Gaming Data Science - Jako Rostami

Cult of the North

Agenda:

Dataset usage

CS-GO Competitive Matchmaking: Data Science

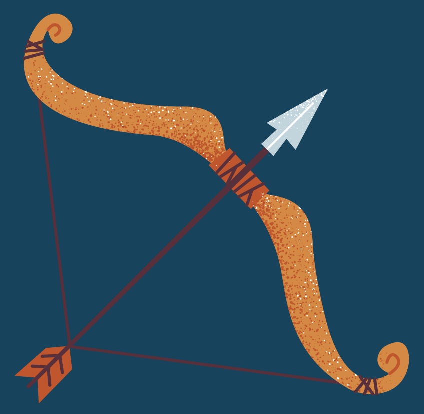
League of Legends: Machine Learning

Bonus: Predicting latitude-longitude with LSTMs

Objectives:

Examine the different use cases

Provide a versatile demonstration of how mathematical methods can be used for different games



Train a model!

Dataset usage

Since there are a variety of datasets from different games, the datasets chosen, w.r.t to the available time, are two of the most popular games. CS-GO and League of Legends.

For the CS-GO dataset it is from competitive matchmaking by extracting demo's of the matches. They consist of event-recorded datapoints which means that only if an event happened will there be data registered.

In the other dataset, League of Legends Diamond matches with the first 10 minutes for both teams. This one is different since it only summarizes the data of the first 10 minutes.

Dataset: CS-Go

Time

Events are recorded in *n* seconds into the match for both the attacker and the victim of the attacker.

Coordinates

Since this dataset has coordinates of the event, we'll be able to see the movement patterns of Terrorists and Counter-Terrorists on the game maps

Weapons

Another interesting part of the data is that we can map out the whole weapon attack pattern over time and over a local geographic area. Also for which specific weapon!

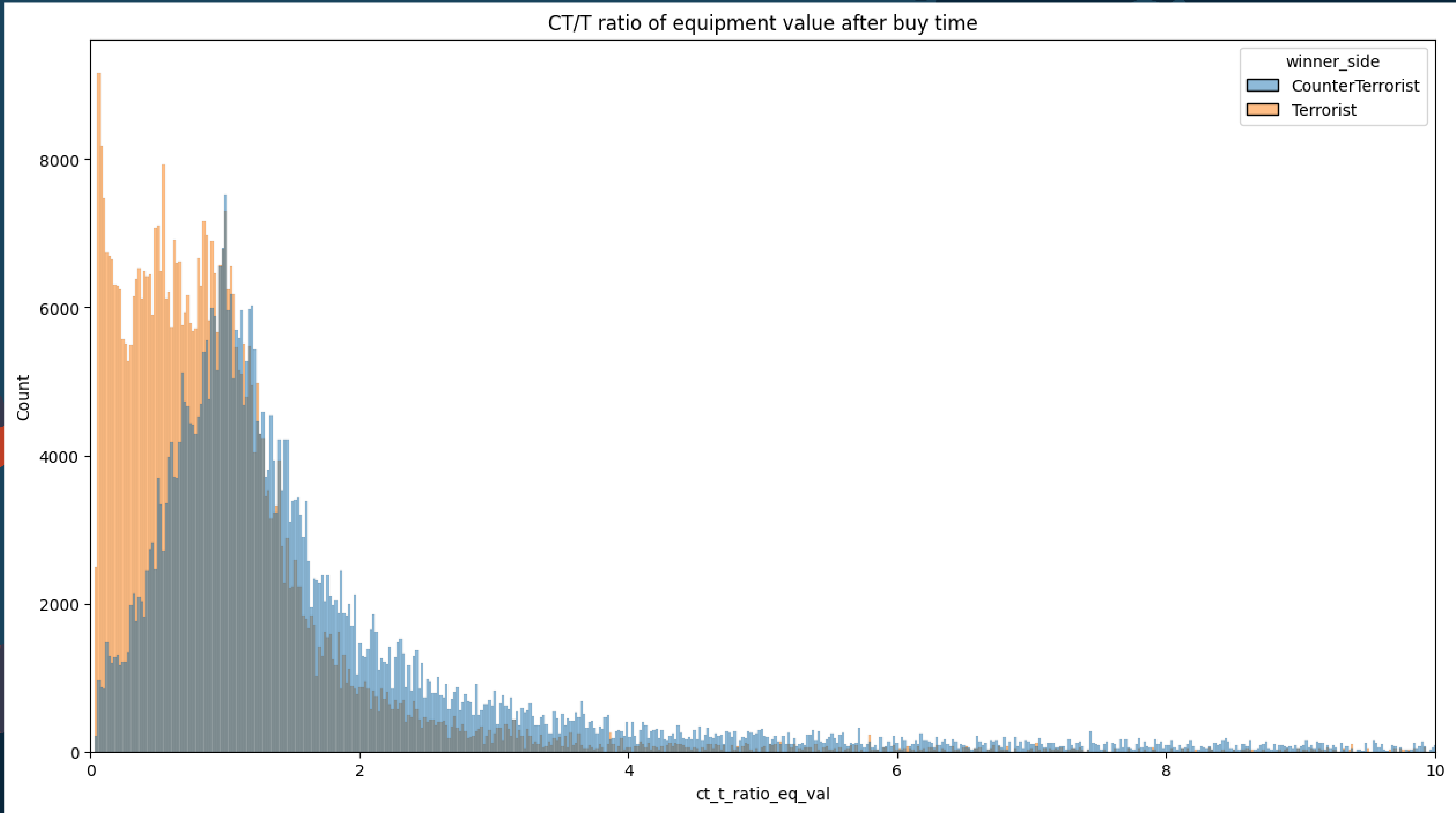
Dataset: LoL

Not as rich as the CS-GO dataset in terms of temporal and spatial events but enough to create machine learning models. We'll be discovering how these Diamond ranked games are extremely competitive as their patterns differ depending on if the other team is highly experienced.

CS-GO

What we'll look at first is the equipment value after buy time for each match and split them by which side won.

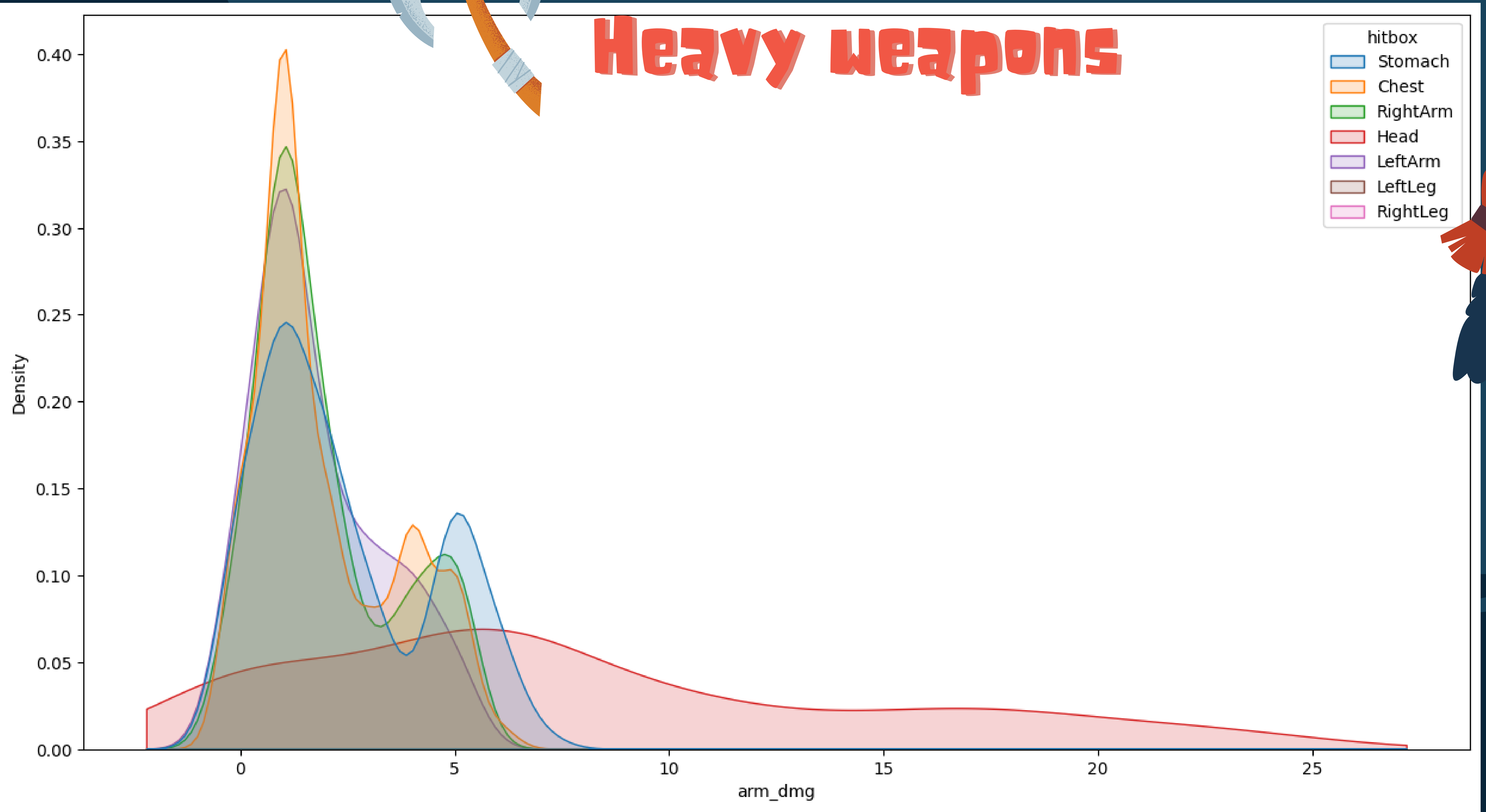
- Counter-Terrorists tend to buy more expensive gear
- Terrorists might be saving depending on the round



Interesting is that when the heavy weapons hit the head, it also tend to have a very large spread on the armor.

