

Winning Fights in VALORANT: A Predictive Analytics Approach

Paper Track: Other Sports

By: NRG DeMars DeRover @DeMarsDeRover

NRG JoshRT, ANDROID, eei, hazed, s0m, tex, Jaime Cohencia

1. Motivation and Background

In the era of analytics, we have found many new ways to analyze data from old sports – as new sports take time to gain popularity and accumulate that data. This was not the case for VALORANT, a new Esport by Riot games released in June 2020 [1].

VALORANT soared to a player base of 15 million and released a full match history data API within the first month [2]. From there, teams were assembled to compete for over 5 million USD in the 2020 and 2021 season [3]. There was an immediate demand for competitive advantages generated from analytics. Given that the data was openly available, the community stepped up to meet this demand.

As a competitive shooter, VALORANT revolves around winning gun fights against the enemy, however, not all fights are created equal:

- you choose to play an agent for a match, some are naturally stronger in combat
- winning one round gives you better weapons and an advantage next round
- certain agents have abilities which increases their likelihood of winning fights
- the geometry of the map favors certain positions (e.g. sharp corners to hide in)

It is difficult to tell if a player's stats reflect individual skill or a weapon/agent advantage. We identify two areas for analytical improvement for professional players and coaches.

1. **Player evaluation:** We build a model which predicts the probability a player will win or lose a gun fight against the enemy. We track this predicted probability and see if a specific player is over or underperforming their expected kills. This statistic is similar to Completion Percentage Above Expected (xComp +/-) in the NFL [4].
2. **Player development:** Using interpretable methods, we can also quantify and visualize certain advantageous gunfights. Players can then adjust their playstyle to increase their chances of winning fights.

Both these considerations are being applied at a professional VALORANT team: NRG Esports, currently ranked 12th in North America [5].

1.1. VALORANT

VALORANT is a 5v5 competitive shooter. Each match takes place on a specific map and is broken into rounds, the first team to 13 rounds wins.

Gun fights - where two players from opposing teams shoot each other - are the most critical part of winning matches.

- Teams typically win a round by winning 5 gun fights, as each fight eliminates one enemy
- Teams can also win rounds by planting or defusing a spike at certain areas of the map, which is significantly easier after winning fights and outnumbering the enemy
- Each team of 5 takes one of two sides and switch sides after the first 12 rounds.
 - The attackers: eliminate all defenders or plant the spike without allowing defusal
 - The defenders: attempt to eliminate all attackers or defuse the spike if it was planted
- Performing well on the current round gives teams in game credits to purchase better weapons and abilities at the start of the next round

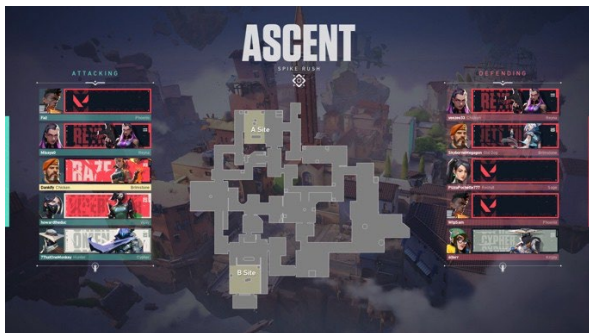


Figure 1 Loading screen before entering a match, shows the 2 teams, 5 players and their respective agents selected



Figure 2 POV of one player on Ascent A site, playing the agent Phoenix



Figure 3 Cloud 9 attacking Ascent A Site in a professional match



Figure 4 The data collected from the VALORANT API, stored on runitback.gg [6]

1.2. VALORANT Champions Tour

The VALORANT Champions Tour is the top competitive tournament for professional players worldwide. It is also sponsored and organized by the developers of the VALORANT, Riot Games [7]. Teams compete from seven major regions to determine the best professional VALORANT team globally [8].

One major challenge facing new sports is player evaluation. Teams in the Champions Tour have seen significant roster changes in professional level play [5] [6], more than many other established Esports [9] [10]. There exists a need to accurately assess a player's underlying ability beyond basic reported stats such as kills, or damage dealt per round to identify new talent and build winning teams.

