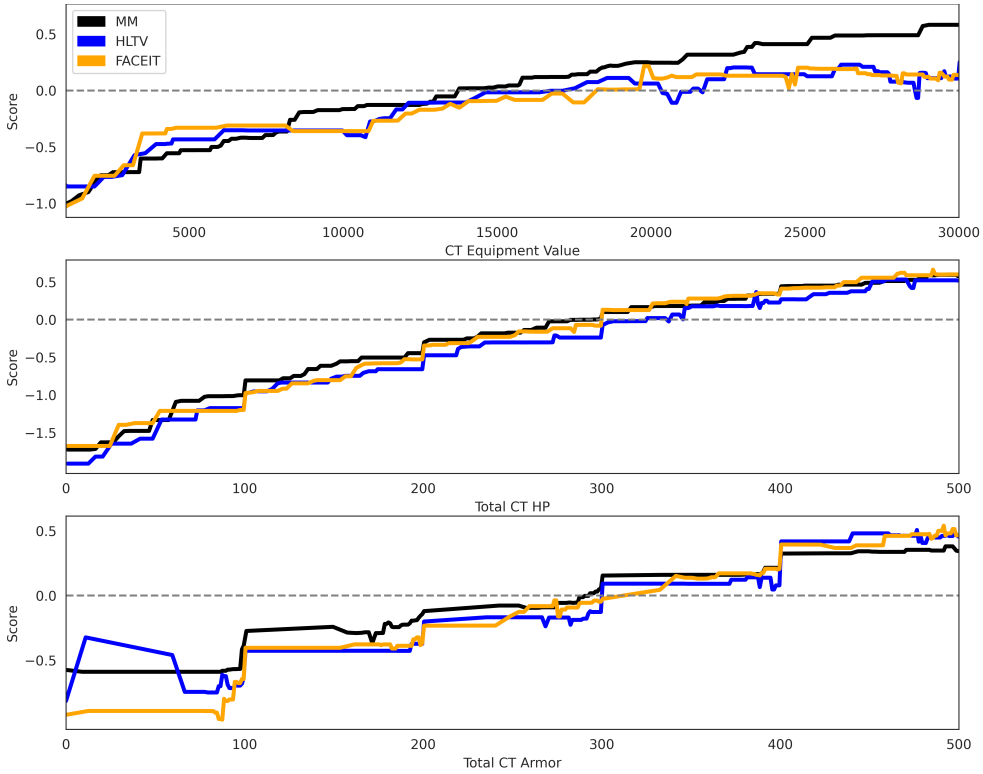

Figure 4: Feature contributions for the CT side equipment value, total HP, and total armor. Line color indicates HLTV (blue), FACEIT (orange) and MM (black).

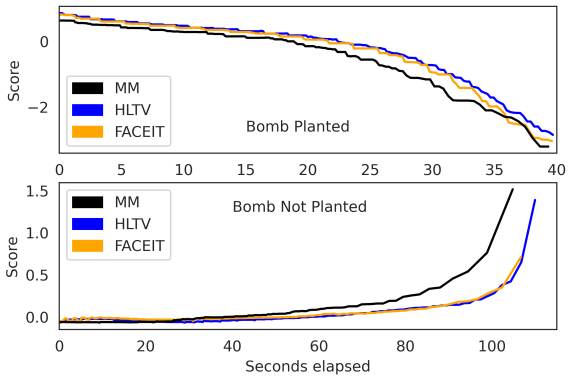

Figure 5: The feature score of seconds elapsed is higher for MM games in both bomb planted and bomb not planted game states.

6.1.1 Equipment Value, HP and Armor. Team equipment values and HP have consistently been shown to be important, positive predictors of CSGO win probability [36]. Concerning equipment value, we see that its global feature contribution is much higher in MM (0.35 CT, 0.29 T) than in HLTV (0.21 CT, 0.18 T) or FACEIT (0.18 CT, 0.09 T). The equipment value of a team effectively decides its "buy type", which is an ordinal feature describing a team's equipment investment. Xenopoulos et al. [35] find that there exists substantial difference in round win rates depending on a side's buy type. Our finding suggests equipment's effect on win probability may be lower for professional than amateur games.