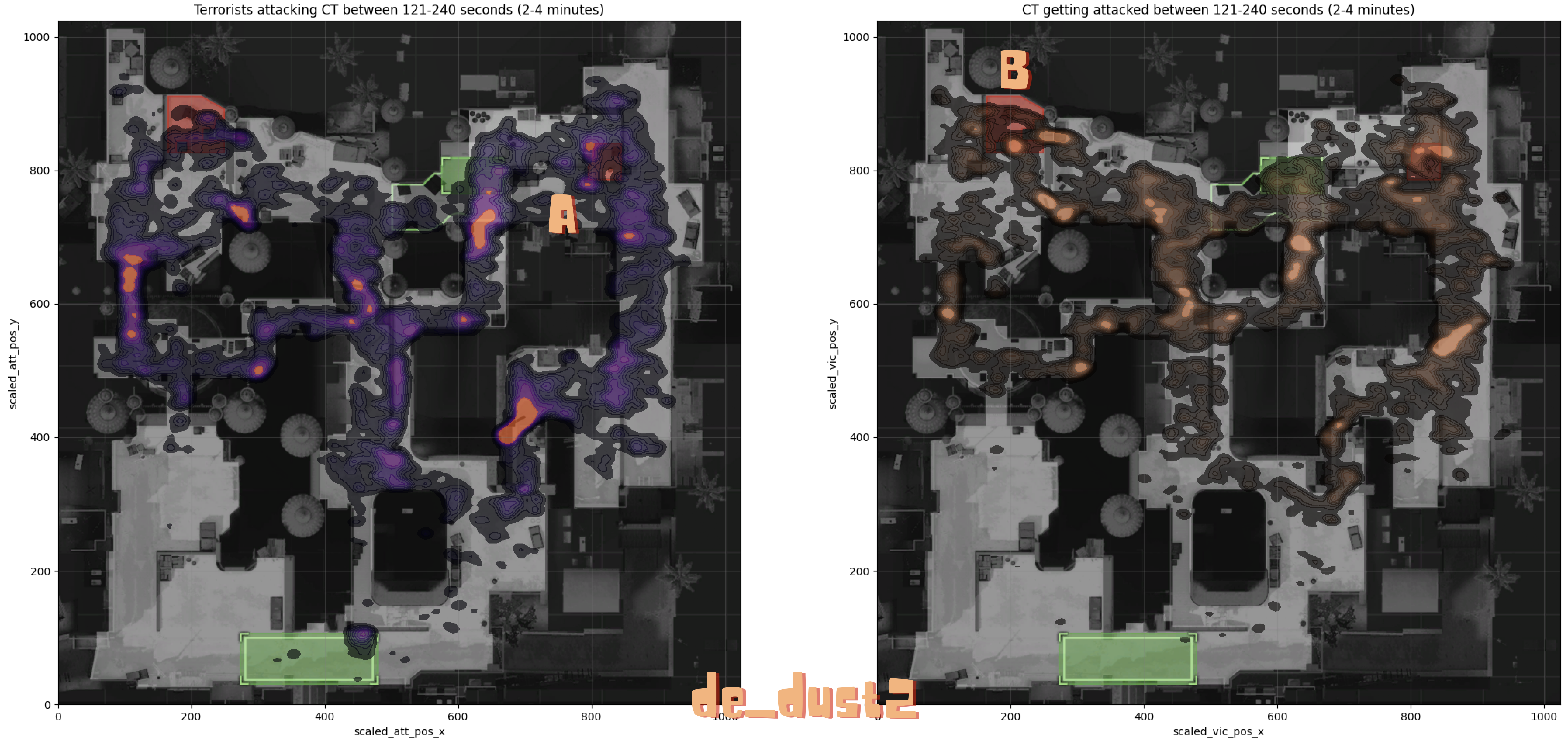
This could either indicate that heavy weaponry gets a damage boost on armor when hitting the head or vice versa.

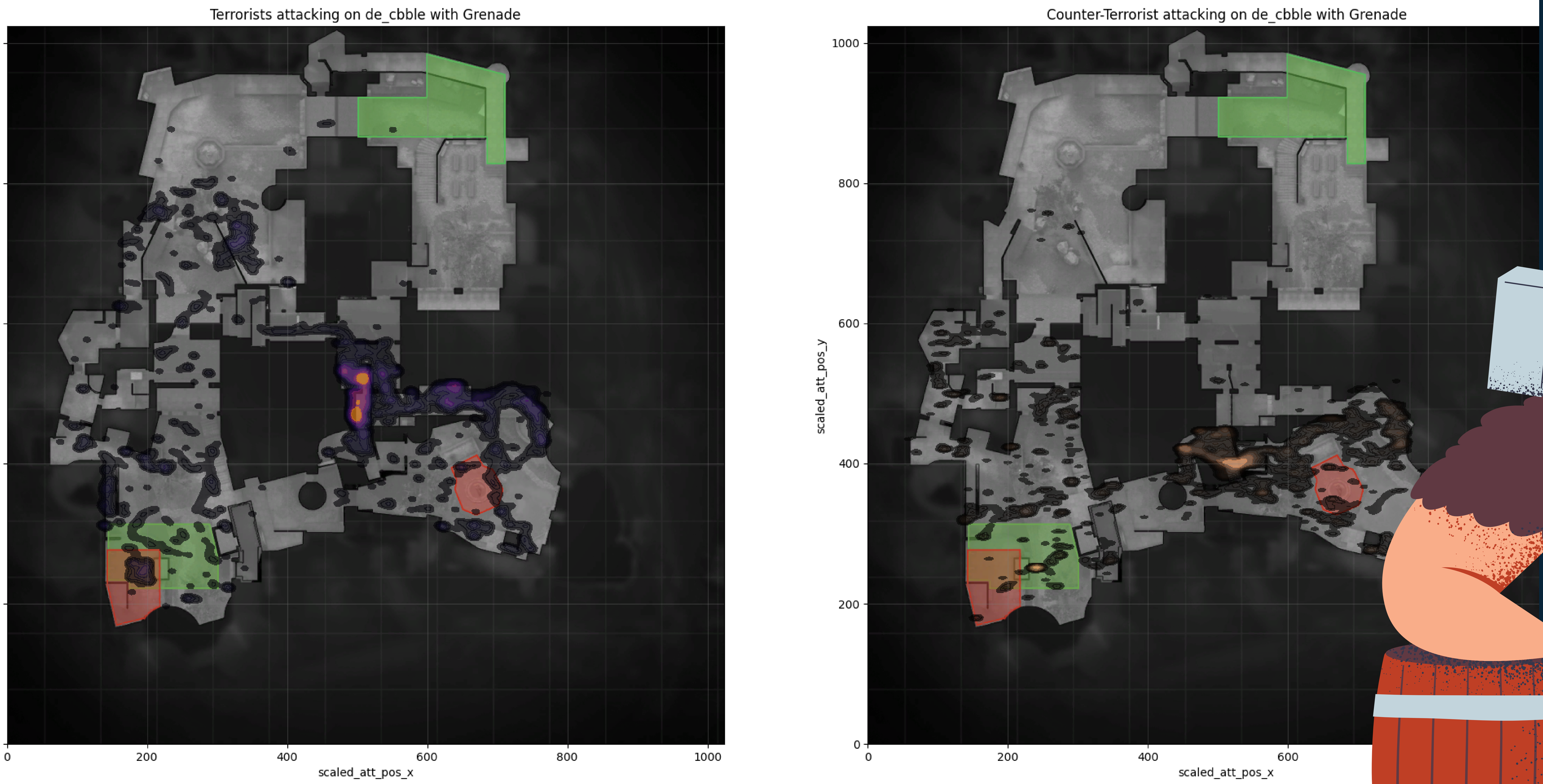
Or, it can indicate that the Kevlar has been heavily damaged after hitting the head and thus shows high variability in *arm_dmg*.



Attack patterns

Terrorists have a tendency to spread around bomb site A while Counter-Terrorists focus more on bomb site B and the center.