Furthermore, the game is undergoing constant updates with new agents, abilities, and maps being added. There is a need to quickly identify the new "meta" or optimal strategies before major tournaments. Data collected from Champions Tour matches in Europe or Korea and other VALORANT tournaments have been used to find successful strategies before North American events even begin [6].

2. Data

The VALORANT API is built off the same framework as Riot's already popular Esport League of Legends [2]. A website called RunItBack tracks competitive matches and saves the data for their own services and the community [6]. Each gun fight occurs at a set time, in a specific round in a specific match. To describe *each particular gun fight* we have the following features:

Column Name	Data	Description
<u>Won</u>	{0,1}	1 if player won the gunfight (a kill), 0 if lost (died)
mapName	string	Name of map for current match
isAtk	{0,1}	1 if player is on attacking side, 0 if on defending side
px,py	float	x, y position of player on map
ex, ey	float	x, y position of the other player that is fighting back (the enemy)
pgun, egun	int	in game price of gun player is using, in game price of gun enemy is using*
parmor, earmor	{0,1,2}	player 0 = no armor 1 = half armor 2 = full armor, enemy armor same mapping
pagent, eagent	string	agent name that player's character, agent name of enemy character
atkNumGun, defNumGuns	[0,5]	Number of expensive** weapons on attacking team, on defending team
atkAlive, defAlive	int [0,5]	# of attackers alive (at start of engagement), # of defenders alive
roundTime	float	Seconds passed since start of round
isPlanted	{0,1}	Has the spike been planted
sitePlanted	{'N', 'A', 'B', 'C'}	N = spike not planted, A, B, C spike planted in that location
spikeBPS	{-1,1,2,4,8}	-1: Not planted, More beeps = closer to spike being detonated -> attackers win
atkWonLast, defWonLast	{0,1}	1 if attackers won last round, 1 if defenders won last round
roundNumber, roundHalf	int [1, 24+)	Current round, Current round of half (teams switch sides ATK/DEF after 12 rounds)
playerId, enemyId	int	Unique ID of player being analyzed, unique ID of enemy being analyzed
roundId, matchId, seriesId	int	Unique ID of current round, match and series
eventId	int	Unique ID of current event (unique tournament)
datesSTR	string	Date string of match

* Higher priced weapons are almost always better

** Expensive guns are the 4 weapons costing 2900 or more, the best guns in the game

The variable we are trying to predict is “Won” but we refrain from using all the features and only use the ones in bold text. Features such as the ID of the round or players are too specific to be useful, we want the model to be trained to a “generic” player to later see if any specific players can overperform.

2.1. Training Test Split

We will split the data by dates to better reflect real competition. As updates to the game (e.g. small adjustments to the in-game costs of some weapons) affects the model’s accuracy.

For this paper, we train and validate the model on matches played from August to September 2021 and then test on the next big tournament in October. In December, the final VALORANT event of the season is occurring [8], and the model’s accuracy will also be evaluated on matches daily.

The dimensionality of the data used for training and testing is as follows:

Dataset	# of fights (rows)	Description
Training	10956	Stage 3 Challengers 1 and 2, 72 matches total
Validation	5779	Stage 3 Challenger Playoffs, 39 matches total
Testing	5542	VCT NA Last Chance Qualifier, 37 matches total

3. Modelling

Winning ($Won = 1$) losing ($Won = 0$) a fight is binary classification task. This can be roughly calculated using features such as $pgun$, $egun$, $atkAlive$, $defAlive$. The challenge lies with the spatial data px , py , ex , ey in a complex environment. The model may not know that a wall exists at $y = 10$ so that players on one side of the wall is fighting in a completely different environment than the other side. When the data is plotted in 2D in Figure 5, we can clearly see where these walls would exist:

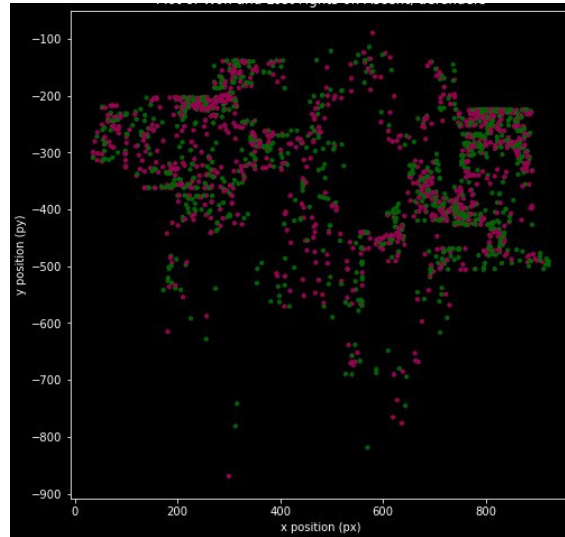


Figure 5 Plot px , py , with kills (green) and deaths (red) for all matches on Ascent, defense side

We naturally gravitate to tree models which partition the data at specific x and y values. Random Forests have performed particularly well in other Esports classification tasks [11] [12]. We will use them as the baseline. We also consider a few other state-of-the-art models:

RandomForest	Collection of decision trees, ensembled [13] <ul style="list-style-type: none"> ■ Pros: Easy to cross validate and tune (out of bag predictions) ■ Cons: Not as accurate as boosted models [14]
XGBoost	Gradient boosted trees, like random forest but trained sequentially [15] <ul style="list-style-type: none"> ■ Pros: Very accurate ■ Cons: Takes a long time to tune [16]
LightGBM	Light Gradient Boosted Machine, like XGBoost but faster [16] <ul style="list-style-type: none"> ■ Pros: Very accurate ■ Cons: Still not as fast as RandomForest in cross validation [17]
Hyperplane trees	From the Interpretable AI library [18] <ul style="list-style-type: none"> ■ Pros: Can split data diagonally [19], good for geometry of map ■ Cons: Slow to train, license restrictions [18]

3.1. Model Evaluation Metric

We are more interested in the probability that a player wins a gun fight rather than a binary prediction. We use these probabilities for player evaluation and to measure impact of changing game states. All four of the models above are able to predict probability. To evaluate the accuracy of these probabilities, we use a metric known as the Brier Score [20], defined as:

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^N (\hat{p}_i - o_i)^2$$

Where \hat{p}_i is the predicted win probability between 0 and 1 and o_i is the actual outcome binary 0 or 1. N is the total number of events forecasted. The Brier Score is always positive. A lower Brier score means the predicted probability is most similar to the outcome, with 0 being ideal.

3.2. Out of sample results

Using the training data described in Section 2.1 we use cross validation to tune the hyperparameters of the above models. We then use the validation set to see which of the four model performs the best on new data, then apply the selected best model on the final test set. The results are shown in the table below.

MODEL	In Sample CV Brier Score	Out of sample Validation Brier Score	Final Test Brier score
Random Forest	0.229	0.235	N/A
XGBoost	0.223	0.231	0.228
LightGBM	0.223	0.232	N/A
Hyperplane Trees	0.226	0.242	N/A

Having selected the best model, we can use a calibration plot in Figure 6 to investigate predictions on the final test set. The x axis shows a collection of predictions in a certain probability bin (eg. roughly 900 fights had a predicted win probability between 20-30%) and the y axis shows the actual outcome win fraction of those 900 fights (roughly 26%).

Per the histogram on the bottom of Figure 6, the model predicts the majority of fights to be close to a 50/50. This was to be expected since rounds are often played with even loadouts and neutral positions.

Nonetheless there are still many gun fights which naturally favor one player, our goal is to:

- 1) help team building by seeing which players can win consistently, even in disadvantaged fights
- 2) help players by interpreting what factors lead to having an advantage in the first place

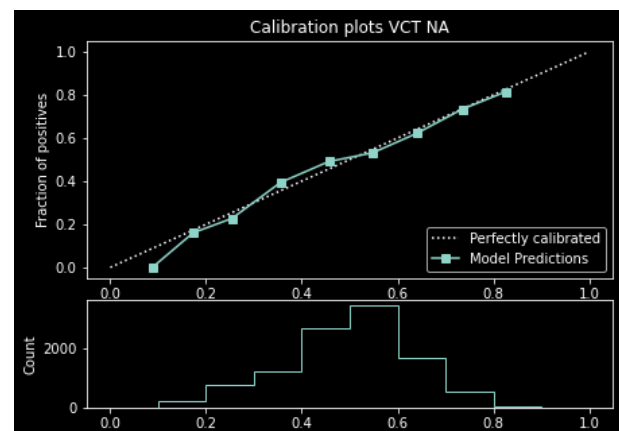


Figure 6 Calibration plot and histogram for all predictions in test set using Xgboost model