Armor is an item which reduces the damage a player takes (except for shots to the head) and is purchasable at the beginning of each round. One unique aspect of the plot for armor is that between large ranges, such as 100, 200, 300, and so on, the feature score is relatively constant. This suggests that even though a player may take a large amount of armor damage from an engagement, their team's win probability is still relatively intact. However, since there are large changes in feature contributions at certain levels, this suggests that armor is still an important investment in CSGO.

Helmets are another form of armor which offsets the damage dealt by shots to a player's head, which typically result in quick kills for many guns. Interestingly, we see that in MM, helmets, particularly for T-side, have a much lower feature score (0.07) compared to HLTV (0.28) or FACEIT (0.27). One possible explanation may be that since the MM player skill level is more varied, and on average lower, than FACEIT or HLTV, headshots may be less common. Thus, in such a scenario, helmets would carry less importance.

6.1.2 Time Remaining and Bomb Plants. While we found intuitive relationships between time remaining and win probability, Figure 5

shows slight differences between the feature contributions for seconds elapsed in MM compared to HLTV or FACEIT in both bomb planted and bomb not planted scenarios. Specifically, the magnitude of feature score for time remaining appeared to be larger in MM compared to HLTV or FACEIT, especially as more time elapsed in situations where the bomb was not planted. There are a few possible explanations for this strong positive effect. For example, in the ends of rounds, if a player has equipment that they do not want to lose, they may elect to hide from the enemy until the round is over (colloquially known as "saving"). Usually, players only save at the ends of rounds, as maps are small enough that hiding for the length of a standard CSGO round is difficult. MM game states which occur late in rounds may be biased towards save strategies than those in HLTV or FACEIT. An alternative explanation is be that players in MM are generally less coordinated as a team than those playing in FACEIT or HLTV, and thus may run out of time more often. While matches in professional environments will almost always use voice communication between players, it is not uncommon for players in MM matches to lack substantial voice communication.

6.1.3 Maps. As shown in Section 5.2, certain maps, like Mirage or Dust2, are exceedingly popular in matchmaking versus professional environments. This divergence in map preference may be explained by a variety of factors. These two maps generally have low average effects on win probability when averaged across the three match environments (0.01 and -0.06, respectively), and the lowest effets are in MM. This relationship may indicate a preference of the amateur CSGO community towards maps with even-sided round outcomes. Interestingly, for some maps, we see large discrepancies in map feature score between skill environments. For example, the magnitude of the feature contribution for Inferno is 9× higher in the HLTV model (-0.18) than the MM model (-0.02). This may indicate that there are certain features of Inferno that are being taken advantage of by professional teams. One possible explanation is the use of grenades. In general, players buy and use more grenades in professional games, almost at a rate of 2× more compared to MM. These grenades can be used to obscure the vision of opponents, lower their HP or clear areas of a map. Thus, geometric properties of maps like Inferno may make grenade usage more impactful than on other maps. These geometric properties of maps may impact player preference by more than their effect on grenade usage. For example, maps like Vertigo and Nuke are very vertically designed when compared to more "flat" maps like Dust2 or Mirage.