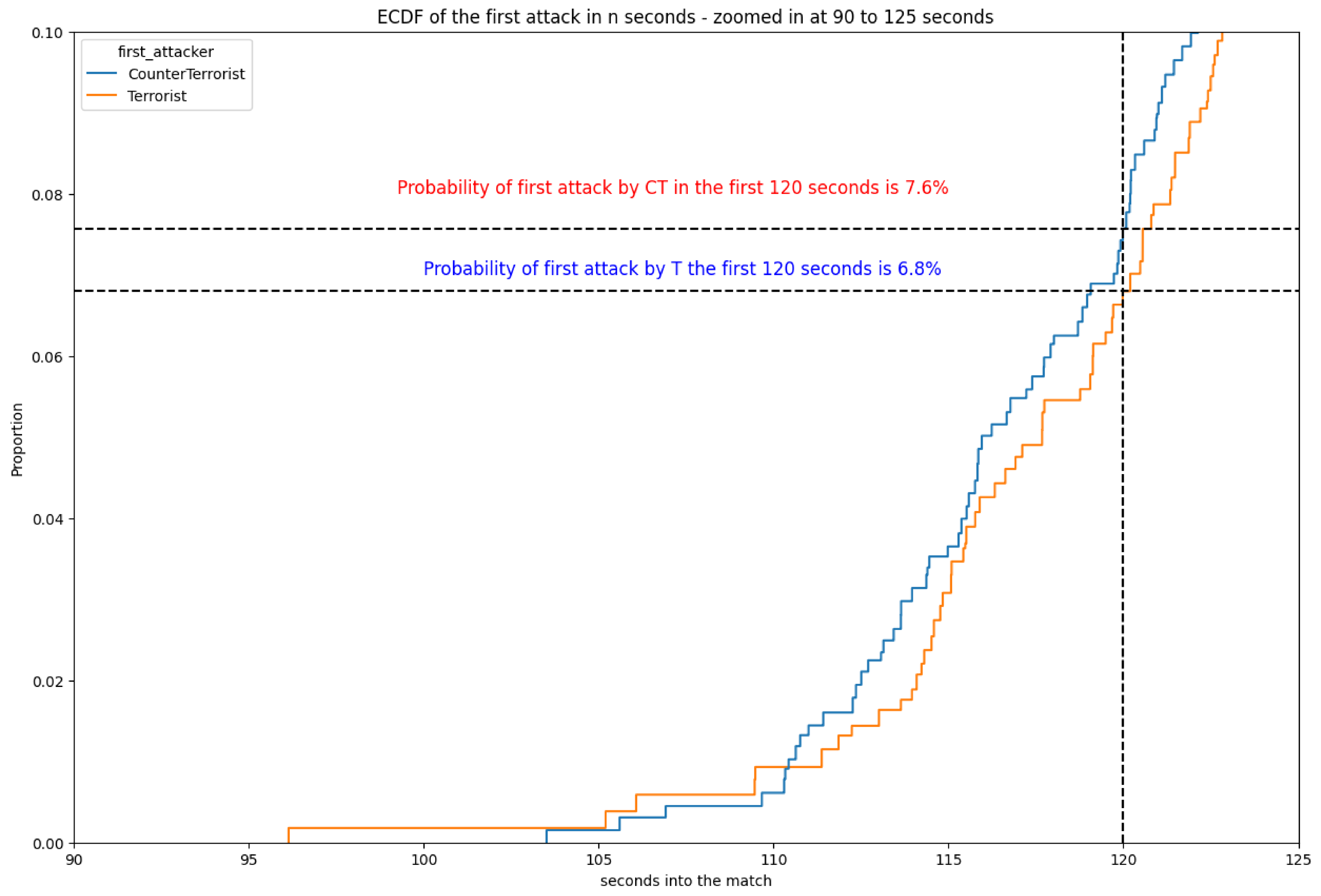


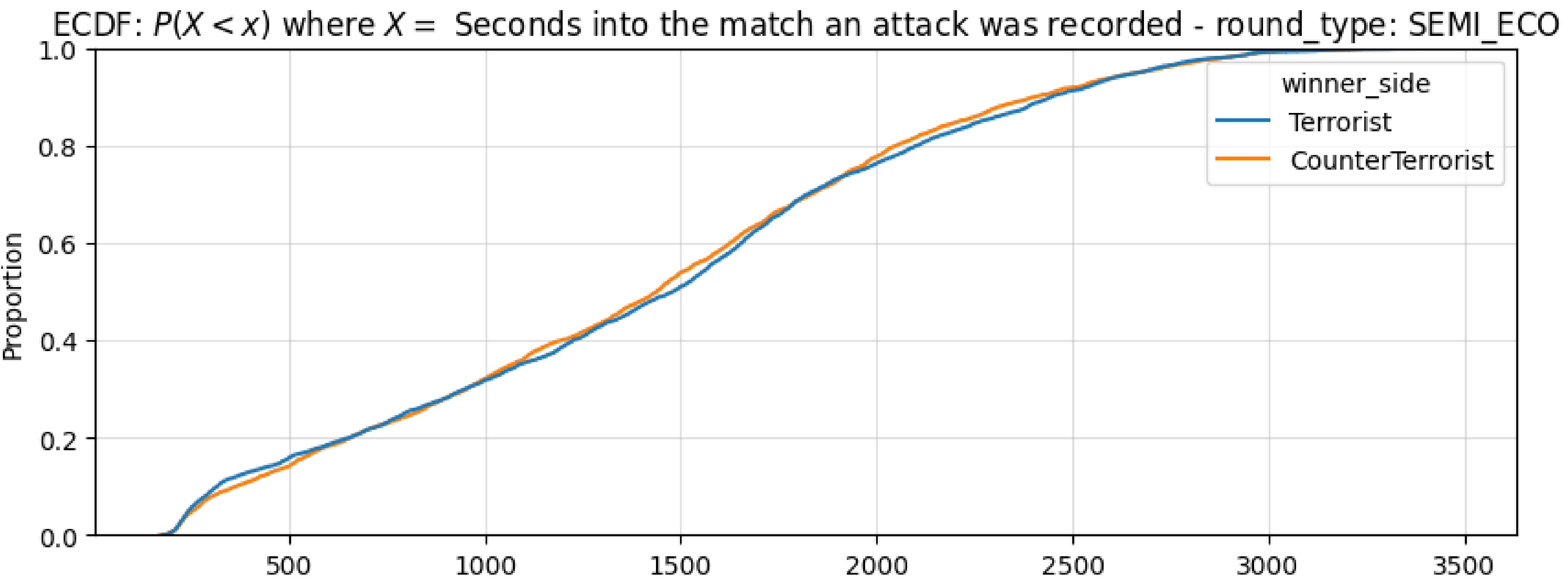
Grenade patterns

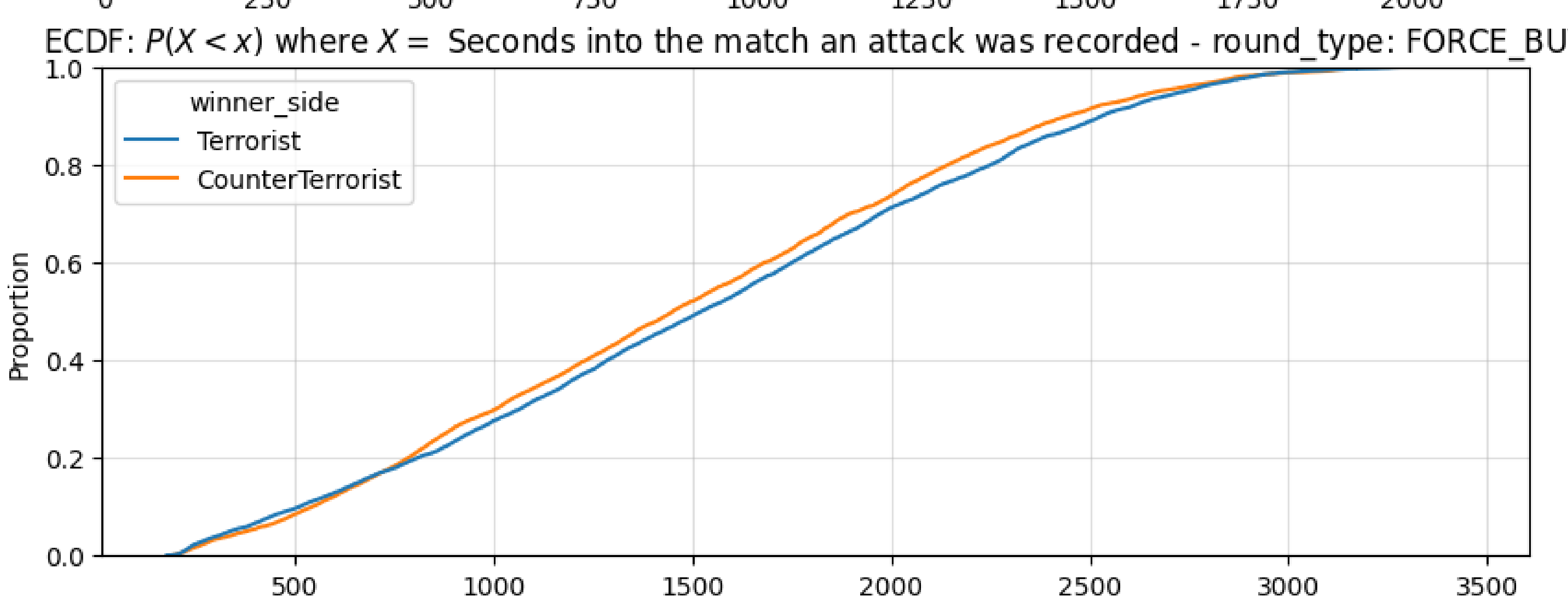
Terrorists cover a lot more of de_cbble compared to Counter-Terrorists when attacking with grenades!