4. Takeaways

4.1. Player evaluation

The test set is a crucial tournament for North American teams. We then calculate *wins above expected* for this tournament as an average of all the gunfights a player takes using this equation:

$$\text{Wins Above Expected} = \frac{1}{N} \sum_{i=1}^N (o_i - p_i)$$

Here, $o_i \in \{0,1\}$ is the actual outcome of a single gun fight $p_i \in [0,1]$ the predicted probability that a player would win. If the model predicts a p_i of 0.3 and the player wins the fight they gain 0.7 towards their wins above expected, and if they lose -0.3 is added to their sum.

We calculate this for all N fights a player takes their matches, higher positive numbers are better. We plot it against their kill death average. KDA (*kill death average*) = number kills/won fights divided by number of deaths/lost fights in Figure 7.

Looking at the predicted performance in our test set, we observe that:

- Players with a higher KDA tends to win more fights and thus also perform well in wins above expected, there is correlation between these features, as expected.
- Looking purely at the Y axis, we can claim that players like *aproto* and *supamen* are actually better at winning gun fights than top players by KDA.

Some players perform extremely well above expected yet still only have a 1.0 KDA, this means they were in many disadvantaged fights yet still won many of them. While others can perform below expected but still end up with many kills, this usually happens when you play on the winning team with lots of easy fights.

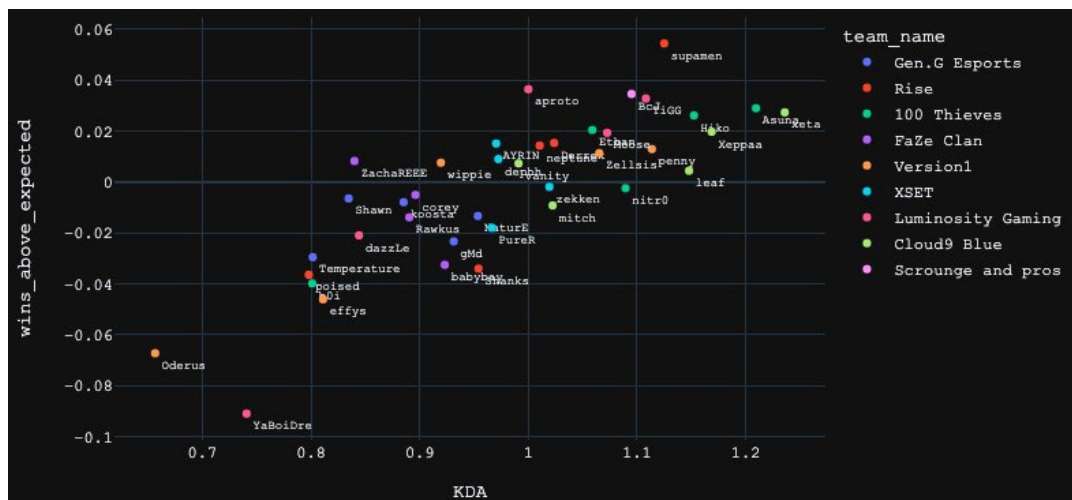


Figure 7 Wins above expected against KDA for all players in test set

We claim that wins above expected is a more robust metric of a player's performance than the current stats used by the game itself: KDA and ADR (average damage per round) [1]. Between the two major NA tournaments in the validation and test set, our new metric showed a higher correlation between players than the two metrics used by Riot Games. Players who win more challenging fights will continue to do so while their KDA and ADR may fluctuate based on other

circumstances such as in game economy and the playstyles of their opponent.

Metric	Between Tournament Correlation
Kill death ratio (In game metric)	0.172
Average damage score (In game metric)	0.185
Wins above expected (New metric)	0.234

4.2. Player development/strategy

We present some findings here for a generic audience. More recent findings are kept confidential to NRG, however the code and data is openly available for the community to their own investigations.

