Universiteit Leiden

Classifying play styles in online multi-player video games

Jacob Otte (s3878090)

Supervisors:
Dr. Arno Knobbe & Guus Toussaint

Bachelor Thesis– Data Science and Artificial Intelligence

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl July 17, 2026

Abstract

This thesis investigates how playstyles can be classified in online multiplayer video games. Furthermore, it proposes a framework for finding suitable player replacements based on playstyle. First, a dataset of Counter-Strike 2 matches is collected. Then a feature space is created that captures player playstyles through behavioral features that consist of spatial-temporal and temporal features. Additionally, another feature space is created through t-distributed Stochastic Neighbor Embedding (t-SNE). The three supervised models K-Nearest Neighbors (KNN) , XGBoost and Random Forest are compared using the Friedman test and stratified cross-fold validation. Also, the original feature space is compared to the t-SNE feature space with a paired t-test. Overall, the results indicate that spatial-temporal features were most informative. A KNN-based player replacement showed promising results. Random Forest achieved the strongest performance overall. While the t-SNE embeddings are great for visualization, they fall short in playstyle classification.

Contents

1 Introduction **1**
- 1.1 The Motivation 1
- 1.2 Counter-Strike 2 1
- 1.3 Research Questions 1
- 1.4 Thesis overview 2

2 Background **2**
- 2.1 Playstyles in Competitive Environments 2
- 2.2 Machine Learning Techniques Applied in Playstyle Analysis 2
- 2.3 E-Sports 2

3 Methodology **3**
- 3.1 Data Preprocessing 3
- 3.2 Feature engineering 3
  - 3.2.1 Contextual Feature Splitting 3
  - 3.2.2 Danger Score 4
  - 3.2.3 Exposure Rate 4
  - 3.2.4 Backstab Rate 4
  - 3.2.5 Spatial Entropy 5
  - 3.2.6 Immobility Ratio 5
  - 3.2.7 Entropy Relative to Team 5
  - 3.2.8 Time of first Engagement 6
  - 3.2.9 Distance to Team Centroid 6
  - 3.2.10 Teammates in Proximity During Engagement 7
  - 3.2.11 Utility Usage 8
- 3.3 Data Post Processing 8
- 3.4 T-SNE 8
- 3.5 Supervised Models 8

4 Experiments **9**
- 4.1 Data Collection 9
- 4.2 Experimental Setup 9
- 4.3 Analyzing feature importance 10
- 4.4 KNN-based Player Replacement 10
- 4.5 Comparing Feature Spaces 10
- 4.6 Comparing Supervised Models 11

5 Results **11**
- 5.1 Analyzing feature importance 11
  - 5.1.1 Important Features for AWP Role 11
  - 5.1.2 Important Features for Opener Role 12
  - 5.1.3 Important Features for Support Role 14
  - 5.1.4 Important Features for Closer Role 15
- 5.2 KNN-based Player Replacement 16

5.2.1 Counter-Terrorist Features 16
5.2.2 Terrorist Features 17
5.2.3 Combined Features 18
5.3 Comparing Feature Spaces 18
5.3.1 Visualizing the original feature space with t-SNE embeddings 18
5.3.2 Counter-Terrorist Features 20
5.3.3 Terrorist Features 20
5.3.4 Combined Features 21
5.4 Comparing Supervised Models 21
5.4.1 Counter-Terrorist Features 21
5.4.2 Terrorist Features 22
5.4.3 Combined Features 23

6 Discussion **24**
6.1 Analyzing feature importance 24
6.1.1 Analyzing important features for AWP players 24
6.1.2 Analyzing important features for opener players 25
6.1.3 Analyzing important features for support players 25
6.1.4 Analyzing important features for closer players 25
6.2 KNN-based Player Replacement 26
6.3 Comparing Feature Spaces 27
6.4 Comparing Supervised Models 27

7 Conclusions and Further Research **27**
7.1 Conclusions 27
7.2 Further Research 28

References **30**

1 Introduction

Correctly identifying the play style of a player is of crucial importance in several areas of sports, such as scouting and tactical preparation [7] [2]. Already, there are many studies attempting to identify team and individual playing styles in traditional sports [18].

1.1 The Motivation

A common problem professional (e-)sports teams face is the unexpected or expected loss of a player. The player may leave the team due to a buyout, retirement or other circumstances. While it is more common for players to leave during the off-season, it is not unusual for transfers to happen during the season. Especially the latter can be a problem for a team. A quick and suitable replacement that fits into the team's dynamic and playstyle is necessary for the team to reduce the negative impact of the transfer.
This problem will be explored in this thesis. The thesis will look into how play styles of players can be classified in online multiplayer video games and how supervised machine learning techniques could help teams to quickly find a suitable replacement to maintain their competitiveness.

1.2 Counter-Strike 2

Counter-Strike 2 (CS2) is a 5v5 first-person-shooter online multiplayer video game. During a game, that consists of two halves, teams alternate between playing as Terrorists (Ts) and Counter-Terrorists (CTs). The Ts can win a round by either eliminating the CTs or by planting and exploding the bomb at one of two bombsites. CTs can win a round by either eliminating the Ts or by stopping the Ts from planting and exploding the bomb. The game includes utility items, such as smoke grenades, high-explosive grenades, molotovs, and flashbangs, which add a tactical layer to the game.
As mentioned above, CS2 is played with a team of five players. Each player in the team is signed with an overall role in mind. A team is usually made up of the five roles: in-game-leader (igl), AWP (sniper), opener, closer and support[12]. In-game-leader also have a secondary role assigned as the role mostly refers to having a voice in strategic decisions during the game.

1.3 Research Questions

This thesis is guided by the following main research question:

RQ: How can play styles be characterized in online multiplayer video games?

Following sub research questions will be answered:

SQ1: How can KNN be used to find a replacement?
SQ2: How does the original feature vector compare to the t-SNE-reduced feature vector?
SQ3: How do supervised machine learning techniques perform in classifying player roles based on play style features?

1.4 Thesis overview

Chapter 1 coveres the reason behind the thesis, introduction to Counter-Strike 2 and states the research questions. **Chapter 2** determines the definition of a playstyle, introduces related work on playstyle analysis in sports and e-sports. **Chapter 3** presents the data pre processing steps, the engineered behavioral features, the data post processing and machine learning algorithms used. **Chapter 4** details the data collection, experimental setup and the experiments used to answer the RQ and the SQs. **Chapter 5** reports the results for each experiment, while **Chapter 6** discusses the experimental results. **Chapter 7** summarizes the results, answers the research questions and proposes further research.

2 Background

2.1 Playstyles in Competitive Environments

A definition for a playstyle is given by Decroos & Davis [7], as they define that a player's playstyle can be characterized by the areas the player occupies and the actions this player makes in those areas. Additionally, they mention an important hypothesis that a play style does not change significantly over a season, as it is more of a gradual change. The definition and hypothesis can be applied to online multiplayer games.
Additionally, recent research has shown that a play style significantly differs between different phases of play [16].
Commonly, a playstyle in sports is characterized by spatial and temporal features.
A widely used spatial feature in playstyle analysis is the positional information of an individual. Often, all positions (on a field) of an individual are aggregated into a heatmap [7][11], which can be used to derive spatial features of an individual. Zimmer et al. [21] utilize the raw positional data directly.
An example of a widely used temporal feature is event chaining. Events, such as shots, crosses or passes in football, are chained into sequences[16]. Playstyles can be inferred from those sequences[8], as they capture behavioral patterns of teams and players.
In addition, playstyles can be characterized by several aggregated numerical features that are often domain-specific. Examples include percentages of various actions in football [17] and velocity or equipment value in CS2 [21].

2.2 Machine Learning Techniques Applied in Playstyle Analysis

Mostly, these studies make use of unsupervised machine learning algorithms to classify the different play styles, while deep learning gains popularity [18]. Principal Component Analysis (PCA) and clustering techniques are popular [8] [7][21].

2.3 E-Sports

On the contrary, the research in e-sports is limited. Published research articles are rare and focus on identification of players [21] rather than classifying players' play styles [20]. While some studies apply play style analysis to the game Counter Strike, they make use of the public ESTA dataset with data from 2021-2022 [21]. Since then, Counter-Strike has progressed from the old version, Counter-Strike: Global Offensive (CS:GO), to the new version, Counter-Strike 2 (CS2). Since the meta is continuously changing, especially with the release of CS2, this thesis will make use of a custom dataset with data from the second half of 2025.

3 Methodology

3.1 Data Preprocessing

All needed positional, in-game and personal information was extracted using the open source library **demoparser2** [13]. The information was extracted at one tick per second, while the demos originally contain 64 ticks per second. In order to ensure meaningful features, players were filtered based on a minimum amount of maps (50) played.