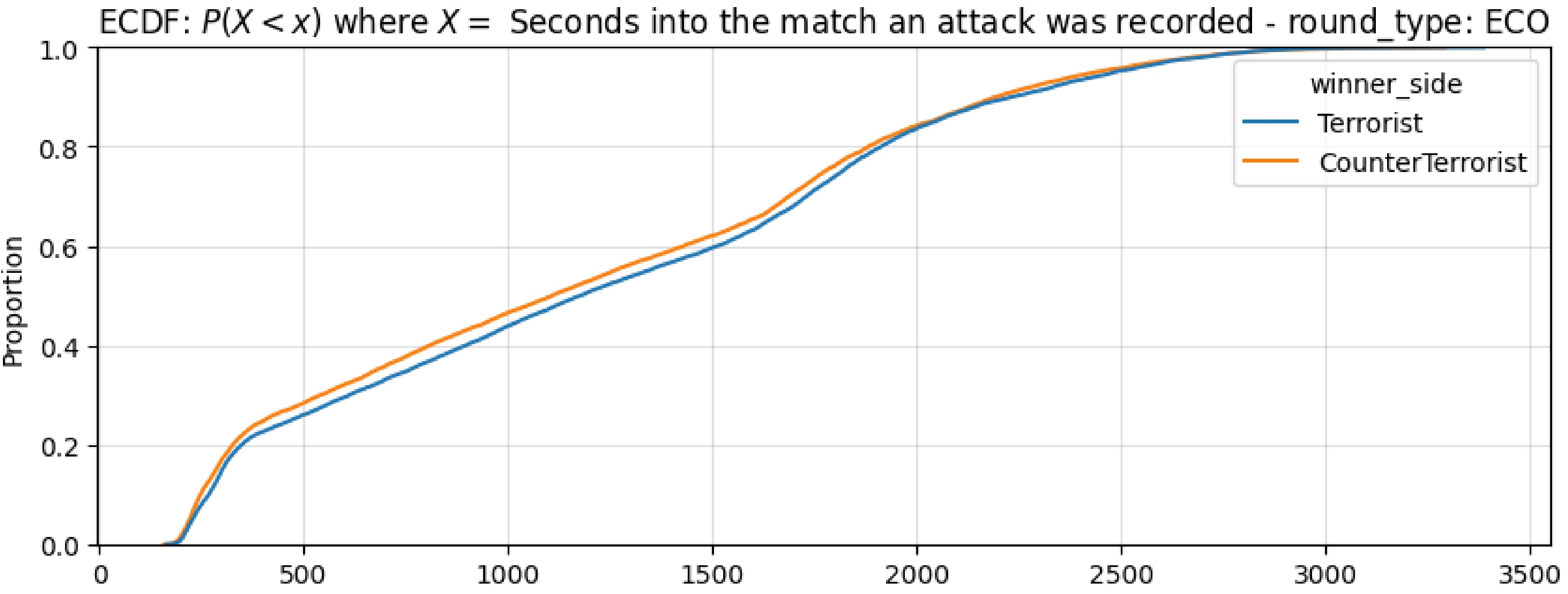
Probability of attack

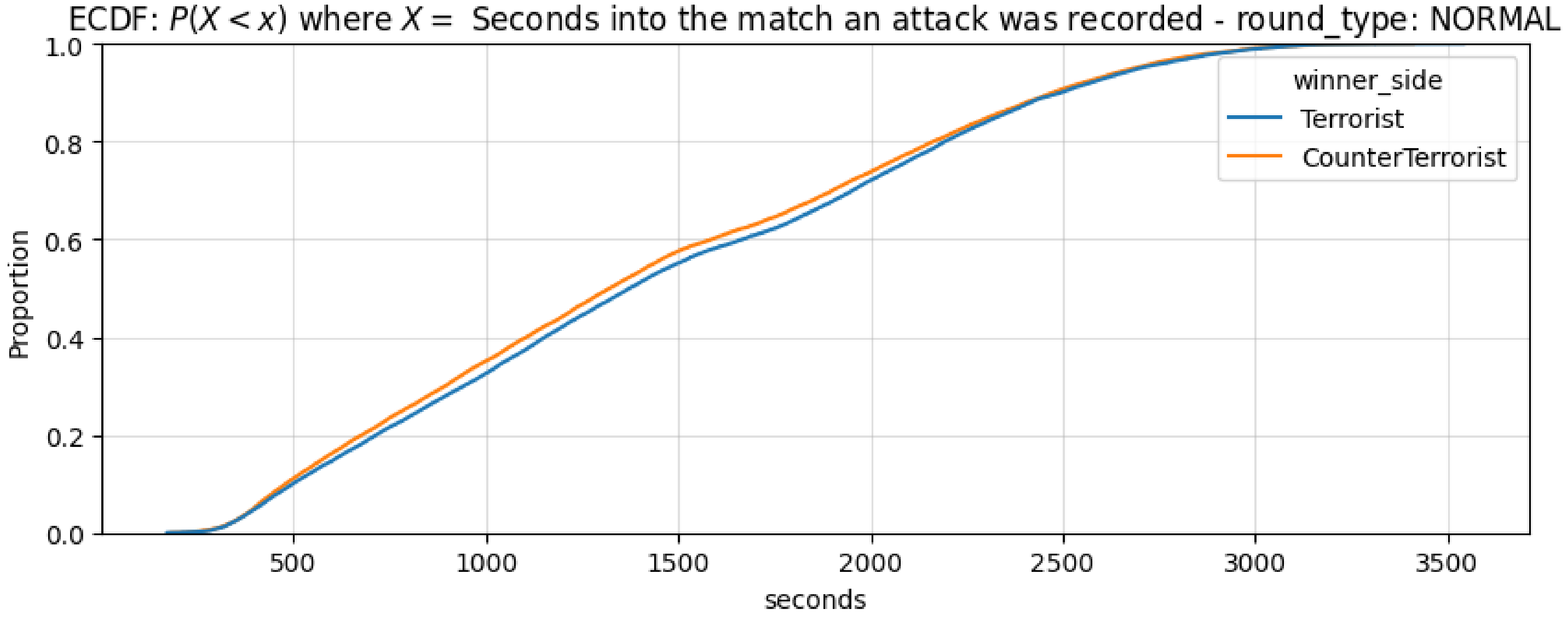
Of all the matches analyzed, we find that Counter-Terrorists have a higher propensity to be the first attacker.