6.2 Using Player Ranks as Skill Priors

Most past win probability modeling work, in both conventional sports and esports, does not consider priors on individual player skill. However, in many matchmaking systems, it is common to track player "skill" levels. In MM and FACEIT, there exist player skill estimates, which we use to train our models for those environments. For MM, player skill is represented as an ordinal feature called a player's "rank", of which there are about 20 levels. For FACEIT, player skill is represented using an Elo-based rating system [5], which is a continuous feature. When players win matches, their rating increases, and when they lose matches, their rating decreases. The size of the increase or decrease is determined by the opposing team's rank. If the opposing team's rank is higher, then the winning

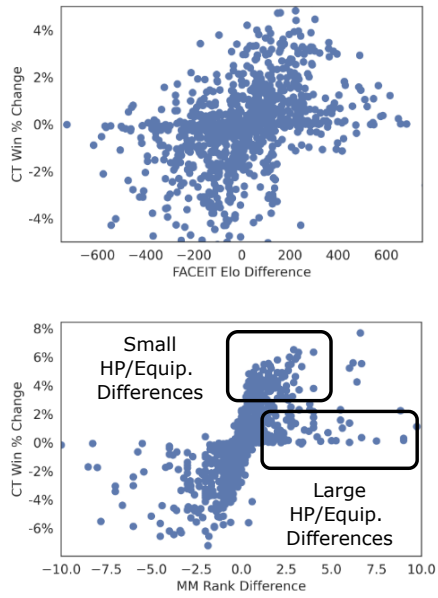

Figure 6: Rank advantages over the opposing players have, on average, a positive effect on a team's win probability in both FACEIT ($r = 0.13$) and MM ($r = 0.59$).

team will gain Elo, and vice-versa. We test the effect of using these player skill metrics as features in our model, including the average rank or Elo of the remaining players on each side for each game state, as well as the difference between the average CT or T ranks.

We calculate the difference in predicted win probability of game states using a model which considers player rank features and a model which lacks player rank features. We plot the relationship between the difference in player skill and the difference in predicted win probability between the two models in Figure 6. While there is a general trend such that a larger skill difference in the CT's favor is correlated with a higher predicted CT win probability, the relationship is much smaller and weaker in our FACEIT model ($r = 0.13$) than our MM model ($r = 0.59$). One explanation for this discrepancy is that there may be less true skill variation in the FACEIT data – the FACEIT league that we consider is invite-only and consists mostly of professional players. Thus, the player pool represents the tail-end of the distribution of player skills.

In Figure 6, we observe that in cases where there is a large difference in ranks, but low CT win probability difference between the two models, there are features which have large differences in feature attribution values between the two sides. For example, in game states where one team has near full HP and the best equipment, their attribution from the side's HP and equipment values will be high. Thus, disparities in player ranks may not overcome the attribution from other features. Conversely, in game states when there is a large difference in the CT win probability when accounting for player ranks, these game states typically have smaller equipment and HP differences, meaning that the feature attribution from the player rank features effects the prediction more.

6.3 Estimating the Effects of Player Abandonment in Matchmaking

Since players are randomly grouped into teams for matchmaking, it is possible for a player to leave (abandon) the match. In such an instance, the team with the abandoned player plays the rest of the match down a player (e.g., 4 vs. 5). The causes for leaving are varied, including "tilt" (anger) towards teammates [34], computer crashes, internet outages, or events in real-life that necessitate a player leaving the game. Such imbalanced in-game scenarios can deteriorate the gaming experience for players in the team with fewer players. As seen in section 5, the number of players remaining for a side has a strong effect on win probability.

To remedy the effect of players leaving, matchmaking services use a variety of interventions. For example, if a player fails to connect within a certain time frame of the match starting, the match is ended preemptively. In FACEIT, a team will receive a bot (computer-controlled) player in place of the player that abandoned. In MM, if a player leaves *during* the match, the players on the abandoning player's team will receive $1,000 of in-game money at the start of every future round. However, it is unclear how effective the additional money is in evening the win probabilities of both teams. In this section, we conduct a detailed study of the current intervention in MM for an abandoned player and offer suggestions to alleviate the effect of an abandoned player.