4.2.1. Individual strengths and weaknesses

By looking at where players tend to win more than expected we can identify their strengths and leverage them in game. For example, *nitr0* on 100 Thieves consistently outperforms when using the \$800 sheriff, win a Wins above expected of 0.059 in these situations.

Nitr0 won 37% of fights when using a \$800 weapon against a premium \$2900 weapon. The model expected an average player to win only 31% of these fights. The particular fight shown in the Figure 8 below had a win probability of 23%, by winning, *nitr0* added 0.77 to his wins above expected total.

We also show the Shapley force plot [21] [22] in Figure 9 to gain an intuition as to why this kill was difficult. *Nitr0* had an inferior weapon, and the attacking team had 5 guns this round and outnumbered him at this particular moment.

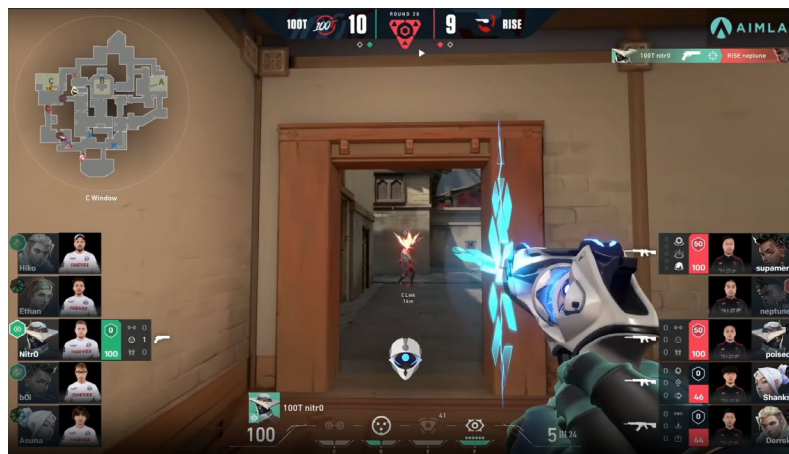


Figure 8 100T vs Rise match, kill by 100T nitr0 (POV) on Rise Neptune [3]

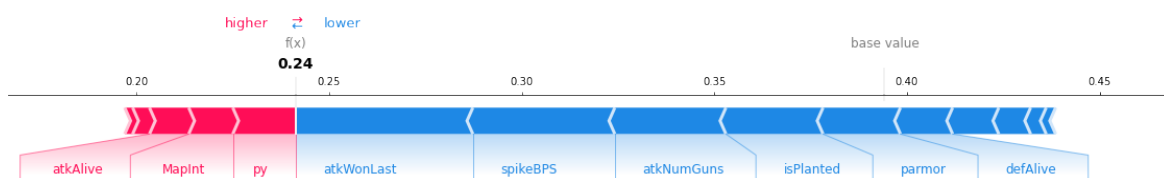


Figure 9 Shapley force plot of kill from Figure 8, describes why the win probability was only 0.24

Conversely, nitr0's teammate *Hiko* struggles against one particular gun in the game, the \$4700 Operator. His wins above expected against this gun is -0.099. Although he did win 40% of his fights against the Operator, the model was able to recognize that some of his kills were not difficult, like this one shown Figure 10 and 11. His team was heavily favored as attackers and the enemy was trapped far away from the site. He was expected to win with 66% probability, thus a kill here is relatively "easy".



Figure 10. 100T vs GENG, kill by 100T Hiko (middle right) on GENG Nature (middle left) [3]

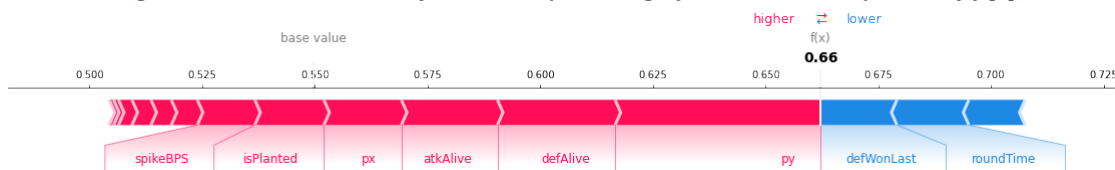


Figure 11 Shapley force plot of fight from Figure 10, describes by Hiko was favored despite having an inferior gun

Given the circumstances in all of *Hiko*'s fights against the operator, the model predicted a player would win 49% of those fights, thus his 40% mark actually shows a weakness against this particular weapon.