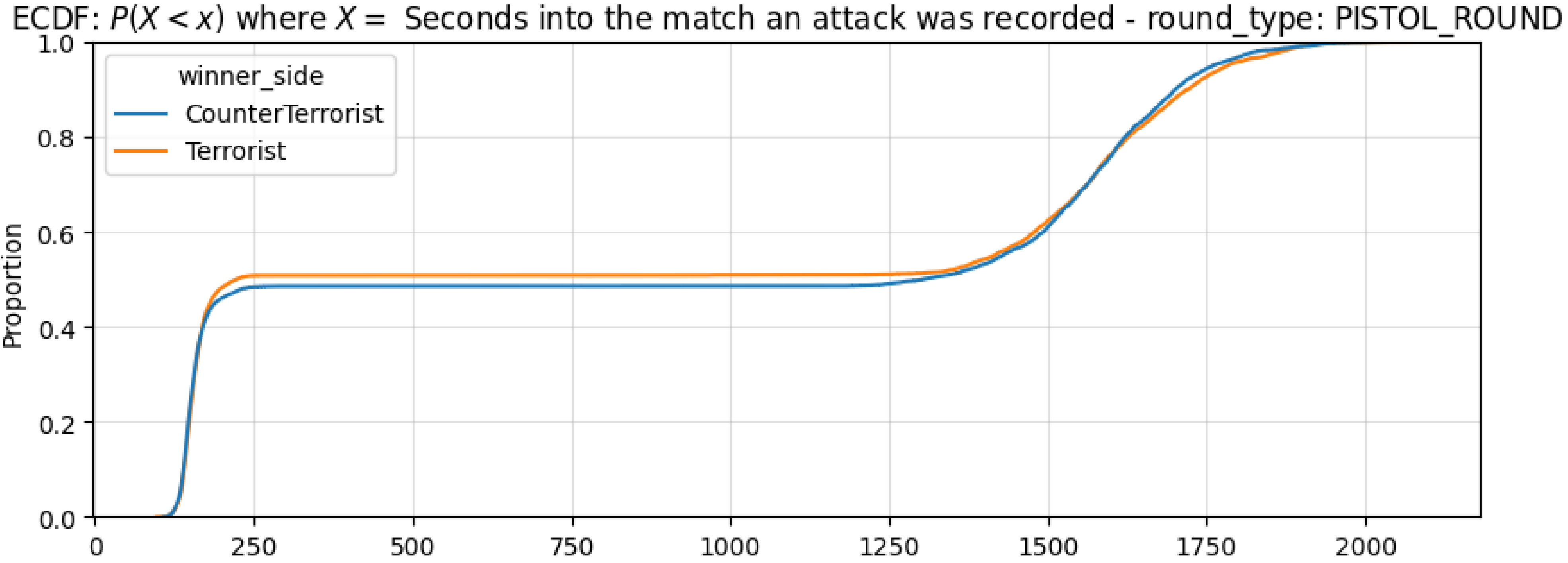






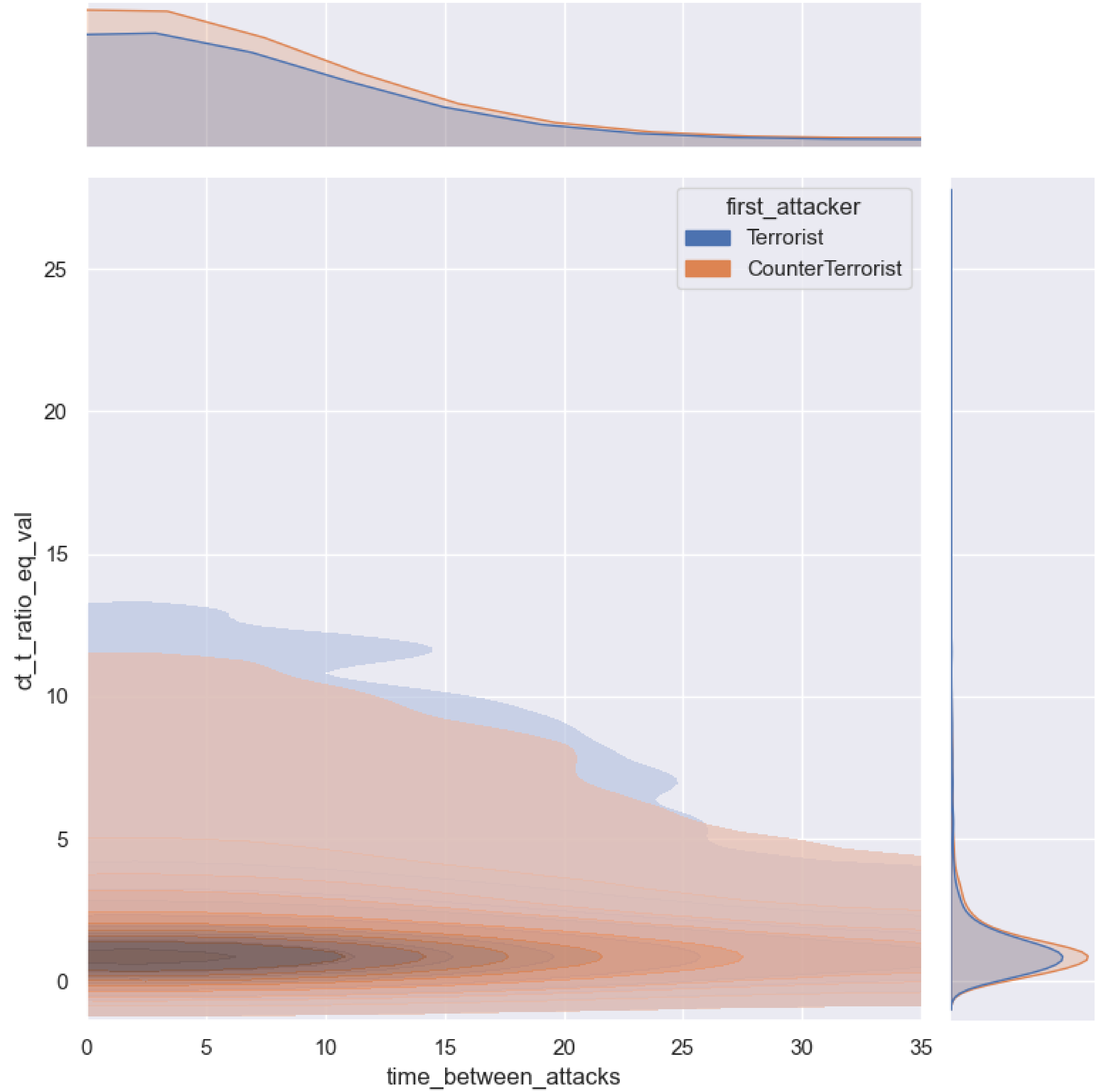






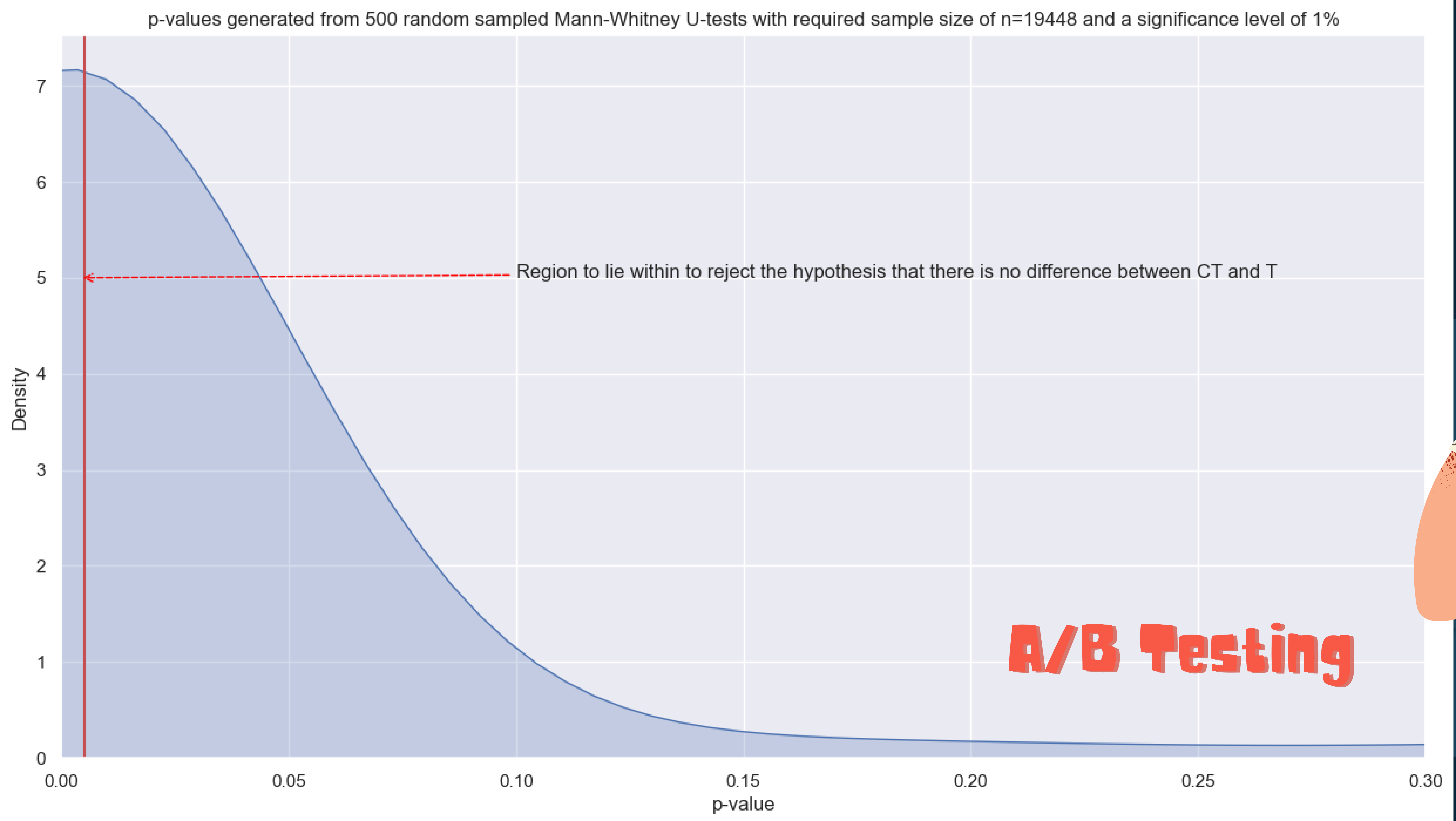
Aggressive push

Another interesting finding which depends on the round type of the match is that Counter-Terrorists tend to be pushing a lot in terms of how aggressive they are throughout the match or in the first minutes of the match. In 3 out of 5 round types we find that Counter-Terrorists have push dominance, Terrorists only for pistol rounds, and a tie for semi-eco rounds.



Time between attacks

By finding the time between attacks in seconds depending on whom the first attacker is, at a first glance it seems as if Terrorists have a lower time between attacks and higher equipment value after buy time if they are the first attacker.



But when we test this difference using a non-parametric statistical test 500 times to see if this happens by chance 1% of the time - the majority of our tests generate values that are compatible with our data. Meaning we don't really find a statistical difference.