We assess the impact of the extra money intervention in MM by considering three scenarios for game states with 4 or fewer CT players: (1) game state with unaltered features, (2) game state perturbed by adding $1,000 to CT equipment value per player that is alive, (3) game state perturbed by adding a full HP and armor player with the average CT equipment value per player. We see that using the extra $1,000 per player intervention, the average CT win probability gain is only 1.7%, compared to 17.3% with an extra player. The effect of an extra player even exists if, instead of a full-strength player, we add a player with half HP and armor, the average CT win probability gain is 13.5%. For a player with 1 HP, no armor, and the default weapon, CT win probability is still, on average, 6% higher. Due to the strong effect that an extra player has on win probability, along with the diminishing returns of equipment value, interventions based on providing extra money to players affected by a teammate abandoning the game have only small effects on their team's win probability.

One potential fix to the player abandonment problem is to return to the use of "bots", which are computer-controlled players. Before the intervention of an extra $1,000 per round, when a player abandoned a match, their side would receive a bot in their place. Then, when another player died in a round, that player would have the opportunity to take over the bot. While bots generally have very low skill, the ability for a human player to take control of the bot is essentially extending a team's current game state win chance.

6.4 Limitations and Future Work

One limitation of our work is that we only consider data from CSGO. We primarily focus on CSGO due to data limitations in other games, since, in CSGO, high-fidelity data is readily available, unlike other popular FPS games such as Valorant or Overwatch. However, despite our focus on CSGO, our approach is easily applicable to other FPS games, since most FPS esports games have similar mechanics to CSGO. Thus, a useful direction for esports research would be to improve data acquisition technologies to attain both player position and action data to enable win probability modeling. One possible direction is the use of computer vision. For example, although not intended for analyzing player performance, Tanaka and Simo-Serra present a collection of League of Legends videos with captions intended for training video description predictors [31].

Another limitation of the paper is that our modeling approach, though interpretable, may be outperformed by other methods. For example, an untuned XGBoost model attains a test set loss of 0.455 on the HLTV test set. Additionally, we could consider other features, such as how much "momentum" a team may have [14]. We can also consider non-vector based approaches to predicting win probability. For example, in American football, Yurko et al. [40] use sequences of game states to predict yards gained on a play. Future work will be directed towards alternative representations of games, such as through graphs or sequences. Finally, our work identifies interesting relationships arising from the win probability models. Much of these directions warrant further study individually, such as the effect of map geometry on different skill groups, or how voice communication affects a team's performance at different skill levels.

7 CONCLUSION

In this paper, we examine the differences between professional, high-skill, and amateur environments in Counter-Strike: Global Offensive, a popular multiplayer first-person shooter game with millions of players. Contrary to prior work, which focused on high-level competitive play, we compare win probability models across different skill groups. Specifically, we train interpretable win probability models on large data sets processed from amateur, semi-pro and professional CSGO game replays. We make the following contributions: (1) we attain superior performance over the current state-of-the-art in win probability prediction and show how certain features, like team equipment values or HP, have different effects in professional versus amateur gaming environments; (2) we explore the use of player skill priors on win probability and find situations where adding player skill as a feature results in prediction improvements, such as post-plant scenarios; (3) we use our win probability models to explore the problem of player abandonment in matchmaking which suggests that the current solution may be suboptimal, and we offer suggestions to improve player experience. While esports data is generally hard to acquire, our approach can easily be extended to other esports given the appropriate game replay data. Future work should be directed towards improved esports data acquisition, as well as exploring the effect of map geometry and voice communication on player performance at different skill levels.

ACKNOWLEDGMENTS

This work was partially supported by NSF awards CCF-1533564, CNS-1544753, CNS-1730396, CNS-1828576, and CNS-1229185. P. Xenopoulos and C. Silva were also funded by Capital One. C. Silva is partially supported by DARPA. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

REFERENCES

[1] David Bednárek, Martin Krulis, Jakub Yaghob, and Filip Zavoral. 2017. Data Preprocessing of eSport Game Records - Counter-Strike: Global Offensive. In *Proceedings of the 6th International Conference on Data Science, Technology and Applications, DATA 2017, Madrid, Spain, July 24-26, 2017*, Jorge Bernardino, Christoph Quix, and Joaquim Filipe (Eds.). SciTePress, 269–276. https://doi.org/10.5220/0006475002690276

[2] Lotte Bransen and Jesse Davis. 2019. Women's football analyzed: interpretable expected goals models for women. In *IJCAI 2021 Workshops*.