3.2 Feature engineering

3.2.1 Contextual Feature Splitting

Playstyles significantly adjust based on the phase of the game [16]. Therefore, certain features will be split on game phases. The **early game** phase consists of all ticks that occur within the first 20 seconds of a round. The **mid game** phase consists of all ticks that happen after the **early game**, while the bomb is not planted. All ticks that happen while the bomb is planted are considered ticks of the **after plant** game phase.
Additionally, features are split on sides (Counter-Terrorist, Terrorist). This is done because sides have inherently different objectives, i.e. defending and attacking.
Table 1 provides an overview of how each feature is split:

Feature	Side-specific	Phase-specific
Danger Score	✓	✓
Exposure Rate	✓	✓
Backstab Rate	✓	✓
Entropy	✓	✓
Immobility Ratio	✓	✓
Entropy relative to team	✓	✓
Time of First Engagement	✓	
Distance to Team Centroid	✓	✓
Teammates in Proximity during Engagement	✓	✓
Utility Usage	✓	

Table 1: Feature splitting strategy.

3.2.2 Danger Score

The danger score was created to have a numerical value that represents how aggressive a player positions themselves. The higher the score, the more aggressive the positioning, e.g. the player spends a lot of time in positions that are death hotspots.
In order to calculate the danger score for each map, a heatmap is computed of all deaths of valid players. The heatmap is then normalized.
Then, for each player, a heatmap is computed of all ticks where the player is alive. The heatmap is then normalized. The danger score for a map can now be computed by the following:

$$D_{p,m} = \sum_{x,y} H_p^{(m)}(x,y) \cdot H_d^{(m)}(x,y)$$

where $D_{p,m}$ is the danger score of a player **p** for map **m**, $H_p^{(m)}(x,y)$ is the player's normalized heatmap for map **m** and $H_d^{(m)}(x,y)$ is the normalized death map for map **m**.

The final mean danger score of a player is computed by taking the mean of all maps:

$$D_p = \frac{1}{|M|} \sum_{m \in M} D_{p,m}$$

Additionally, the standard deviation is computed across all maps:

$$\sigma_p = \sqrt{\frac{1}{|M|} \sum_{m \in M} (D_{p,m} - D_p)^2}$$

where D_p is the mean danger score for player **p** and σ_p is the standard deviation for a player **p** across all maps.

3.2.3 Exposure Rate

Exposure rate was created to be a feature that captures the aggressiveness of a player. More aggressive players tend to be exposed to more opponents than players who carefully pick their duels.
The exposure rate of a player is calculated by making use of the game's internal **approximate_spotted_by** mask, which indicates at every tick to which opponent(s) a player is visible. Ticks where the mask is empty are filtered out. Then, for each player, the mean and standard deviation of all ticks of the length of the mask is taken.

3.2.4 Backstab Rate

The backstab rate captures the rate of a player's engagements where the player is not visible to an opponent. This can either happen because of a flank/backstab or a well-timed peek (when the opponent is looking elsewhere).
The backstab rate is calculated by:

$R_p = B_p/E_p$

where R_p is the backstab rate for player $\mathbf{p}$, B_p is the count of engagements where the player is not visible to an opponent and E_p is the count of all engagements.

3.2.5 Spatial Entropy

The entropy of a player's positional data is calculated. Static players will have a lower entropy compared to roaming players.
For each player and map, a heatmap is computed of the positional data. The heatmap is filtered to only include cells above the 90th percentile. This is done to remove noise. The heatmap is normalized, transforming the heatmap into a probability distribution.
Then the Shannon entropy for each player is calculated by:

$$S(H_p) = -\sum_{i:p_i>0} p_i \log_2(p_i)$$

where $S(H_p)$ is the Shannon entropy for probability distribution $\mathbf{H}$ of player $\mathbf{p}$.

3.2.6 Immobility Ratio

The immobility ratio measures the ratio of ticks a player's velocity is under a certain threshold. The feature is adapted from Herault et al. [9]. They calculated the immobility ratio of climbers by considering complete and partial immobility. Therefore, a threshold (velocity < 20) is set such that small movements are considered immobile, as the player does not actively move towards a point. The immobility ratio is then calculated by:

$R_p = I_p/T_p$

where R_p is the immobility ratio for player $\mathbf{p}$, I_p is the count of ticks where the velocity of $\mathbf{p}$ is below the threshold and T_p is the count of all ticks of $\mathbf{p}$.

3.2.7 Entropy Relative to Team

The feature was created to capture how similar a player's positional probability distribution is to their team. A lower value suggests that the player occupies areas similar to their team, whereas a higher score suggests that the player prefers to occupy areas alone.
As in other features, the player's heatmap is calculated per map. Additionally, the team's heatmap is calculated. Both heatmaps are normalized and therefore transformed to positional probability distribution.
The distributions can be used to calculate the Jensen-Shannon distance (JSD):

$$D_{p,m} = JSD(H_{p,m}, H_{t,m})$$

where $D_{p,m}$ is the JSD for player **p** for map **m**, $H_{p,m}$ is the player's positional distribution and $H_{t,m}$ the one of the player's team.
Finally, the mean distance is calculated across all maps:

$$D_p = \frac{1}{|M|} \sum_{m \in M} D_{p,m}$$

Additionally, the standard deviation is computed across all maps:

$$\sigma_p = \sqrt{\frac{1}{|M|} \sum_{m \in M} (D_{p,m} - D_p)^2}$$

where D_p is the mean Jensen-Shannon distance for player **p** and σ_p is the standard deviation across all maps.

3.2.8 Time of first Engagement

Time of first engagement was created to indicate how aggressive or passive a player is during a round. Aggressive players are expected to have a lower time of first engagement compared to their more passive teammates.
For each round, the time of the first engagement, e.g. when the player is visible to an opponent or an opponent is visible to the player, is calculated. Then, the mean and standard deviation of all timings is taken.

3.2.9 Distance to Team Centroid

The distance to the team centroid was calculated to capture how connected a player is with their team. The lower the score, the more central the player positions themselves in the team, while a high score shows that the player is peripheral.
First, for each tick, we calculate the mean x and y coordinate for each side (team).
Then, the Euclidean distance to the team centroid for each player is calculated:

$$D_{p,t} = \sqrt{(x_{p,t} - x_{c,t})^2 + (y_{p,t} - y_{c,t})^2}$$

where, $D_{p,t}$ is the distance to the team centroid for player **p** at tick **t**, $(x_{p,t}, y_{p,t})$ is the player's position and $x_{c,t}, y_{c,t}$ is the team's centroid.
Finally, the mean is calculated across all ticks:

$$D_p = \frac{1}{|T|} \sum_{t \in T} D_{p,t}$$

Additionally, the standard deviation is calculated:

$$\sigma_p = \sqrt{\frac{1}{|T|} \sum_{t \in T} (D_{p,t} - D_p)^2}$$

where D_p is the mean distance to the team centroid for player **p** and σ_p is the standard deviation across all ticks.

3.2.10 Teammates in Proximity During Engagement

Knowing when and what engagements to take is an important part of the game. Therefore, this feature captures the mean number of teammates in proximity during engagements. A player with a higher score is more tradable in engagements compared to a player with a lower score taking engagements without nearby teammates to help out.
Here, the distance threshold is set at 500 units.
The number of teammates in proximity is calculated as:

$$N_{p,t} = \sum_{q \in M_p} \mathbf{1}\left(d_{p,q,t} \leq 500\right)$$

$$d_{p,q,t} = \sqrt{(x_{p,t} - x_{q,t})^2 + (y_{p,t} - y_{q,t})^2}$$

where $N_{p,t}$ denotes the number of teammates in proximity for player **p** at tick **t** and $d_{p,q,t}$ denotes the Euclidean distance between player **p** and player **q** at tick **t**.
Finally, the mean number is taken across all ticks:

$$N_p = \frac{1}{|T|} \sum_{t \in T} N_{p,t}$$

Additionally, the standard deviation is calculated:

$$\sigma_p = \sqrt{\frac{1}{|T|} \sum_{t \in T} (N_{p,t} - N_p)^2}$$

where N_p is the mean number of teammates in proximity during engagements for player $\mathbf{p}$ and σ_p the standard deviation.

3.2.11 Utility Usage

As mentioned earlier, an important aspect of CS2 is utility. It is expected to be helpful for play style classification. For example, supporting players are expected to throw more utility to set up their team. Therefore, for each player the mean grenades thrown per round is recorded. Additionally, for each grenade type the mean is calculated separately.

3.3 Data Post Processing

After the feature vectors are computed, the features are normalized per team. The thesis makes use of group-wise z-score normalization. This is done to eliminate team playstyle bias. In order to guarantee correct normalization, only teams with five players are considered for the final dataset. This is done because normalization on teams with less than five players results in skewed features.

3.4 T-SNE

T-SNE is a non-linear dimensionality reduction technique that projects high-dimensional data into low-dimensional data while preserving local relationships in data[15][4]. First, t-SNE is used to reduce the dimensions of the original feature space to two dimensions. The resulting t-SNE embeddings are used for visualization.
Additionally, the t-SNE feature space is compared to the original feature space in subsequent analysis.

3.5 Supervised Models

This thesis makes use of three supervised machine learning models to predict player roles.
KNN is a classification algorithm that classifies instances based on the class of their nearest neighbors. The classification can either be conducted by majority voting (uniform distance) or by distance voting, where closer neighbors classes count more. The amount of neighbors the algorithm considers is defined by **K**[6].
XGBoost is a scalable end-to-end tree boosting system. It makes use of caching, sharding and data compression, allowing it to scale well beyond billions of instances. It builds trees sequentially, adding a new tree at every boosting iteration that improves the ensemble prediction the most[5].
Random Forest is an ensemble learning method, consisting of a collection of decision trees. Each decision tree is trained on a subset of the training data and features, reducing the possibility of a single feature dominating the prediction. The core idea is that each tree outputs a prediction and then the final prediction is obtained based on the tree outputs through majority voting. Therefore, the random forest model is robust and less affected by noise in data[3].