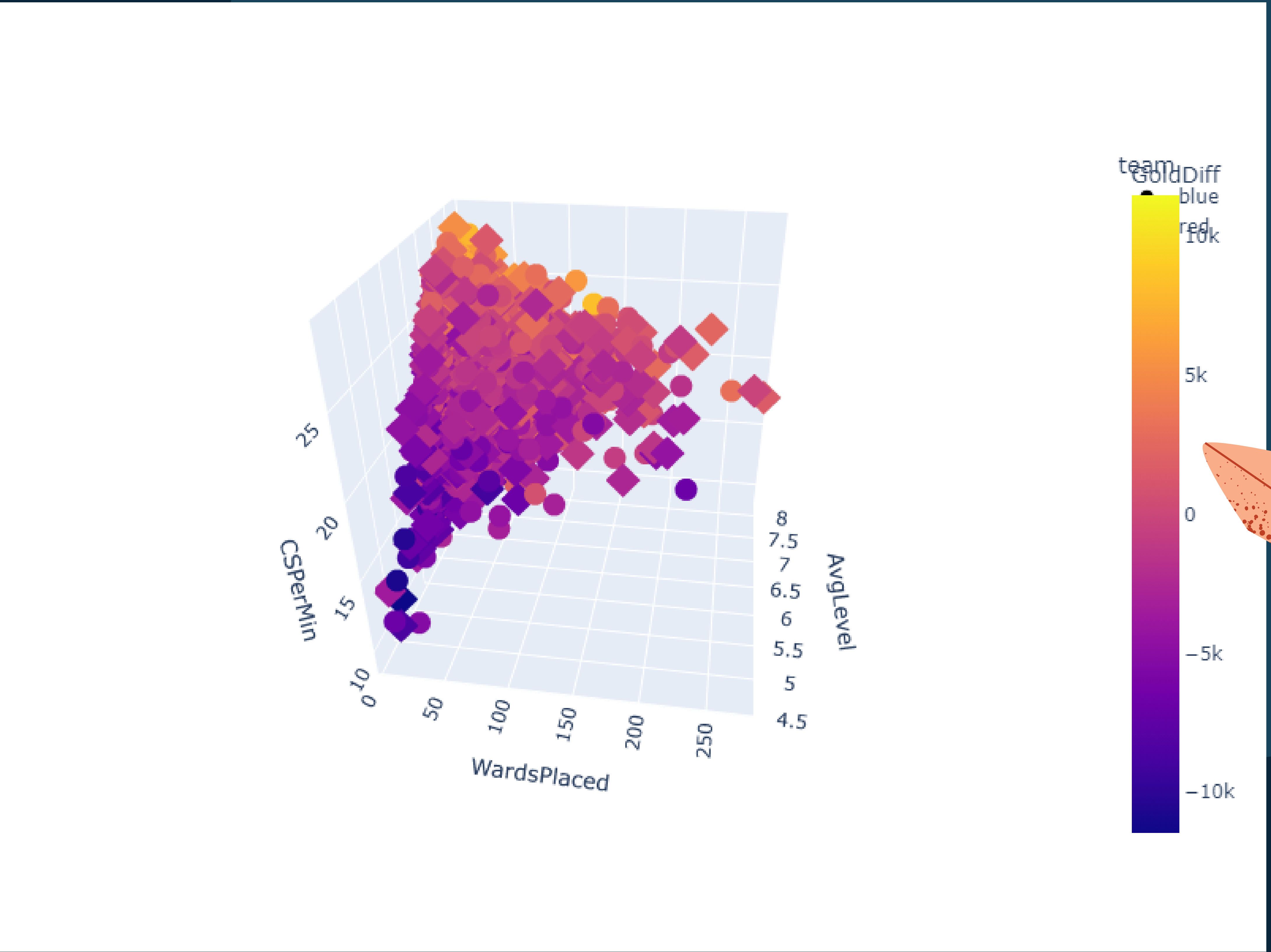
Wrap Up

- Counter-Terrorists have a higher push dominance than Terrorists in 3 out of 5 round types
- Counter-Terrorists are more likely to be the first attacker
- Terrorists spend less on equipment than Counter-Terrorists
- Terrorists cover the area around bomb site A on de_dust2 heavily while Counter-Terrorists cover the area around bomb site B heavily
- There is no statistical difference between the time of the attacks regardless of whom the first attacker is even though Counter-Terrorists are more aggressive and more likely to be the first to attack

League of Legends

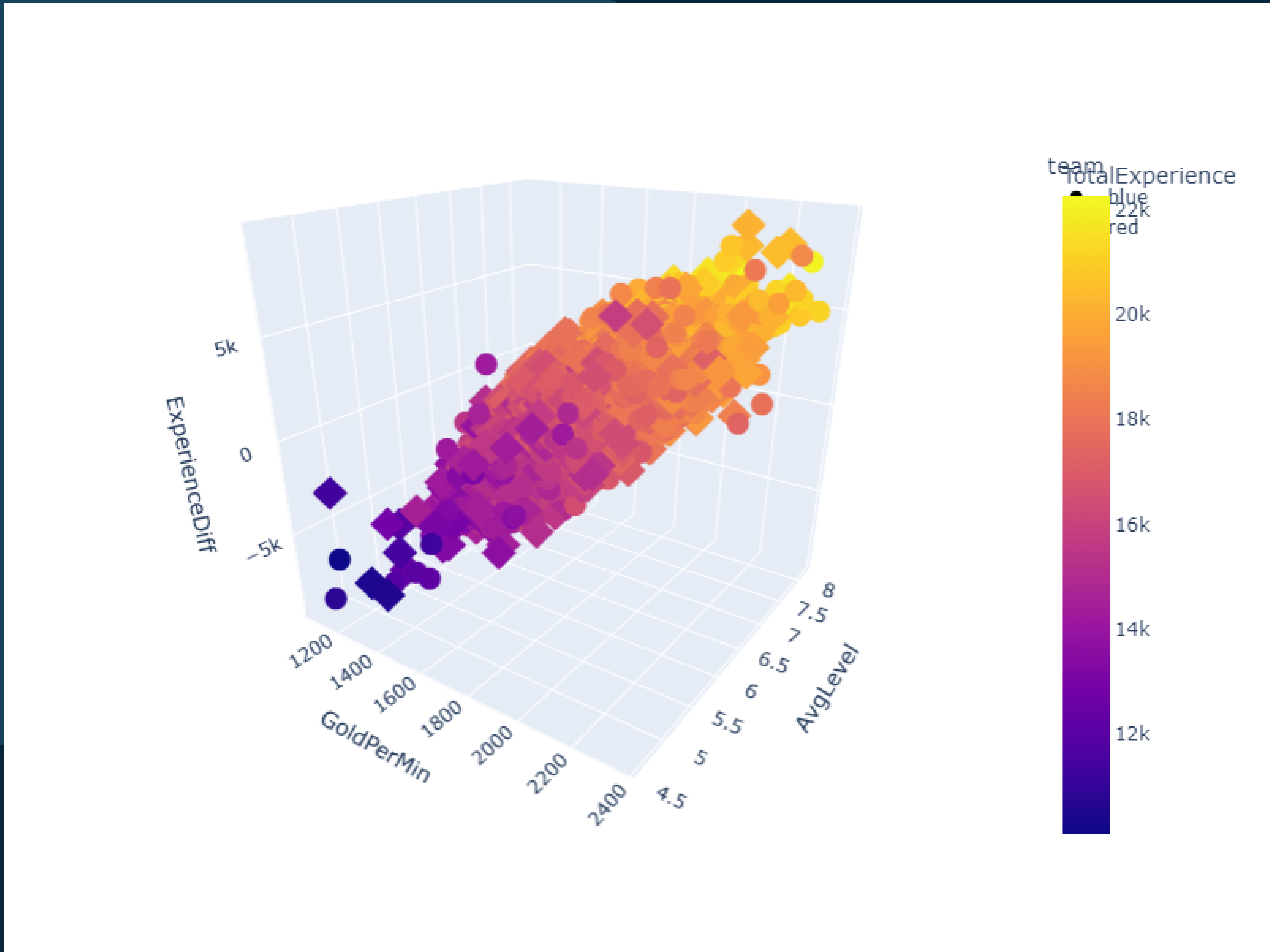
For LoL, at a first glance we find the following:

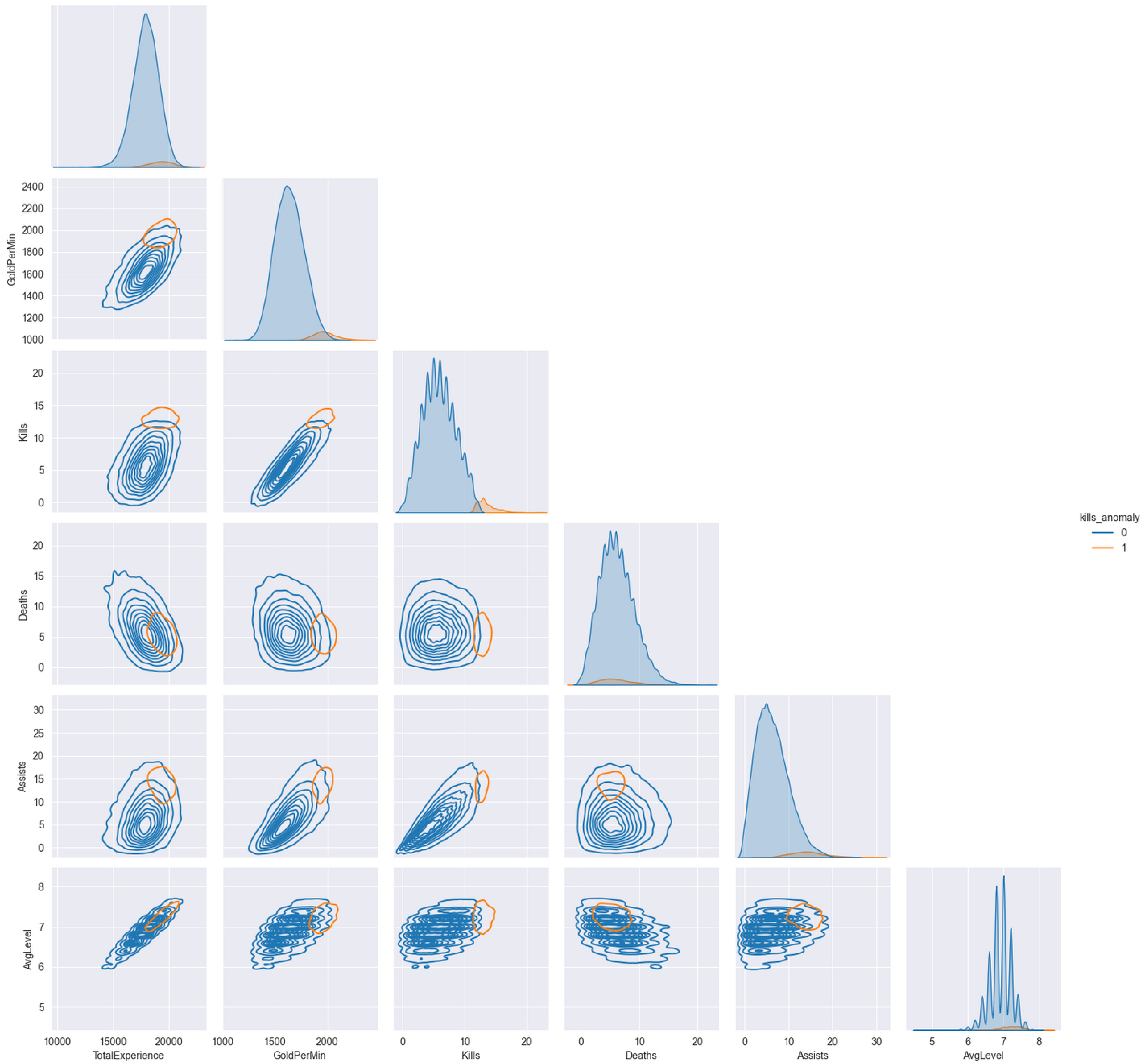
- Higher team champion level comes with high team minions killed
- More wards placed by the team also comes with a higher team champion level



Skill Level

Trying to infer skill level from three variables we find that, for Diamond games, that the higher the difference in experience is between the teams then the more experienced team will generate more gold per minute whilst also having a higher champion level.