4.2.2. Phantom vs Vandal

The model is also able to help with generic decisions in game for all players. As an example, the two most popular weapons in the game, the Phantom and the Vandal cost the same amount of in game credits (\$2900) and is often chosen based on a player's preference. Looking at the gunfights which took place with either the Phantom or Vandal, the model is able to

- predict the probability of winning with the current gun
 - predict the probability of winning if the other gun was used, by changing one feature in the data
- Looking at the difference between the two we see that the phantom is clearly favored on some maps but not unanimously as some of the community is led to believe [23].

Map	Side	Vandal Avg	Phantom Avg	Phantom Advantage
Breeze	ATK	0.508	0.490	-0.018
Icebox	DEF	0.509	0.493	-0.016
Breeze	DEF	0.507	0.496	-0.011
Ascent	ATK	0.499	0.498	-0.001
Icebox	ATK	0.500	0.507	0.007
Split	ATK	0.489	0.505	0.016
Haven	ATK	0.488	0.506	0.018
Bind	ATK	0.493	0.518	0.025
Ascent	DEF	0.476	0.528	0.052
Bind	DEF	0.468	0.523	0.055
Split	DEF	0.466	0.533	0.067
Haven	DEF	0.465	0.542	0.077

We can go even further and look at what specific locations favor each of the two guns on a particular map. We see that the less popular Vandal is superior around the perimeter where longer engagements are expected, the Phantom remains the best around close corners. Note that this map is for attackers, they can start with a Vandal at the perimeters and then “pick up” a Phantom if they win their first fight in the interior

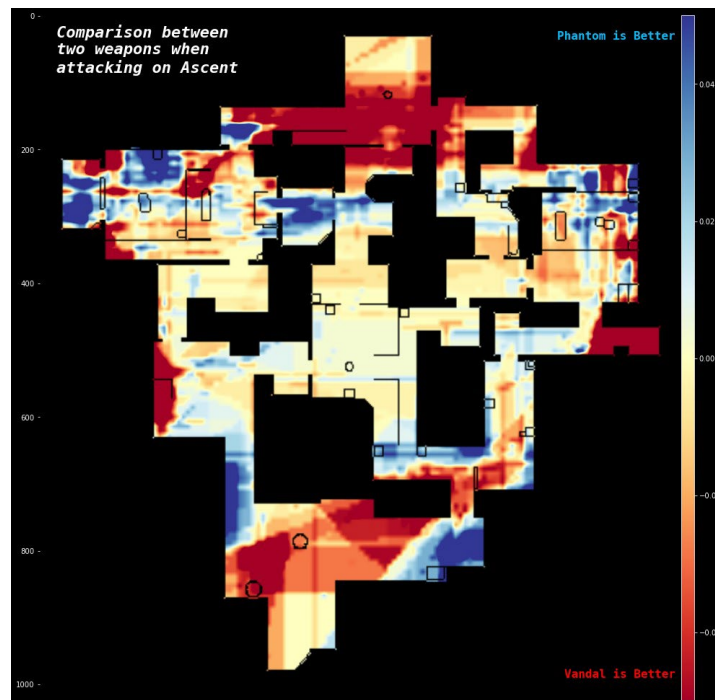
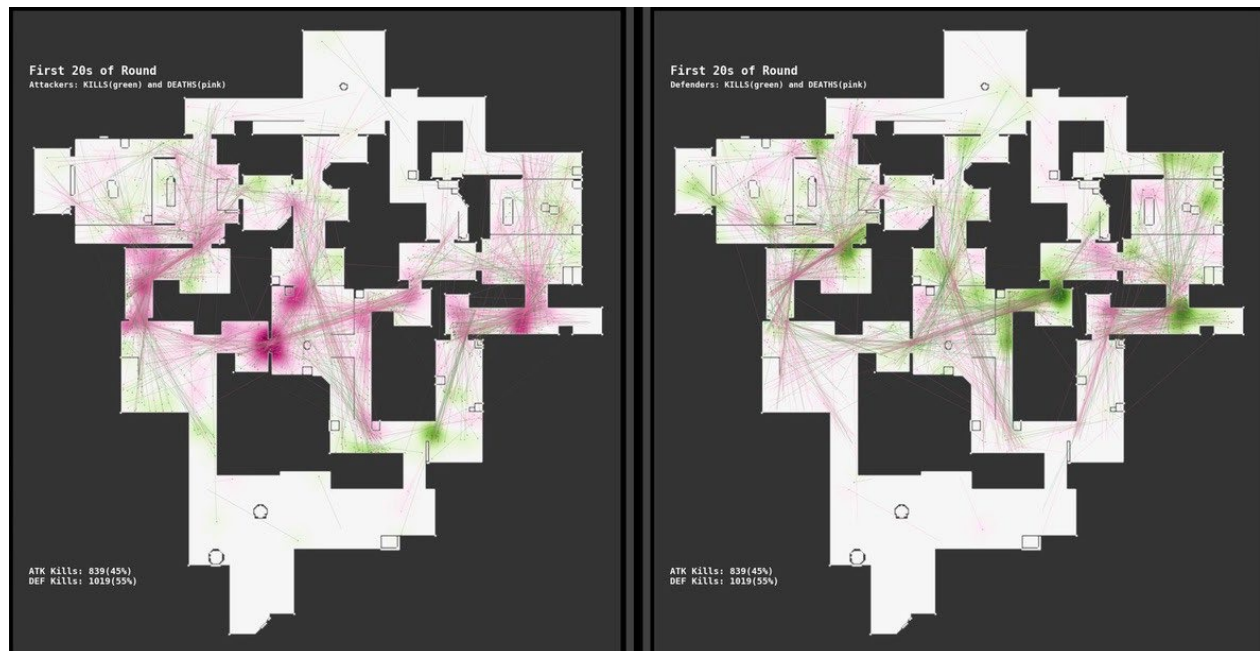