[3] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. 2019. Actions Speak Louder than Goals: Valuing Player Actions in Soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 1851–1861. https://doi.org/10.1145/3292500.3330758

[4] Michael M Goldman and David P Hedlund. 2020. Rebooting content: Broadcasting sport and esports to homes during COVID-19. *International Journal of Sport Communication* 13, 3 (2020), 370–380.

[5] Thore Graepel and Ralf Herbrich. 2006. Ranking and matchmaking. *Game Developer Magazine* 25 (2006), 34.

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 1321–1330. http://proceedings.mlr.press/v70/guo17a.html

[7] Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. TrueSkill^{TM}: A Bayesian Skill Rating System. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, Bernhard Schölkopf, John C. Platt, and Thomas Hofmann (Eds.). MIT Press, 569–576. https://proceedings.neurips.cc/paper/2006/hash/f44ee263952e65b3610b8ba51229d1f9-Abstract.html

[8] Victoria Hodge, Sam Devlin, Nick Sephton, Florian Block, Peter Cowling, and Anders Drachen. 2019. Win Prediction in Multi-Player Esports: Live Professional Match Prediction. *IEEE Transactions on Games* (2019), 1–1. https://doi.org/10.1109/TG.2019.2948469

[9] Seok-Kyu Kang and Jee-Hyong Lee. 2020. An E-sports video highlight generator using win-loss probability model. In *SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, online event, [Brno, Czech Republic], March 30 - April 3, 2020*, Chih-Cheng Hung, Tomás Cerný, Dongwan Shin, and Alessio Bechini (Eds.). ACM, 915–922. https://doi.org/10.1145/3341105.3373894

[10] Nikita Khromov, Alexander Korotin, Andrey Lange, Anton Stepanov, Evgeny Burnaev, and Andrey Somov. 2019. Esports Athletes and Players: A Comparative Study. *IEEE Pervasive Comput.* 18, 3 (2019), 31–39. https://doi.org/10.1109/MPRV.2019.2926247

[11] Dong-Hee Kim, Changwoo Lee, and Ki-Seok Chung. 2020. A Confidence-Calibrated MOBA Game Winner Predictor. In *IEEE Conference on Games, CoG 2020, Osaka, Japan, August 24-27, 2020*. IEEE, 622–625. https://doi.org/10.1109/CoG47356.2020.9231878

[12] Denis Koposov, Maria Semenova, Andrey Somov, Andrey Lange, Anton Stepanov, and Evgeny Burnaev. 2020. Analysis of the Reaction Time of eSports Players through the Gaze Tracking and Personality Trait. In *29th IEEE International Symposium on Industrial Electronics, ISIE 2020, Delft, The Netherlands, June 17-19, 2020*. IEEE, 1560–1565. https://doi.org/10.1109/ISIE45063.2020.9152422

[13] Sang-Kwang Lee, Seung-Jin Hong, and Seong-Il Yang. 2020. Predicting Game Outcome in Multiplayer Online Battle Arena Games. In *International Conference on Information and Communication Technology Convergence, ICTC 2020, Jeju Island, Korea (South), October 21-23, 2020*. IEEE, 1261–1263. https://doi.org/10.1109/ICTC49870.2020.9289254

[14] Quan Li, Peng Xu, Yeukyin Chan, Yun Wang, Zhipeng Wang, Huamin Qu, and Xiaojuan Ma. 2017. A Visual Analytics Approach for Understanding Reasons behind Snowballing and Comeback in MOBA Games. *IEEE Trans. Vis. Comput. Graph.* 23, 1 (2017), 211–220. https://doi.org/10.1109/TVCG.2016.2598415