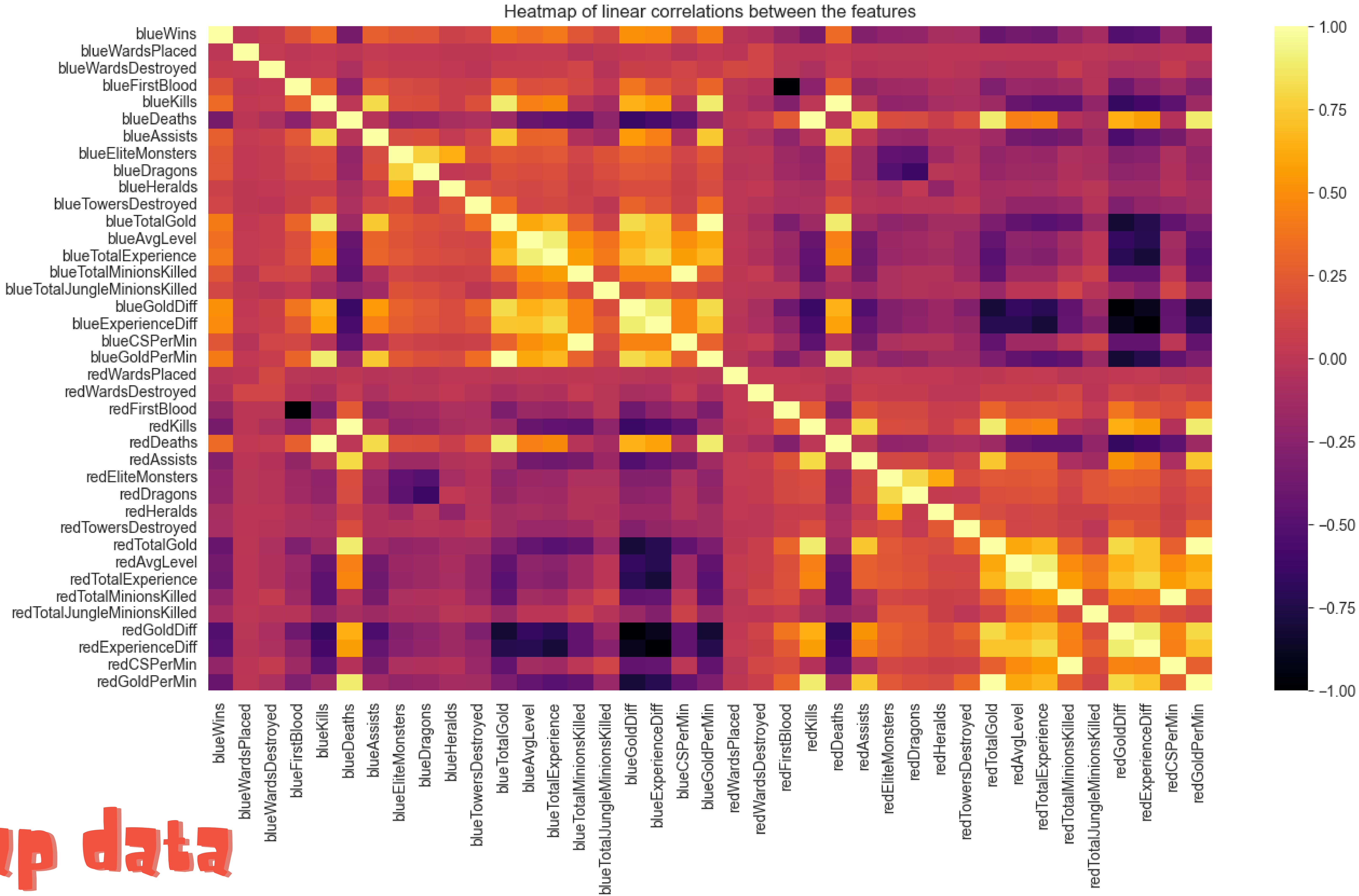




Anomalies

If a team has an extremely high kill rate they tend be very different in their first 10 minutes of the game.

- lower deaths than the non-anomaly teams
- higher kill assists,
- higher kills
- more gold per minute
- higher total experience
- higher champion level.



Heatmap data

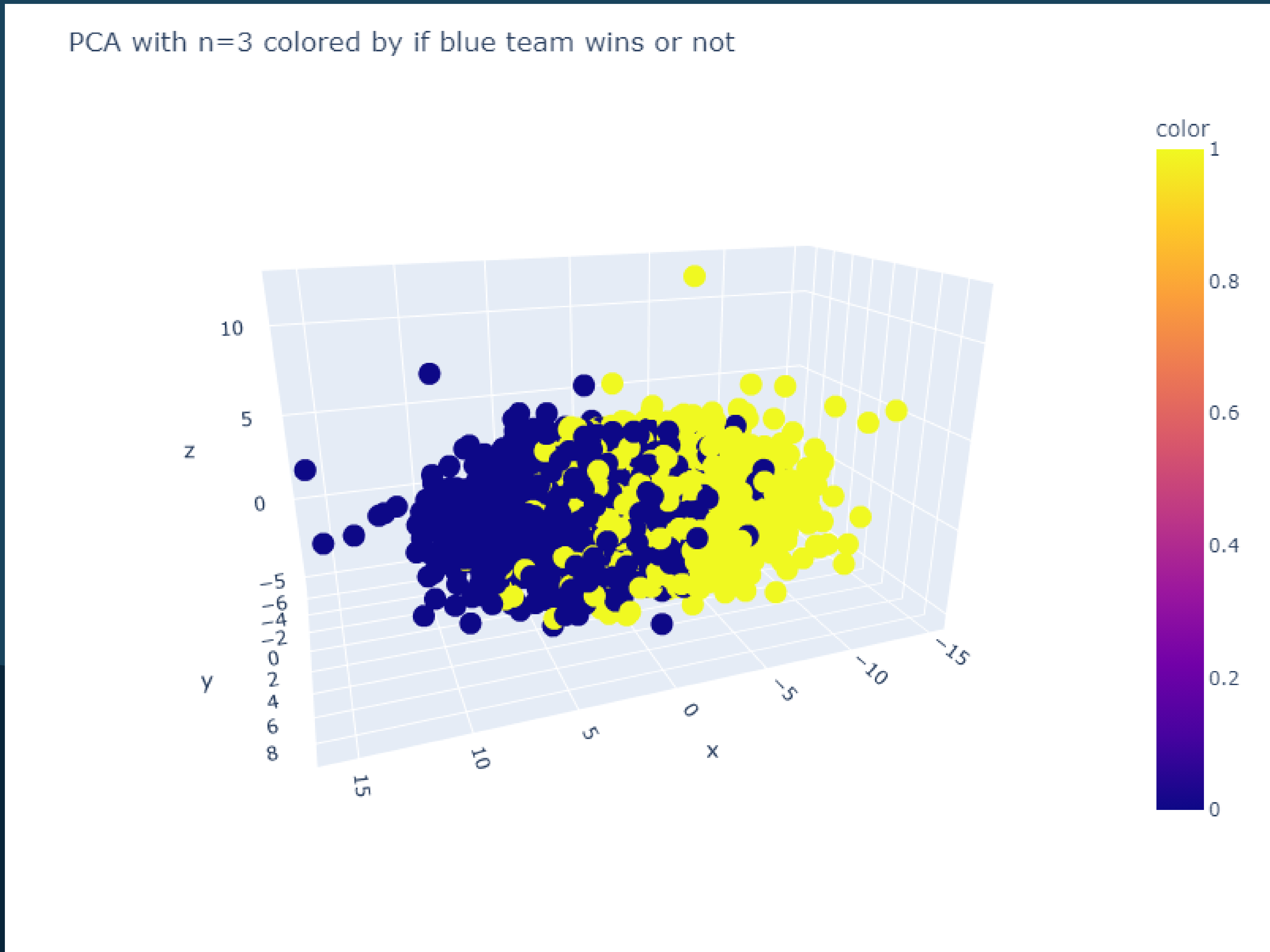
Training time

Our target will be to predict which team wins based on the dataset we have (see heatmap).

We have 5 models that trains on 4 different augmented or reduced datasets. The training and testing grounds are done in a way to represent real-world dynamics such that we can conclude something based on robust methods.