4 Experiments

4.1 Data Collection

The dataset is a collection of around 1,300 matches. Only LAN matches were considered, as they offer a more consistent environment as opposed to online matches. Those matches contain 1-5 demo files (one per map), downloaded from **hltv.org**[10]. In total, this results in 2,972 maps. The timeframe includes July 2025 to December 2025. The timeframe was chosen since it spans exactly one season in professional Counter-Strike. According to the hypothesis about gradual change in playstyle [7], it can be assumed that the playstyles were generally stable for all players.
Additionally, the roles (labels) of the players were collected from a positional database, which is maintained by one of the leading analysts in CS2 e-sports [19].

4.2 Experimental Setup

Every experiment follows the same experimental setup. The feature space the models are trained on depends on the experiment, but all experiments use the player roles as target labels.
Each experiment is run on three subsets of the feature space. The subsets are: (1) T-sided features (n=46 features), (2) CT-sided features (n=46 features) and (3) the combined set of (1) and (2) (n=92 features).
Every model used in a setup has its hyperparameters tuned with grid search combined with stratified k-fold validation (k=5). The different combinations of hyperparameters are evaluated based on their mean macro F1 score on the validation folds.
The search grid for each model are shown in the corresponding tables below.

hyperparameter	values
k	[1,30)
weights	distance

Table 2: Search grid for KNN.

hyperparameter	values
n_estimators	[50, 100, 200, 350, 500]
max_depth	[3, 4, 5, 7]
learning_rate	[0.0001, 0.001, 0.01, 0.1, 0.2]
subsample	[0.6, 0.8, 1.0]
colsample_bytree	[0.6, 0.8, 1.0]

Table 3: Search grid for XGBoost.

hyperparameter	values
n_estimators	[100, 200, 500]
max_depth	[None, 5, 10]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
max_features	[sqrt, log2]

Table 4: Search grid for random forest.

4.3 Analyzing feature importance

In order to answer the RQ, to figure out what features best characterize a playstyle, the thesis will compute the importance of features. The importance will be computed per side and per role.
The analysis will be done with **tree-shap** [14].
The model that will be used is the random forest model, as it performs the best on the original feature space (see 5.4). The five most important features will be reported for each role/side. The results will give insights into what behavioral characteristics are important for the prediction of player roles. This can inherently be connected to what features are important to characterize a playstyle.

4.4 KNN-based Player Replacement

SQ1 is answered by conducting a KNN-based player replacement experiment. The experiment follows the experimental setup from 4.2. The KNN model is trained on the original feature space. As results, the hyperparameters will be reported. In order to demonstrate the application of a player replacement with the created feature space, the player **ZywOo** is selected as a case study. The selection was made because he is one of the best-known professional players. The replacements, suggested by the KNN algorithm, will be qualitatively evaluated based on the similarity in role and playstyle.

4.5 Comparing Feature Spaces

In order to answer SQ2, an experiment will be conducted that compares the predictive performance of models on the original feature space with the t-SNE embedded feature space. The t-SNE embedding was generated using the parameters shown in Table 5.
This experiment follows the setup from 4.2. The models are KNN, XGBoost and random forest.
The mean macro F1 score for each feature space will be reported for each model separately. Finally, a paired t-test will be performed between the feature spaces for each model to evaluate whether differences are statistically significant.

hyperparameter	value
n_components	2
random_state	5
perplexity	30
n_iter_without_progress	500
early_exaggeration	20
init	random
learning_rate	16

Table 5: Hyperparameters of t-SNE algorithm.

4.6 Comparing Supervised Models

The objective of this experiment is to answer SQ3. The experiment setup follows 4.2 and compares three different supervised machine learning models: KNN, XGBoost and random forest.
The evaluation is done with the Friedman test. The models are trained and evaluated with stratified k-fold (k=5) over ten random seeds to add robustness. The mean rank for each model will be reported, as well as the Friedman statistic and p-value. Subsequently, the post-hoc Nemenyi test is applied. A critical difference plot will be reported, evaluating whether the differences between models are statistically significant[1].

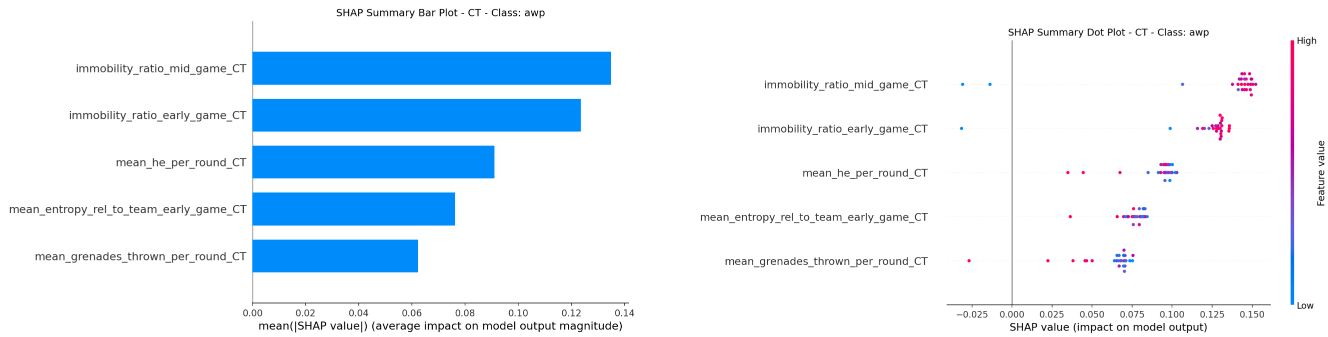
5 Results

5.1 Analyzing feature importance

Below, the five most important features for each role for each feature subset based on their mean absolute SHAP value are reported. An overall bar plot (a) as well as a beeswarm plot (b) is shown. While the bar plot shows the mean absolute impact on the model's output, the beeswarm plot shows the impact for each instance. Additionally, each point is color mapped revealing information about the features value of a given instance.

5.1.1 Important Features for AWP Role

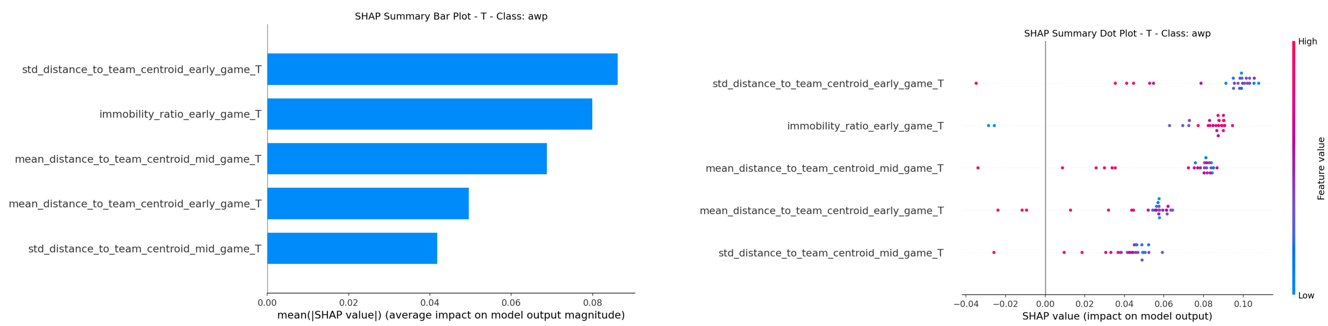
The plots below show the five most important features for classifying AWPers with the Random Forest model. Figure 1 shows the CT-sided feature subset, Figure 2 the T-sided feature subset, and Figure 3 the combined feature subset.



(a) SHAP feature importance.

(b) SHAP beeswarm plot.

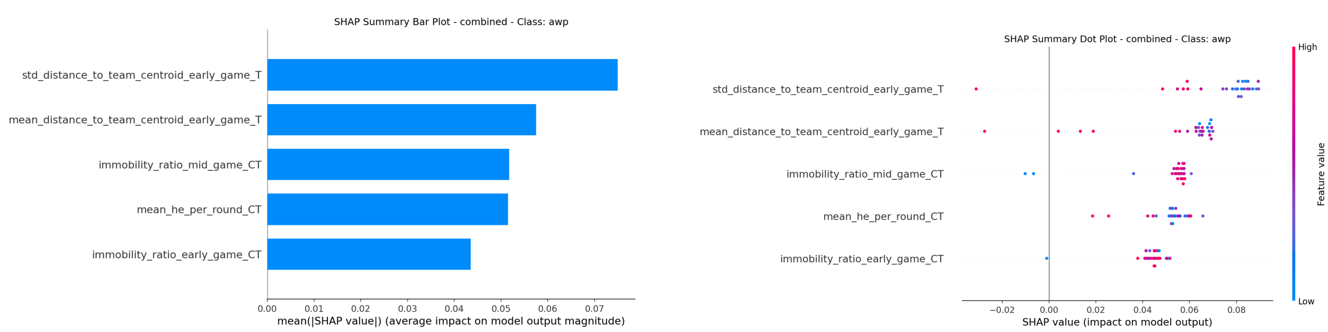
Figure 1: SHAP analysis of the Random Forest model predicting player roles with CT-sided feature subset.