[15] Guiliang Liu and Oliver Schulte. 2018. Deep Reinforcement Learning in Ice Hockey for Context-Aware Player Evaluation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 3442–3448. https://doi.org/10.24963/ijcai.2018/478

[16] Shengmei Liu, Mark Claypool, Atsuo Kuwahara, Jamie Sherman, and James J. Scovell. 2021. Lower is Better? The Effects of Local Latencies on Competitive First-Person Shooter Game Players. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn, and Steven Mark Drucker (Eds.). ACM, 326:1–326:12. https://doi.org/10.1145/3411764.3445245

[17] Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible models for classification and regression. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, Qiang Yang, Deepak Agarwal, and Jian Pei (Eds.). ACM, 150–158. https://doi.org/10.1145/2339530.2339556

[18] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. 2013. Accurate intelligible models with pairwise interactions. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy (Eds.). ACM, 623–631. https://doi.org/10.1145/2487575.2487579

[19] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4765–4774. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[20] Ilya Makarov, Dmitry Savostyanov, Boris Litvyakov, and Dmitry I. Ignatov. 2017. Predicting Winning Team and Probabilistic Ratings in "Dota 2" and "Counter-Strike: Global Offensive" Video Games. In *Analysis of Images, Social Networks and Texts - 6th International Conference, AIST 2017, Moscow, Russia, July 27-29, 2017, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 10716)*, Wil M. P. van der Aalst, Dmitry I. Ignatov, Michael Yu. Khachay, Sergei O. Kuznetsov, Victor S. Lempitsky, Irina A. Lomazova, Natalia V. Loukachevitch, Amedeo Napoli, Alexander Panchenko, Panos M. Pardalos, Andrey V. Savchenko, and Stanley Wasserman (Eds.). Springer, 183–196. https://doi.org/10.1007/978-3-319-73013-4_17

[21] Akira Matsui, Anna Sapienza, and Emilio Ferrara. 2020. Does streaming esports affect players' behavior and performance? *Games and Culture* 15, 1 (2020), 9–31.

[22] Tom Minka, Ryan Cleven, and Yordan Zaykov. 2018. TrueSkill 2: An improved Bayesian skill rating system. *Tech. Rep.* (2018).

[23] Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring Calibration in Deep Learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 38–41. http://openaccess.thecvf.com/content_CVPRW_2019/html/Uncertainty_and_Robustness_in_Deep_Visual_Learning/Nixon_Measuring_Calibration_in_Deep_Learning_CVPRW_2019_paper.html

[24] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. InterpretML: A Unified Framework for Machine Learning Interpretability. *arXiv preprint arXiv:1909.09223* (2019).

[25] Guido Petri, Michael Stanley, Alec Hon, Alexander Dong, Peter Xenopoulos, and Claudio Silva. 2021. Optimal Team Economic Decisions in Counter-Strike. In *IJCAI 2021 Workshops*.

[26] Paul Power, Héctor Ruiz, Xinyu Wei, and Patrick Lucey. 2017. Not All Passes Are Created Equal: Objectively Measuring the Risk and Reward of Passes in Soccer from Tracking Data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 1605–1613. https://doi.org/10.1145/3097983.3098051

[27] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-Agnostic Interpretability of Machine Learning. *CoRR* abs/1606.05386 (2016). arXiv:1606.05386 http://arxiv.org/abs/1606.05386

[28] Anthony Sicilia, Konstantinos Pelechrinis, and Kirk Goldsberry. 2019. DeepHoops: Evaluating Micro-Actions in Basketball Using Deep Feature Representations of Spatio-Temporal Data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 2096–2104. https://doi.org/10.1145/3292500.3330719