Figure 12 Locations where Phantom (blue) or Vandal (red) is the superior choice

4.2.3. Optimal Positioning

One weakness of the model is that it is unable to predict when or where a fight would occur. Only once we know when and where, it is able to predict the winner. Although the common angles and locations of fights are well known by professional players, we can also plot those locations using a heatmap, this is particularly useful for new maps which are added every season of in the game and scouting the tendencies of a particular opponent.

In these maps, pink area shows where players are losing fights while green areas show winning. The left plot is for attackers at the start of the round, who are typically disadvantaged until they break through the defense. The map on the right show's defenders, also for the first 20 seconds of the round having a clear advantage.



Note that these maps only gives a broad overview of positioning, as it does not consider the other game states of weapons/armor used and timings of a round.

Models which aim to predict when and where these fights occur are in development [6]. They would have to consider team and player tendencies (in the sports world: which play/strategy will a team likely run?) rather than the current model that is trained universally across the player base (in the sports world: what are the chances this play/player is successful?).

5. Conclusion

Current VALORANT metrics of kills/deaths per game and damage per round do not consider the context of which a fight is won or lost. It is much easier to accumulate these stats on a winning team with in-game economy/equipment advantages, or when using more powerful agents with an ability advantage. In this paper we:

1. Created a more robust metric to measure a player's underlying ability across gunfights. This is useful for team building in the professional scene
2. Used the model to identify the strengths and weaknesses of specific players or specific decisions, this is helpful for developing talent and taking more optimal fights

Nonetheless, VALORANT is more than simply winning gunfights and a player's communication, teamwork and ability usage greatly impact winning [24]. It is important to build statistics which accurately reflects a player's value, gunfights are the first step. Having all these metrics will help new talent stand out and help grow the sport.

Lastly, we note that Esports reaches a young audience who are passionate about their teams and have built a community around the game. Another model of "win probability added" was suggested and is being development with the community [25]. The open source nature of this work will be help introduce the field of analytics and data science to young gamers through their passions.