(a) SHAP feature importance.

(b) SHAP beeswarm plot.

Figure 2: SHAP analysis of the Random Forest model predicting player roles with T-sided feature subset.



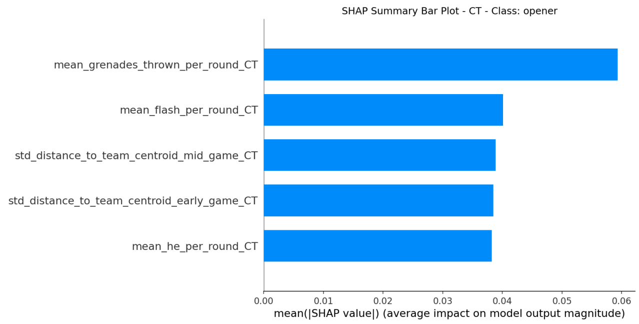
(a) SHAP feature importance.

(b) SHAP beeswarm plot.

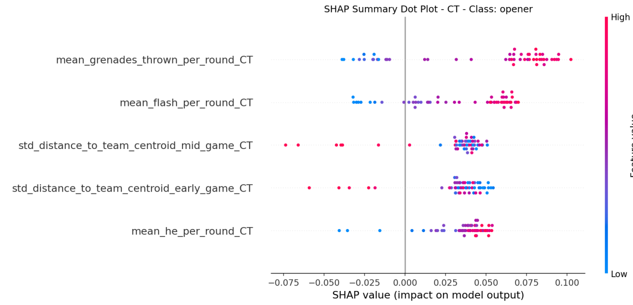
Figure 3: SHAP analysis of the Random Forest model predicting player roles with combined feature subsets.

5.1.2 Important Features for Opener Role

The plots below show the five most important features for classifying openers with the Random Forest model. Figure 4 shows the CT-sided feature subset, Figure 5 the T-sided feature subset, and Figure 6 the combined feature subset.

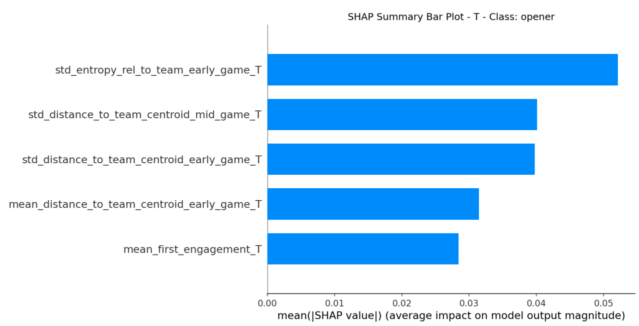


(a) SHAP feature importance.

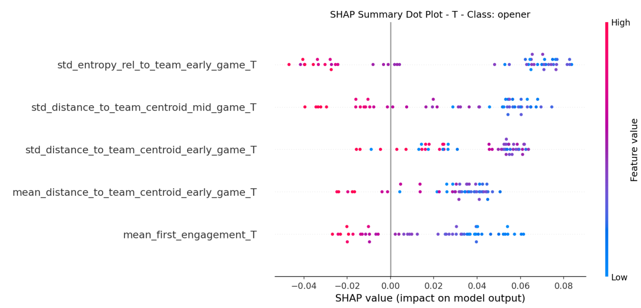


(b) SHAP beeswarm plot.

Figure 4: SHAP analysis of the Random Forest model predicting player roles with CT-sided feature subset.

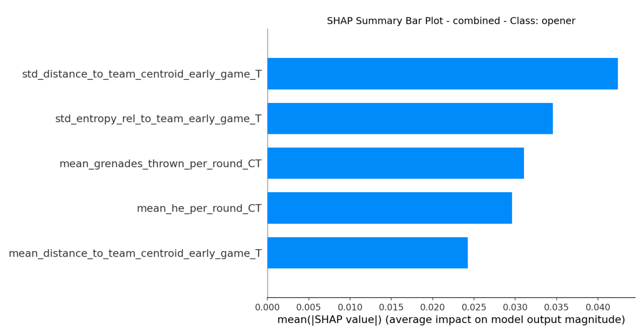


(a) SHAP feature importance.

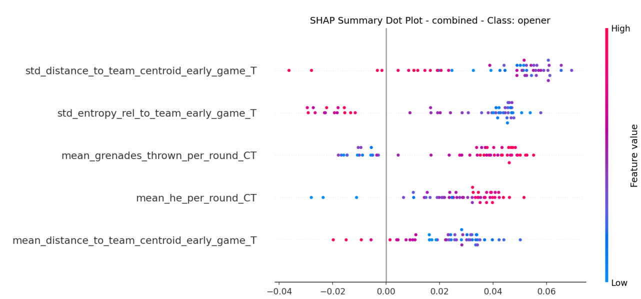


(b) SHAP beeswarm plot.

Figure 5: SHAP analysis of the Random Forest model predicting player roles with T-sided feature subset.



(a) SHAP feature importance.

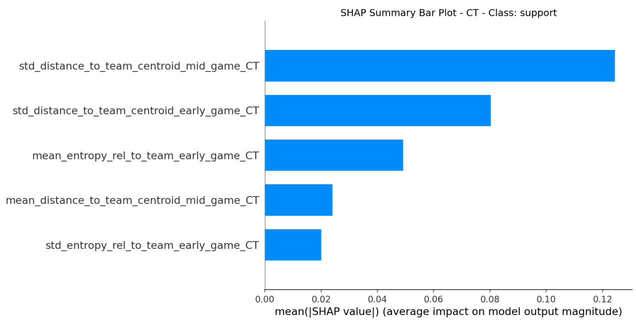


(b) SHAP beeswarm plot.

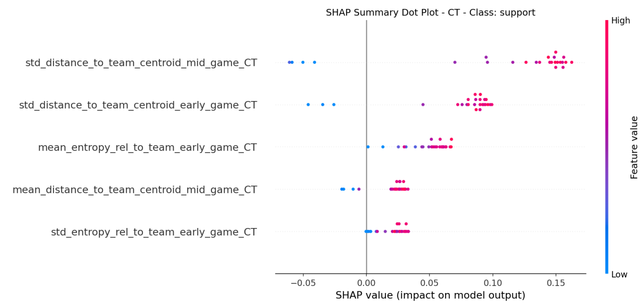
Figure 6: SHAP analysis of the Random Forest model predicting player roles with combined feature subsets.

5.1.3 Important Features for Support Role

The plots below show the five most important features for classifying supporters with the Random Forest model. Figure 7 shows the CT-sided feature subset, Figure 8 the T-sided feature subset, and Figure 9 the combined feature subset.

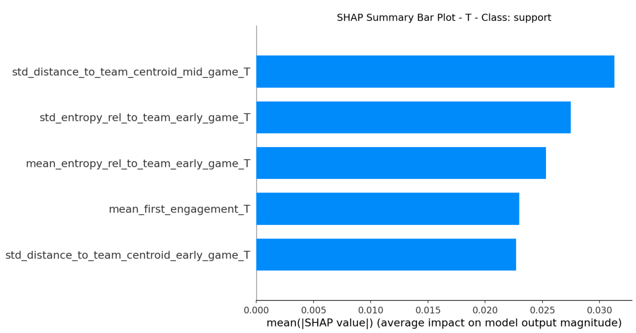


(a) SHAP feature importance.

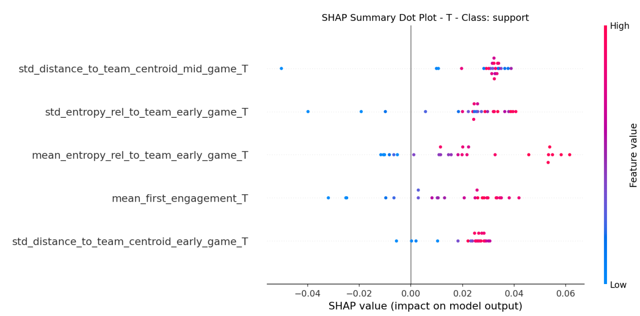


(b) SHAP beeswarm plot.

Figure 7: SHAP analysis of the Random Forest model predicting player roles with CT-sided feature subset.

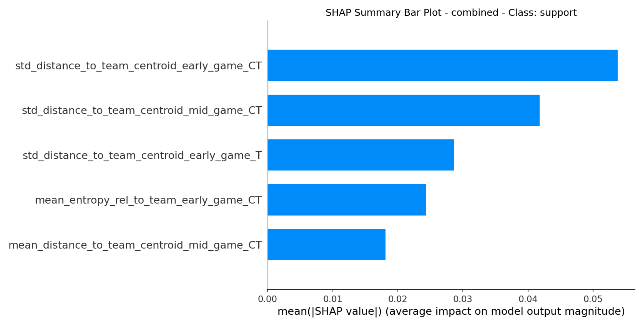


(a) SHAP feature importance.