[29] Anton Smerdov, Anastasia Kiskun, Rostislav Shaniiazov, Andrey Somov, and Evgeny Burnaev. 2019. Understanding Cyber Athletes Behaviour Through a Smart Chair: CS: GO and Monolith Team Scenario. In *5th IEEE World Forum on Internet of Things, WF-IoT 2019, Limerick, Ireland, April 15-18, 2019*. IEEE, 973–978. https://doi.org/10.1109/WF-IoT.2019.8767295

[30] Anton Smerdov, Bo Zhou, Paul Lukowicz, and Andrey Somov. 2020. Collection and Validation of Psycophysiological Data from Professional and Amateur Players: a Multimodal eSports Dataset. *CoRR* abs/2011.00958 (2020). arXiv:2011.00958 https://arxiv.org/abs/2011.00958

[31] Tsunehiko Tanaka and Edgar Simo-Serra. 2021. LoL-V2T: Large-Scale Esports Video Description Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4557–4566.

[32] Kenton Varda. 2008. Protocol Buffers. http://code.google.com/apis/protocolbuffers/.

[33] Michael G Wagner. 2006. On the Scientific Relevance of eSports.. In *International conference on internet computing*. Las Vegas, NV, 437–442.

[34] Minerva Wu, Je Seok Lee, and Constance Steinkuehler. 2021. Understanding Tilt in Esports: A Study on Young League of Legends Players. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister,

Takeo Igarashi, Pernille Bjørn, and Steven Mark Drucker (Eds.). ACM, 321:1–321:9. https://doi.org/10.1145/3411764.3445143
[35] Peter Xenopoulos, Bruno Coelho, and Claudio Silva. 2021. Optimal Team Economic Decisions in Counter-Strike. In *IJCAI 2021 Workshops*.
[36] Peter Xenopoulos, Harish Doraiswamy, and Cláudio T. Silva. 2020. Valuing Player Actions in Counter-Strike: Global Offensive. In *IEEE International Conference on Big Data, Big Data 2020, Atlanta, GA, USA, December 10-13, 2020*, Xintao Wu, Chris Jermaine, Li Xiong, Xiaohua Hu, Olivera Kotevska, Siyuan Lu, Weija Xu, Srinivas Aluru, Chengxiang Zhai, Eyhab Al-Masri, Zhiyuan Chen, and Jeff Saltz (Eds.). IEEE, 1283–1292. https://doi.org/10.1109/BigData50022.2020.9378154
[37] Peter Xenopoulos and Cláudio T. Silva. 2021. Graph Neural Networks to Predict Sports Outcomes. In *IEEE International Conference on Big Data, Big Data 2021*. IEEE.
[38] Yifan Yang, Tian Qin, and Yu-Heng Lei. 2017. Real-time eSports Match Result Prediction. *CoRR* abs/1701.03162 (2017). arXiv:1701.03162 http://arxiv.org/abs/1701.03162
[39] Zelong Yang, Zhu Feng Pan, Yan Wang, Deng Cai, Shuming Shi, Shao-Lun Huang, and Xiaojiang Liu. 2020. Interpretable Real-Time Win Prediction for Honor of Kings, a Popular Mobile MOBA Esport. *CoRR* abs/2008.06313 (2020). arXiv:2008.06313 https://arxiv.org/abs/2008.06313
[40] Ronald Yurko, Francesca Matano, Lee F Richardson, Nicholas Granered, Taylor Pospisil, Konstantinos Pelechrinis, and Samuel L Ventura. 2020. Going deep: models for continuous-time within-play valuation of game outcomes in American football with tracking data. *Journal of Quantitative Analysis in Sports* 16, 2 (2020), 163–182.