References

- [1] R. Games, "Play VALORANT," 2020. [Online]. Available: <https://playvalorant.com/en-us/>.
- [2] "VALORANT API Launch and Policies," Riot Games, 8 Jul 2020. [Online]. Available: <https://www.riotgames.com/en/DevRel/valorant-api-launch>.
- [3] "VALORANT Champions Tour 2021," Esports Charts, 2021. [Online]. Available: <https://escharts.com/tournaments/valorant-champions-tour-2021>.
- [4] N. Next Gen Stats, "Next Gen Stats: Intro to Completion Probability (Part II)," 28 9 2018. [Online]. Available: <https://www.nfl.com/news/next-gen-stats-intro-to-completion-probability-part-ii-0ap3000000967238>.
- [5] Vlr.gg, "NRG Esports," 1 11 2021. [Online]. Available: <https://www.vlr.gg/team/1034/nrg-esports>.
- [6] Runitback, "Runitback.gg," 1 11 2021. [Online]. Available: <https://rib.gg/>.
- [7] R. Games, "2021 VALORANT CHAMPIONS TOUR," 26 03 2021. [Online]. Available: <https://playvalorant.com/en-gb/news/esports/2021-valorant-champions-tour-overview/>.
- [8] Liquipedia, "VALORANT Champions 2021," [Online]. Available: https://liquipedia.net/valorant/VALORANT_Champions_Tour/2021/Champions.
- [9] "VALORANT 2021 offseason roster changes tracker," Dot Esports, 01 11 2021. [Online]. Available: <https://dotesports.com/valorant/news/valorant-2021-offseason-roster-changes-tracker>.
- [10] "All OVERWATCH League Roster Moves for the 2021 Season," ESTNN, 20 11 2020. [Online]. Available: <https://estnn.com/complete-overwatch-league-post-season-2020-roster-tracker/>.
- [11] A. K. Simon Demediuk, "Performance Index: A New Way To Compare Players," *MIT Sloan Sports Analytics Conference*, 2020.
- [12] M. J. A.-C. Jason G Reitman, "Esports Research, A Literature Review," *Games and Culture*, 2020.
- [13] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, 2001.
- [14] L. Freeman, "Data Science Framework," 2017. [Online]. Available: <https://www.kaggle.com/ldfreeman3/a-data-science-framework-to-achieve-99-accuracy#Step-5:-Model-Data>.
- [15] C. G. Tianqi Chen, "XGBoost: A Scalable Tree Boosting System," *arXiv*, 2016.
- [16] Q. M. T. F. Guolin Ke, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *31st Conference on Neural Information Processing Systems*.
- [17] D. B. Allison O'Hair, *The Analytics Edge*.
- [18] I. AI, "OptimalTrees Documentation," [Online]. Available: <https://docs.interpretable.ai/stable/OptimalTrees/>.
- [19] J. D. Dimitris Bertsimas, *Machine Learning Under a Modern Optimization Lens*, Belmont, Massachusetts: Dynamic Ideas LLC, 2019.
- [20] U. o. V. Library, "A Brief on Brier Scores," [Online]. Available: <https://data.library.virginia.edu/a-brief-on-brier-scores/>.
- [21] S. Lundberg, "shap," [Online]. Available: <https://github.com/slundberg/shap>.
- [22] C. Molnar, "9.6 SHAP (SHapley Additive exPlanations)," [Online]. Available: <https://christophm.github.io/interpretable-ml-book/shap.html>.
- [23] J. Wilkinson, "The Vandal is OVERRATED! Here's why... — Plat Strats," 13 6 2021. [Online]. Available: <https://www.youtube.com/watch?v=EpvR1r5TcDA>.
- [24] Runitback.gg, "Interview with Zombs," 14 8 2021. [Online]. Available: <https://rib.gg/article/rib-gg-interview-with-zombs-following-the-na-challengers-playoffs-upper-final>.
- [25] ReviveMeStrats, "r/Valorant," 01 11 2020. [Online]. Available: https://www.reddit.com/r/VALORANT/comments/o9kgpd/we_found_common_kill_and_deaths_spots_on_every/.
- [26] [Online]. Available: <https://www.kaggle.com/>.
- [27] A. Gramfort, "Probability Calibration Curves," [Online]. Available: https://scikit-learn.org/stable/auto_examples/calibration/plot_calibration_curve.html.