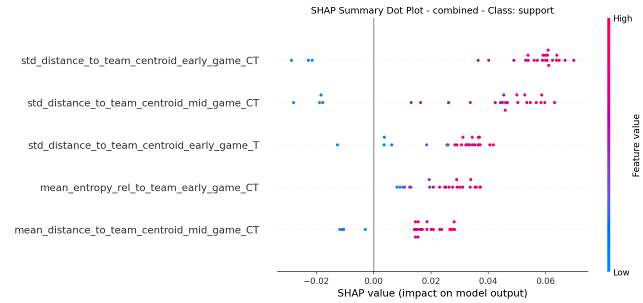


(b) SHAP beeswarm plot.

Figure 8: SHAP analysis of the Random Forest model predicting player roles with T-sided feature subset.



(a) SHAP feature importance.

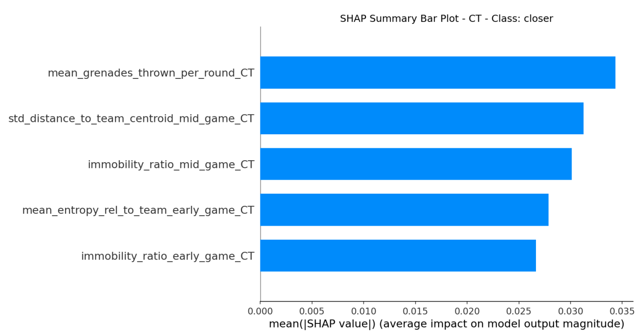


(b) SHAP beeswarm plot.

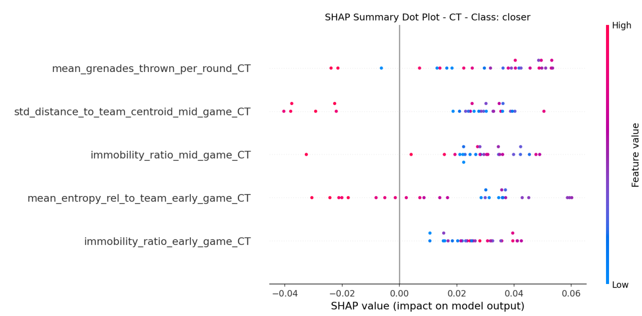
Figure 9: SHAP analysis of the Random Forest model predicting player roles with combined feature subsets.

5.1.4 Important Features for Closer Role

The plots below show the five most important features for classifying closers with the Random Forest model. Figure 10 shows the CT-sided feature subset, Figure 11 the T-sided feature subset, and Figure 12 the combined feature subset.

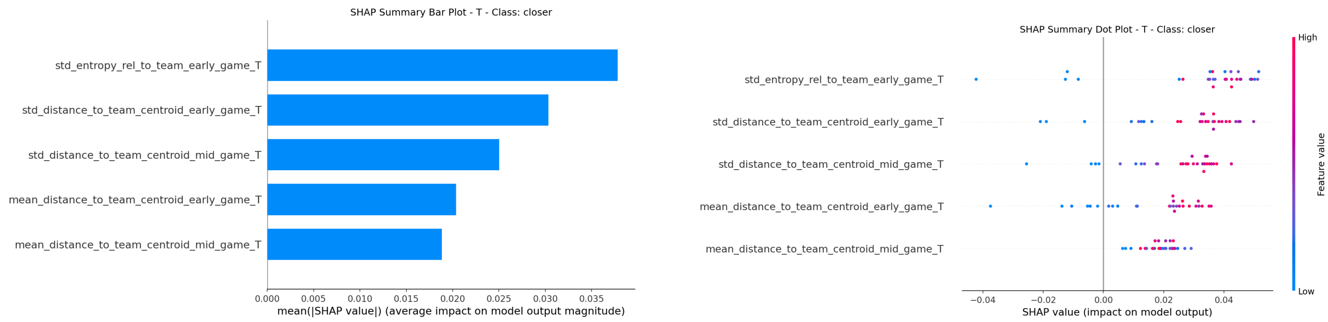


(a) SHAP feature importance.



(b) SHAP beeswarm plot.

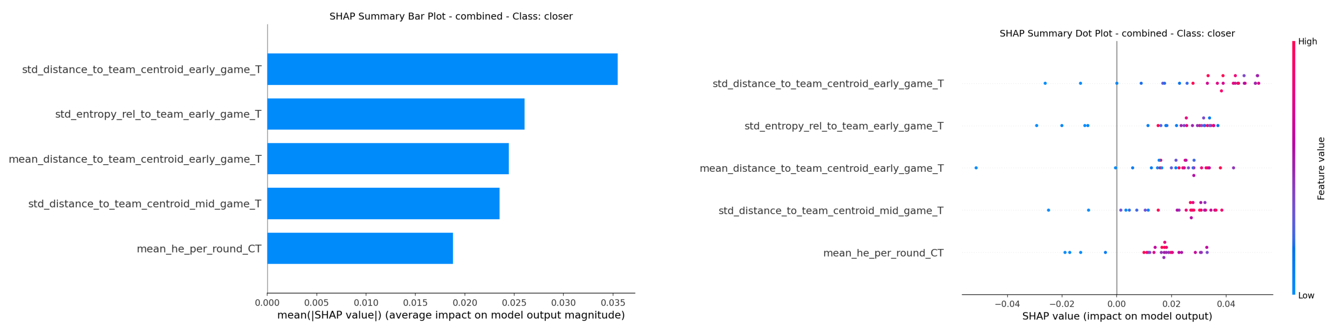
Figure 10: SHAP analysis of the Random Forest model predicting player roles with CT-sided feature subset.



(a) SHAP feature importance.

(b) SHAP beeswarm plot.

Figure 11: SHAP analysis of the Random Forest model predicting player roles with T-sided feature subset.



(a) SHAP feature importance.

(b) SHAP beeswarm plot.

Figure 12: SHAP analysis of the Random Forest model predicting player roles with combined feature subsets.

5.2 KNN-based Player Replacement

5.2.1 Counter-Terrorist Features

Table 6 shows the best hyperparameters for KNN with the CT-sided feature subset that were obtained using the optimization procedure described in Section 4.2. Figure 13 shows the player recommendations with their rank in relation to the target player **ZywOo** (rank 0).

hyperparameter	values
k	17
weights	distance

Table 6: Best Hyperparameters for KNN with CT-sided features.

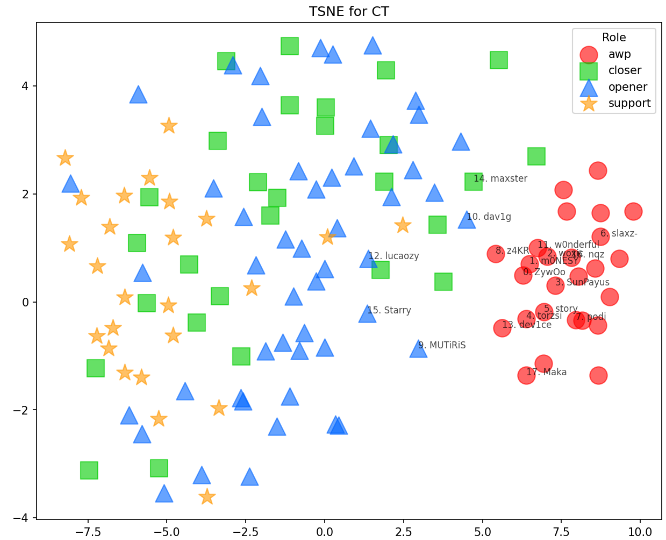


Figure 13: Player replacements visualized by rank with t-SNE (CT).

5.2.2 Terrorist Features

Table 7 shows the best hyperparameters for KNN with the T-sided feature subset that were obtained using the optimization procedure described in Section 4.2. Figure 14 shows the player recommendations with their rank in relation to the target player **ZywOo** (rank 0).

hyperparameter	values
k	3
weights	distance

Table 7: Best Hyperparameters for KNN with T-sided features.

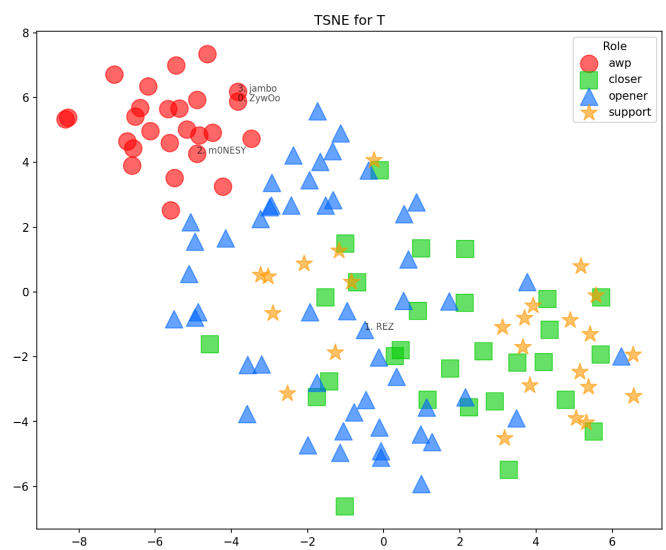


Figure 14: Player replacements visualized by rank with t-SNE (T).