A PARSING CSGO DEMOS

To parse CSGO demos, we use the awpy library, which provides functions to analyze, parse and visualize CSGO demos. CSGO demos contain an event stream which is encoded using Google's Protocol Buffers [32]. Bednarek et al. [1] provide an in-depth description of CSGO demo files and the challenges in working with them. Awpy decodes and organizes the event stream into a series of rounds and frames. The parsed output is a JSON file, and contains not only round events, such as kills, damages or grenade throws, but also the game states. We show the example parsing code and example output in Figure 7. Awpy also provides functions to clean the parsed data, which often contains artifacts like warmup rounds or round restarts, which are especially prevalent in professional demos due to internet failures or third-party server plugins. We use these functions to clean the demos. Additionally, we remove demos from our data set which present obviously incorrect data, such as demos which have more than five players playing on a single side. It is important to note that we are not able to identify cheating players, which are more likely prevalent in our MM data than in FACEIT or HLTV data. We do not include demos played on Ancient since it was introduced to the competitive map pool only a few days before our data begins, and thus most players had not played the map.

The key parameter when parsing CSGO demos is the parse rate, which determines how many frames per second the output JSON file contains. We can think of each frame as a "snapshot" of the game (game state), and we collect these game states at 1-second intervals. These game states are recorded in a list in the "frames" key for each round. With the aforementioned parameter settings, parsing a demo on a computer running Ubuntu 20.04 and an AMD Ryzen 5 1600 takes approximately 3-5 seconds, depending on the size of the demo. The size of the demo file is a function of both the length of the game, which is dependent on both the number and length of the rounds, as well as the tick rate of the demo, which is how often the server writes to the demo. For MM matches, the game servers have a lower tick rate of 64 compared to matches on HLTV or FACEIT, which have a tick rate of 128. MM demos are

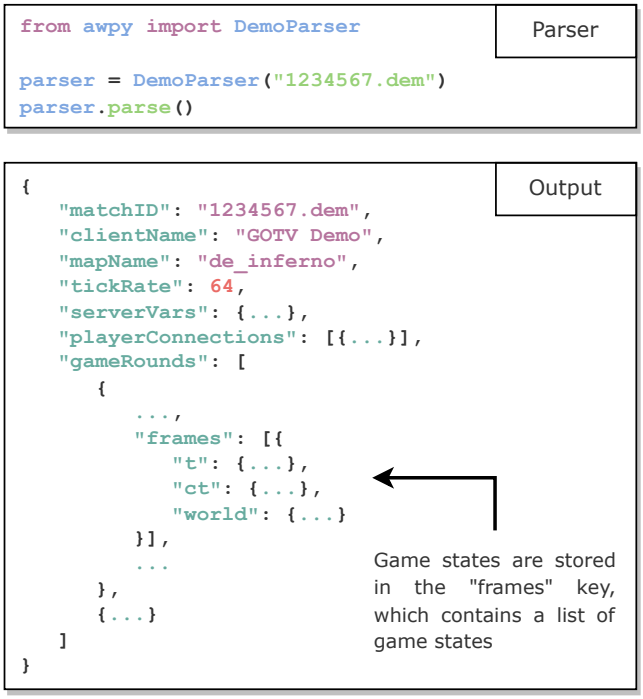

Figure 7: Example parsing code with example JSON output.

usually around 50-100 megabytes for MM demos while professional demos are usually between 100 and 400 megabytes.

B EXPLAINABLE BOOSTING MACHINES

We use the explainable boosting machine (EBM) implementation provided by [24]. Although EBMs are generally slow to train, obtaining predictions from them is usually quick since each feature function is essentially a lookup table, and the contributions from each feature are then simply added. EBMs present a few advantages over conventional GAMs. The first advantage is that each feature function is learned using techniques like bagging and gradient boosting. Furthermore, the effects of collinearity are mitigated by a round-robin procedure which trains on one feature at a time. The second advantage of EBMs is that they automatically detect pairwise interactions between features and include these interaction effects in the final model. In [24], the default number of interactions is set to 10. Further information on EBMs can be found in [17, 18].