5.2.3 Combined Features

Table 8 shows the best hyperparameters for KNN with the combined feature subset that were obtained using the optimization procedure described in Section 4.2. Figure 15 shows the player recommendations with their rank in relation to the target player **ZywOo** (rank 0).

hyperparameter	values
k	8
weights	distance

Table 8: Best Hyperparameters for KNN with combined features.

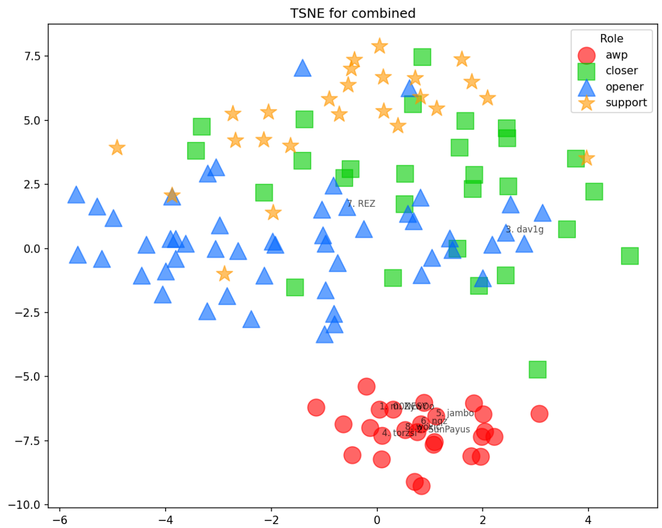


Figure 15: Player replacements visualized by rank with t-SNE (combined).

5.3 Comparing Feature Spaces

5.3.1 Visualizing the original feature space with t-SNE embeddings

The following plots show the t-SNE visualizations of the original feature space for each feature subset. Figure 16 shows the CT-sided feature subset, Figure 17 the T-sided feature subset, and Figure 18 the combined feature subset.

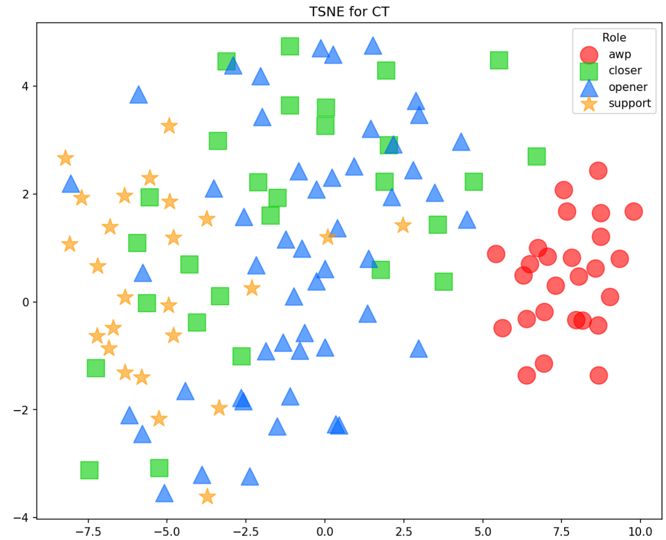


Figure 16: T-SNE visualization of the original feature space (CT).

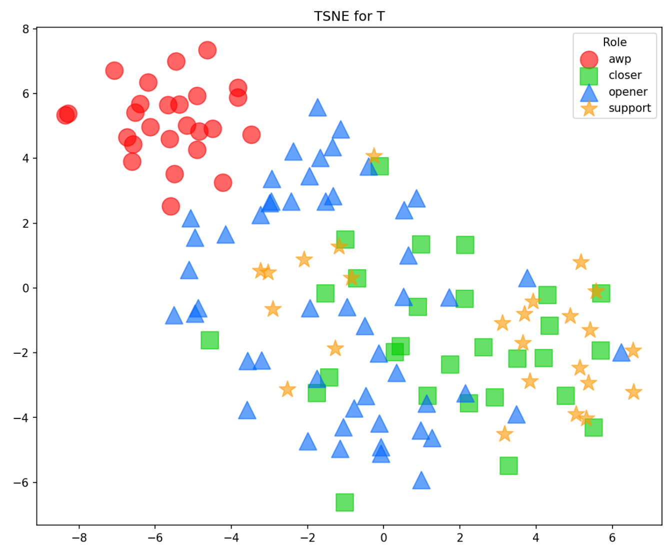


Figure 17: T-SNE visualization of the original feature space (T).

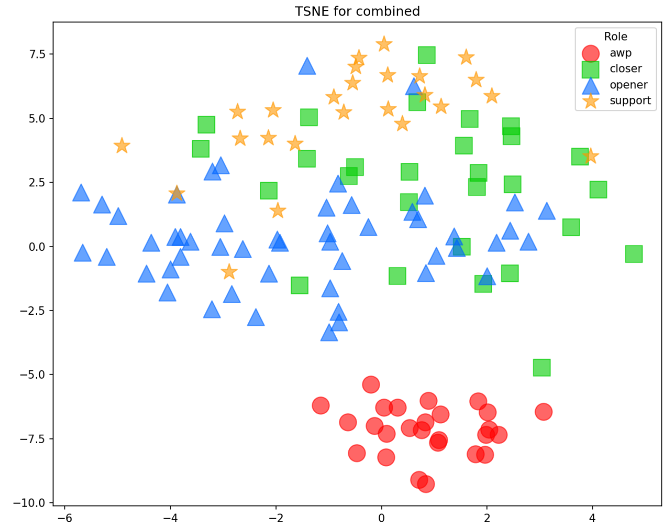


Figure 18: T-SNE visualization of the original feature space (combined).

5.3.2 Counter-Terrorist Features

Model	t-SNE (mean macro F1)	original (mean macro F1)	t-statistic	p-value
KNN	0.586 ± 0.058	0.621 ± 0.061	-2.911	0.005
XGBoost	0.562 ± 0.059	0.672 ± 0.075	-8.290	<0.0001
Random Forest	0.569 ± 0.069	0.709 ± 0.079	-9.557	<0.0001

Table 9: Results of pairwise t-tests between original and t-SNE feature space with different models (CT).

5.3.3 Terrorist Features

Model	t-SNE (mean macro F1)	original (mean macro F1)	t-statistic	p-value
KNN	0.593 ± 0.072	0.608 ± 0.08	-0.938	0.352
XGBoost	0.569 ± 0.068	0.639 ± 0.076	-5.330	<0.0001
Random Forest	0.608 ± 0.074	0.645 ± 0.087	-2.242	0.03

Table 10: Results of pairwise t-tests between original and t-SNE feature space with different models (T).

5.3.4 Combined Features

Model	t-SNE (mean macro F1)	original (mean macro F1)	t-statistic	p-value
KNN	0.727 ± 0.074	0.719 ± 0.077	0.714	0.479
XGBoost	0.73 ± 0.081	0.728 ± 0.080	0.145	0.886
Random Forest	0.738 ± 0.074	0.770 ± 0.069	-3.214	0.002

Table 11: Results of pairwise t-tests between original and t-SNE feature space with different models (combined).

5.4 Comparing Supervised Models

5.4.1 Counter-Terrorist Features

Below, the results of the Friedman test combined with the post-hoc Nemenyi test are shown for the CT-sided feature subset.
Table 12 reports the mean macro F1 score for each model, Table 13 the mean rank. Table 14 shows the Nemenyi p-value matrix, while Figure 19 shows the critical difference diagram.

Model	mean macro F1 score
KNN	0.621 ± 0.061
XGBoost	0.672 ± 0.075
Random Forest	0.709 ± 0.079

Table 12: Mean and Std macro F1 score for Friedman test (CT).

Model	mean Rank
KNN	2.56
XGBoost	1.90
Random Forest	1.54

Table 13: Mean Rank for Friedman test (CT).

The Friedman test indicates a significant difference with a Friedman statistic of **26.76** and a p-value of <**0.0001**.

Model	KNN	XGBoost	Random Forest
KNN	1	0.002780	0.000001
XGBoost	0.002780	1	0.169555
Random Forest	0.000001	0.169555	1

Table 14: Nemenyi p-value Matrix (CT).

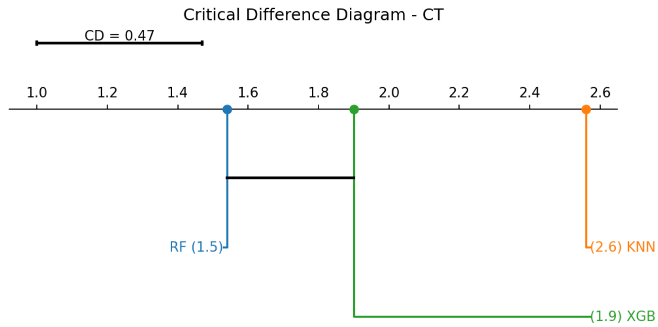


Figure 19: Critical Difference Diagram for Friedman Test (CT).

5.4.2 Terrorist Features

Below, the results of the Friedman test combined with the post-hoc Nemenyi test are shown for the T-sided feature subset.
Table 15 reports the mean macro F1 score for each model, Table 16 the mean rank. Table 17 shows the Nemenyi p-value matrix, while Figure 20 shows the critical difference diagram.

Model	mean macro F1 score
KNN	0.608 ± 0.08
XGBoost	0.639 ± 0.076
Random Forest	0.645 ± 0.087

Table 15: Mean and Std macro F1 score for Friedman test (T).

Model	mean Rank
KNN	2.32
XGBoost	1.86
Random Forest	1.82

Table 16: Mean Rank for Friedman test (T).

The Friedman test indicates a significant difference with a Friedman statistic of **7.72** and a p-value of **0.021**.

Model	KNN	XGBoost	Random Forest
KNN	1	0.055792	0.033242
XGBoost	0.055792	1	0.978190
Random Forest	0.033242	0.978190	1

Table 17: Nemenyi p-value Matrix (T).

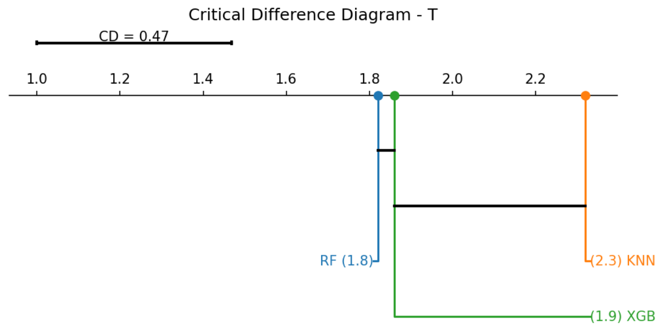


Figure 20: Critical Difference Diagram for Friedman Test (T).

5.4.3 Combined Features

Below, the results of the Friedman test combined with the post-hoc Nemenyi test are shown for the T-sided feature subset.
Table 18 reports the mean macro F1 score for each model, Table 19 the mean rank. Table 20 shows the Nemenyi p-value matrix, while Figure 21 shows the critical difference diagram.

Model	mean macro F1 score
KNN	0.719 ± 0.077
XGBoost	0.728 ± 0.080
Random Forest	0.770 ± 0.069

Table 18: Mean and Std macro F1 score for Friedman test (Combined).

Model	mean Rank
KNN	2.26
XGBoost	2.16
Random Forest	1.58

Table 19: Mean Rank for Friedman test (Combined).

The Friedman test indicates a significant difference with a Friedman statistic of **13.61** and a p-value of **0.001**.

Model	KNN	XGBoost	Random Forest
KNN	1	0.871308	0.001948
XGBoost	0.871308	1	0.010429
Random Forest	0.001948	0.010429	1

Table 20: Nemenyi p-value Matrix (Combined).

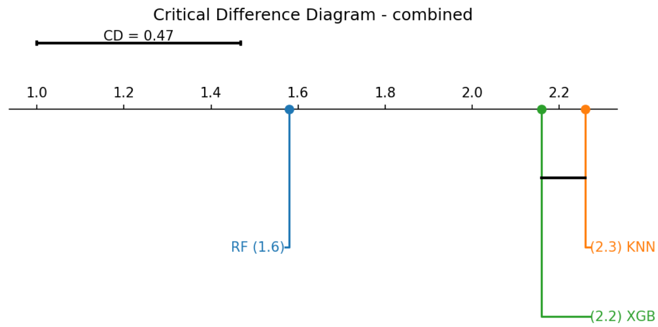


Figure 21: Critical Difference Diagram for Friedman Test (Combined).

6 Discussion

6.1 Analyzing feature importance

6.1.1 Analyzing important features for AWP players

For AWPers, Figures 1, 2, 3 show that the immobility ratio in the early and mid game is an important feature for all feature subsets. A high immobility ratio has a positive impact on the model's output, while low values tend to have lower or negative impacts.
This is consistent with the observed behavior of AWPers in CS2. The AWP slows a player down and is heavily inaccurate during movement. Therefore, it was expected to see the immobility ratio being an important feature when classifying AWPers.
It is important to note that the T-sided feature subset only values the early game immobility ratio as important. Again, this is consistent with the nature of CS2. On T-side the team has to be a lot more active and cover more distance, while the CT-side can be more static, just covering their bombsites.
Spatial features such as mean entropy to the team or distance to the team centroid also have a high impact on the model's output. AWPers tend to consistently move close to the team's centroid, especially in the early game.
Again, this is expected. As mentioned above, the AWP makes you immobile, requiring team mates nearby to help you get good angles. The distance to the team's centroid only plays a role for T-side. This makes sense considering that CT-side players commonly have priority over T-side players when holding a choke point. Therefore, AWPers on CT-side do not need as much help. This is also supported by the fact that a low entropy relative to the team in the early game contributes positively to the models output.
The last important feature for CT-side, the mean high-explosive grenades thrown per round, probably would not have the same impact in earlier versions of Counter-Strike. With the release of CS2, the game allows for smoke grenades (used to obscure vision) to be blown open with high-explosive grenades. Especially AWPers are using this to their advantage by using high-explosive grenades to reopen the line of sight in crucial moments. Therefore, the feature's importance is not a surprise. Interestingly, the feature does not make the top five for the T-sided subset, as T-side AWPers find themselves less likely to be obscured by smoke grenades.

6.1.2 Analyzing important features for opener players

As with other roles, the standard deviation to the team's centroid has an important impact on the model when predicting the role opener, as seen in Figures 4,5, and 6. Here, mid to low values have a positive impact on the model's output. Additionally, Figures 5 and 6 show that the mean distance to the team centroid is also an important feature. Again, mid to low values have a positive impact. Consequently, the opener role stays close to the pack of the team consistently over the early and mid game. This is consistent with the observed behavior of openers. Since opening is risky, the opener needs teammates nearby who can throw utility or trade the openers death by eliminating the opponent.

Furthermore, Figure 4 shows a surprising result. A high usage of grenades, especially flash and high-explosive grenades, has a significant impact on the model's output for the CT-sided feature subset. For the T-sided feature subset, features revolving around grenade usage do not appear in the top 5. Therefore, the opener role significantly differs between the two sides. The opener role is more distinctive on the T-side, which can be seen by the fifth feature in Figure 5. A mid to low mean time to the first engagement has a positive impact on the model's output. This aligns with the common definition of the opener, as the role is the first to take contact in the team when entering bombsites or choke points.

6.1.3 Analyzing important features for support players

The Figures 7, 8, and 9 suggest that support players are mainly classified by standard deviations of spatial features such as the distance to the team centroid or the entropy relative to the team. Regarding the CT-sided feature subset, the high to mid values for the standard deviation of the distance to the team centroid in early and mid game are having a high impact on the model. It follows that support players have high variance regarding their positioning relative to the team. They take positions close to the main group of the team while also being able to position themselves at the extremities of the map.

Additionally, for the T-sided feature subset, the standard deviation and mean of the entropy relative to the team in the early game play an important role. Again, higher values have a higher positive impact on the prediction. This is consistent with the conclusion above, as the results say that supporting players are positioning themselves independently from their team in the early game, while the standard deviation indicates that it differs greatly between maps. Furthermore, the support playstyle on T-side can be classified by the mean time to first engagement, with higher values providing positive impact on the output. This is consistent with the definition of the role. Support players provide value by setting up teammates with utility and joining the fight after their opener took space.

When analyzing the feature importance of the combined feature subset, the top five features are similar. An important observation is that CT-sided features are dominating T-sided features, suggesting that support players have a more distinctive playstyle on CT-side. Additionally, the standard deviation of team centroid features shows more impact than the mean.

6.1.4 Analyzing important features for closer players

Overall, the closer role is the hardest role to classify for the models. This can also be seen in Figures 10, 11, and 12. Many of the top five most important features have a high variance in their

values, e.g. low to high values contribute equally to the model's output.
For the CT-sided feature subset, the most important feature appears to be mean grenades thrown per round, where mostly mid to high values have a positive impact. Low values can have a slightly positive impact. The reason for this could be the fact that closers usually play in anchor positions on bombsites. Therefore, closers tend to invest into utility to stop the enemies from taking the bombsite.
Another important insight is the difference in feature values of the immobility ratio between early and mid game for the CT-sided feature subset. Mid to high values have a higher impact on the model for the early game. Conversely, for the mid game, mid to low values are more dominant. It follows that closer players like to be static at one position during the early phase, while the mid game forces them to be mobile as they need to adjust their positioning (rotations).
Figure 11 indicates that spatial features dominate the model's output for the T-sided feature subset, especially the distance to the team centroid. Mostly mid to high values for standard deviation and mean have a positive impact on the models output. This is consistent, as the closer role often lurks on the T-side further away from their teammates. Additionally, it can be seen that in the mid game the low to high values for the mean contribute positively to the model's output. Again, this is consistent with observed behavior, as during the mid game, teams tend to group up to hit a bombsite together.
The combined feature subset shows similar results to the CT-sided and T-sided subsets. However, features from the early game are the most influential to the model's output. This indicates that closer roles can mostly be classified by behavior early in the game.

6.2 KNN-based Player Replacement

Across the different feature subsets, the KNN model suggests similar and different players as seen in Figures 13,14, and 15. The player that is recommended the most is **Monesy** with ranks of 1, 2 and 1.
While some players are recommended for all feature subsets, the ranking order often differs. This indicates that playstyles can differ between sides.
Several of the recommended players are AWP players such as **ZywOo**. This shows that the feature space captures AWP stylistic behaviors well. However, KNN also recommends players with other roles such as closer and opener. This indicates that the feature space captures stylistic similarities that are not immediately related to roles.
Lastly, this experiment would suggest **Monesy** as the best replacement for **ZywOo** based on playstyle. This is a good replacement due to role matching. Additionally, both players are widely regarded as star players. They are known for exceptional plays, as well as a calculated aggressiveness.
It is important to mention certain limitations. First of all, the replacement suggestion is purely based on behavioral playstyle. It ignores other important factors in e-sports such as communication, attitude, language and experience. Additionally, KNN could suffer from the curse of dimensionality, especially with the large combined feature subset.
Overall, the results indicate that the feature space captures playstyles at a rate where machine learning methods can be used to give a selection of players. The e-sports team could then conduct further analysis on the players to find the best possible replacement.

6.3 Comparing Feature Spaces

As seen in the t-SNE visualizations 16, 17, 18, the AWP role is quite well separated. Furthermore, the rifle roles (opener, closer, support) are clustered together. A slight separation between the support and opener roles is visible, while the closer role has a high spread.
The results of the pairwise t-test to compare the original feature space and the t-SNE feature space tell a mixed story. For the CT-sided subset (Table 9), the original feature space performs significantly better with all models. With the T-sided subset (Table 10), the KNN model already shows no statistically significant difference, while XGBoost and Random Forest are still performing better with the original feature space. With the combined feature subset (Table 11), only the Random Forest finds a statistically significant difference in favor of the original feature space.
The results indicate that t-SNE is more suitable for visualization of role structure than as a feature representation for downstream prediction tasks. Overall, it falls short against the original feature space, especially when using Random Forest as the classifier.

6.4 Comparing Supervised Models

The three supervised machine learning models KNN, Random Forest and XGBoost were evaluated based on their performance of predicting player roles given their behavioral features. The Friedman test was used in combination with the post-hoc Nemenyi test.
Firstly, the Friedman test indicates a statistically significant difference for all feature subsets. This means that the performance of at least two of the three models are significantly different.
The post-hoc Nemenyi test (Table 14) shows that the Random Forest model and XGBoost model are performing the best at predicting player roles with CT-sided features. However, Table 13 shows that the average rank of Random Forest is better than that of XGBoost.
Furthermore, the results for T-sided features are slightly different. While the Random Forest model performs significantly better than KNN, XGBoost does not perform significantly better anymore as seen in Figure 20.
On the contrary, when using the combined feature subset, Random Forest performs significantly better than both XGBoost and KNN. This can be seen in Table 20 and Figure 21.

7 Conclusions and Further Research

7.1 Conclusions

The aim of this thesis was to research how playstyles can be classified in online multiplayer video games. The underlying goal was to provide a framework that can be used by analysts to quickly identify player replacements.
Regarding the research question, the feature importance analysis showed that spatial-temporal features such as the distance to the team centroid dominated. However, temporal features also had a positive impact on the model's output for a selection of roles. Additionally, the standard deviations of spatial-temporal were more important to the model's output than the means of the same feature. Conversely, for temporal features the mean had a larger weight. Features computed for the early game and mid game were more useful than features for the after-plant phase.

With respect to SQ1, the results showed that the created feature space captures meaningful similarities in player playstyles. A KNN-based player replacement approach allows analysts to get a selection of players that show similar playstyles to the replaced subject. Furthermore, the results suggest that behavioral tendencies differ between CT- and T-side, as the KNN recommendations vary between feature subsets.
The answer to SQ2 is mixed, as the results state that for the Random Forest model the original feature space performs significantly better for all feature subsets. However, the XGBoost model performs better with the original feature space only for the CT-sided and T-sided feature subset. The KNN model only performs better with the original feature space on the CT-sided subset. Therefore, the answer to SQ2 depends on the model and feature subset. It can be concluded that while the t-SNE feature space still retains some information about players' playstyles, it is better used for visualization than classification tasks.
Regarding SQ3, the results indicate that the Random Forest model performs significantly better on the combined feature subset than the XGBoost or KNN. However, for the CT-sided and T-sided subset, random forest does not outperform XGBoost significantly. Also, only on the CT-sided subset does XGBoost outperform KNN. Overall, these findings suggest that Random Forest is the most robust model across feature configurations, while KNN generally shows lower performance in comparison.

7.2 Further Research

Although the thesis created promising results, there are several opportunities for further research. First, the overall dataset can be extended. By pulling in more matches, the final number of players who meet the preprocessing requirements would grow. Therefore, the models would have more data to learn from.
Regarding the player replacement experiment, further research could involve incorporating non-behavioral features into the feature space. Performance-based features are especially interesting.
Additionally, one could look into using different labels. The thesis makes use of original labels defined by experts and teams. However, analysts have found ways of implementing new labels that take into consideration how many resources a player gets inside the game.
Furthermore, future work could involve the use and comparison of different visualization algorithms such as UMAP or PCA. Adding other models, such as deep neural networks, to the model comparison could also prove useful for identifying the best model for classifying play styles.
Finally, the thesis proposes a framework for online multiplayer video games. It is evident that in the future, researchers could apply this framework to other games and report the performance.

References

[1] Dr. Mitra Baratchi. Ensemble models and machine learning experiments. Lecture notes from Machine Learning, Universiteit Leiden, April 2025.

[2] Monica Benjamin, Sreedeep S S, R. Dhanalakshmi, Arulmozhi P, and Hema Mahalingam. Best fit player acquisition and squad personnel system in football for sports. In *2022 1st International Conference on Computational Science and Technology (ICCST)*, pages 1013–1018, 2022.

[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] Hao Chen, Pengfei Qian, Lin Lv, and Guoheng Zhang. Tsne+k-means based heritage classification system. In *2023 International Conference on Applied Physics and Computing (ICAPC)*, pages 26–31, 2023.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[6] Padraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers: 2nd edition (with python examples). *CoRR*, abs/2004.04523, 2020.

[7] Tom Decroos and Jesse Davis. Player vectors: Characterizing soccer players' playing style from match event streams. In *Machine Learning and Knowledge Discovery in Databases*, volume 11908 of *Lecture Notes in Computer Science*, pages 569–584. Springer, Cham, 2020.

[8] Ege Demir, Yusuf H. Şahin, and Nazım Kemal Üre. How do football teams play? a deep embedded clustering approach to reveal playing styles. In Jin-song Dong, Jing Sun, Xiaofei Xie, and Kan Jiang, editors, *Sports Analytics*, pages 53–68, Cham, 2026. Springer Nature Switzerland.

[9] Romain Hérault, David Orth, Ludovic Seifert, et al. Comparing dynamics of fluency and inter-limb coordination in climbing activities using multi-scale jensen–shannon embedding and clustering. *Data Mining and Knowledge Discovery*, 31:1758–1792, 2017.

[10] HLTV. Hltv.org – counter-strike statistics and match demos. https://www.hltv.org, 2026. Accessed: 2026-02-18.

[11] Stephanie A. Kovalchik. Player tracking data in sports. *Annual Review of Statistics and Its Application*, 10:677–697, 2023.

[12] Yurii I. Kuzhii and Yuriy M. Furgala. Feature engineering for role assessment in counter-strike 2. *Electronics and Information Technologies*, 32:121–130, 2025.

[13] LaihoE. demoparser2: Counter-strike 2 demo parser. https://github.com/LaihoE/demoparser, 2026. Accessed: 2026-02-18.

[14] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[15] Mudit Mittal, Praveen Gujjar J, Guru Prasad M S, Raghavendra M Devadas, Lubna Ambreen, and Vikash Kumar. Dimensionality reduction using umap and tsne technique. In *2024 Second International Conference on Advances in Information Technology (ICAIT)*, volume 1, pages 1–5, 2024.

[16] Samuel J. Moffatt, Ritu Gupta, Suman Rakshit, and Brad S. Keller. Identifying team playing styles across phases of play: A user-specific cluster framework. In Jin Song Dong, Masoumeh Izadi, and Zhe Hou, editors, *Sports Analytics*, pages 129–136, Cham, 2024. Springer Nature Switzerland.

[17] José Pino-Ortega, Daniel Rojas-Valverde, Carlos D. Gómez-Carmona, and Markel Rico-González. Training design, performance analysis, and talent identification—a systematic review about the most relevant variables through the principal component analysis in soccer, basketball, and rugby. *International Journal of Environmental Research and Public Health*, 18(5):2642, 2021.

[18] Spyridon Plakias, Serafeim Moustakidis, Christos Kokkotis, Marina Papalexi, Themistoklis Tsatalas, Giannis Giakas, and Dimitrios Tsaopoulos. Identifying soccer players' playing styles: A systematic review. *Journal of Functional Morphology and Kinesiology*, 8(3), 2023.

[19] Harry Richards. Positions database ner0cs, 2026.

[20] Hao Wang, Hao-Tsung Yang, and Chuen-Tsai Sun. Thinking style and team competition game performance and enjoyment. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(3):243–254, 2015.

[21] Franziska Zimmer, Mhd Irvan, Maharage Nisansala Sevwandi Perera, Ryosuke Kobayashi, and Rie Shigetomi Yamaguchi. Fair play and identity: In-game behavioral biometrics for player identification in competitive online games. In *2025 IEEE Conference on Games (CoG)*, pages 1–8, 2025.