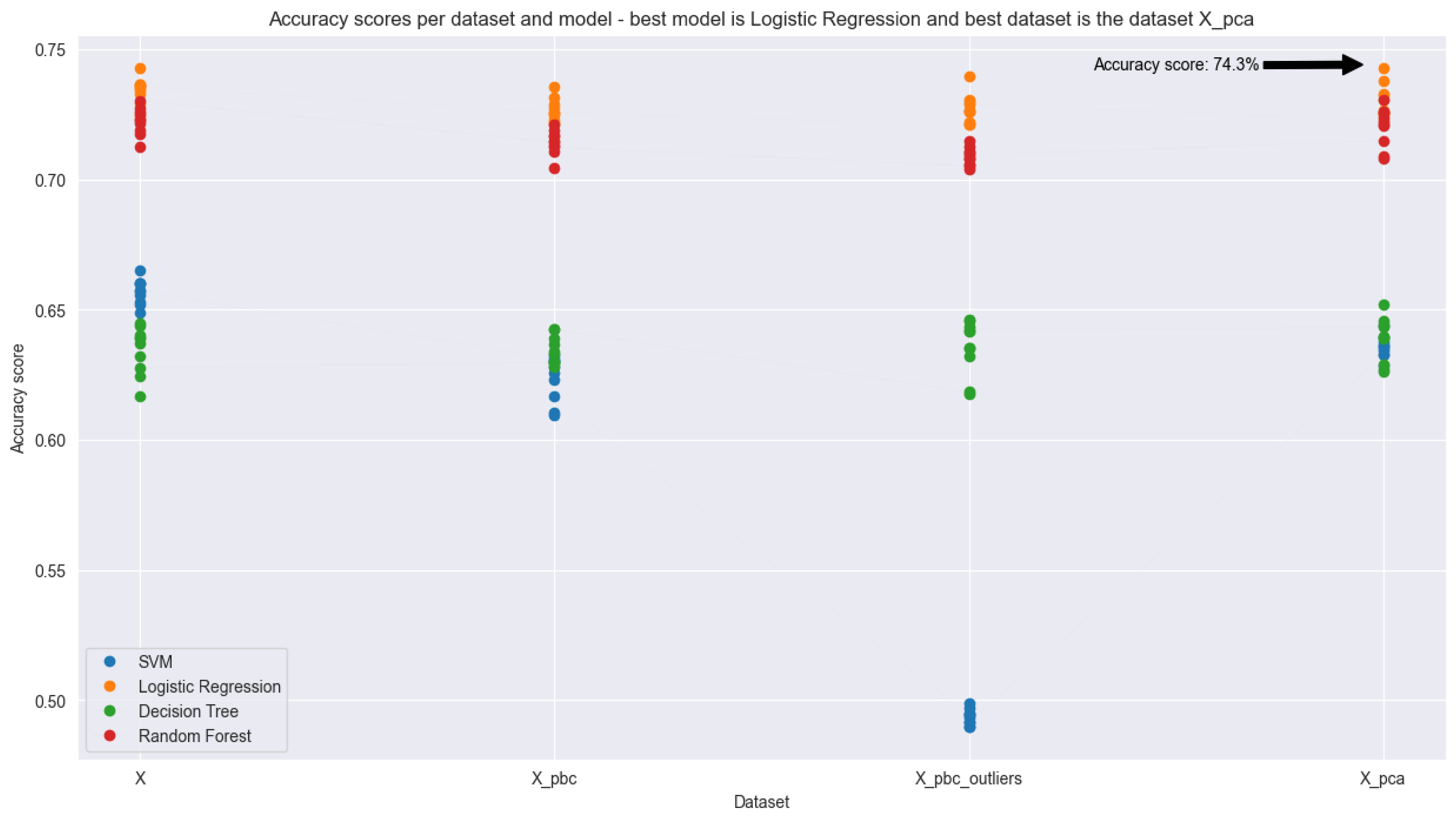
The heatmap data showed all the features of the dataset and their linear correlations. We will be using them and we'll create our own as well by augmenting it with embedding vectors from a PCA model. The PCA model gives us 3 embeddings and below is the reconstruction of the whole dataset in a lower dimensional space. Blue team = Yellow.

```
Using 38 features
SVM: Average Accuracy: 0.66 | Average F1-score: 0.66 | Average F-Beta: 0.66
Logistic Regression: Average Accuracy: 0.73 | Average F1-score: 0.73 | Average F-Beta: 0.73
Decision Tree: Average Accuracy: 0.63 | Average F1-score: 0.63 | Average F-Beta: 0.63
Random Forest: Average Accuracy: 0.72 | Average F1-score: 0.72 | Average F-Beta: 0.72
AdaBoost: Average Accuracy: 0.72 | Average F1-score: 0.72 | Average F-Beta: 0.73
--------------------------------------------------------------------------------------------
Using 16 features
SVM: Average Accuracy: 0.62 | Average F1-score: 0.62 | Average F-Beta: 0.62
Logistic Regression: Average Accuracy: 0.73 | Average F1-score: 0.73 | Average F-Beta: 0.73
Decision Tree: Average Accuracy: 0.63 | Average F1-score: 0.62 | Average F-Beta: 0.64
Random Forest: Average Accuracy: 0.71 | Average F1-score: 0.71 | Average F-Beta: 0.71
AdaBoost: Average Accuracy: 0.72 | Average F1-score: 0.72 | Average F-Beta: 0.72
--------------------------------------------------------------------------------------------
Using 18 features
SVM: Average Accuracy: 0.49 | Average F1-score: 0.20 | Average F-Beta: 0.16
Logistic Regression: Average Accuracy: 0.73 | Average F1-score: 0.73 | Average F-Beta: 0.72
Decision Tree: Average Accuracy: 0.64 | Average F1-score: 0.62 | Average F-Beta: 0.63
Random Forest: Average Accuracy: 0.71 | Average F1-score: 0.71 | Average F-Beta: 0.71
AdaBoost: Average Accuracy: 0.72 | Average F1-score: 0.72 | Average F-Beta: 0.72
--------------------------------------------------------------------------------------------
Using 41 features
SVM: Average Accuracy: 0.64 | Average F1-score: 0.64 | Average F-Beta: 0.63
Logistic Regression: Average Accuracy: 0.73 | Average F1-score: 0.73 | Average F-Beta: 0.73
Decision Tree: Average Accuracy: 0.64 | Average F1-score: 0.63 | Average F-Beta: 0.64
Random Forest: Average Accuracy: 0.72 | Average F1-score: 0.72 | Average F-Beta: 0.72
AdaBoost: Average Accuracy: 0.72 | Average F1-score: 0.72 | Average F-Beta: 0.72
--------------------------------------------------------------------------------------------
```



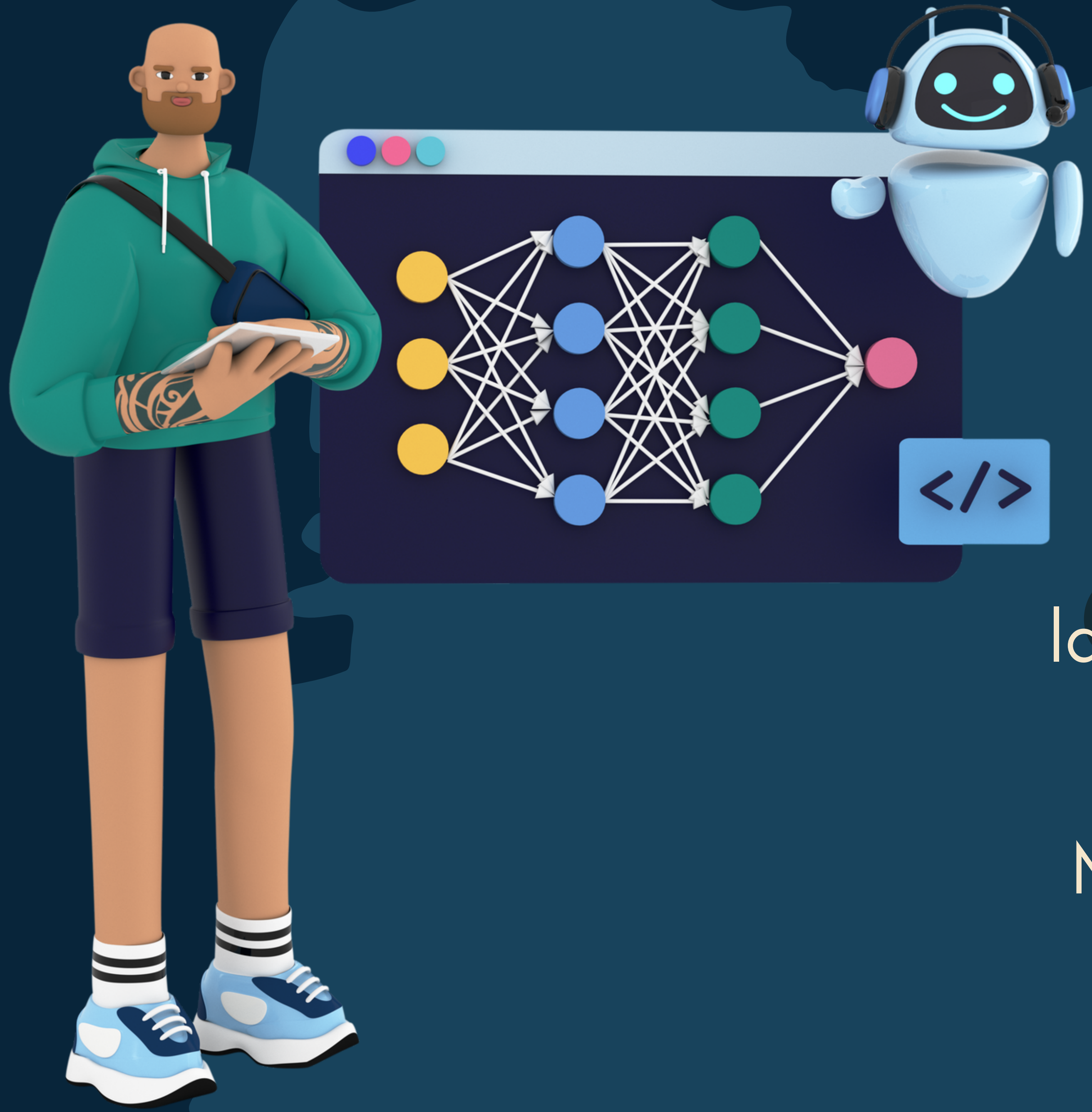
Model results

The best model is a logistic regression using our augmented dataset with the PCA embeddings. The second best model is the Random Forest which is a tree-based model.



Wrap Up

- Highly experienced teams stand out in the total minions they kill, their champion level, and the warding totems they place
- Highly experienced teams also generate more gold per minute and have higher champion level
- We can augment the dataset to gain more data
- The lower dimensional space shows us the different teams as 2 different inter-mixed clusters
- A real-world model in production would be to have a logistic regression with an augmented dataset with PCA embeddings

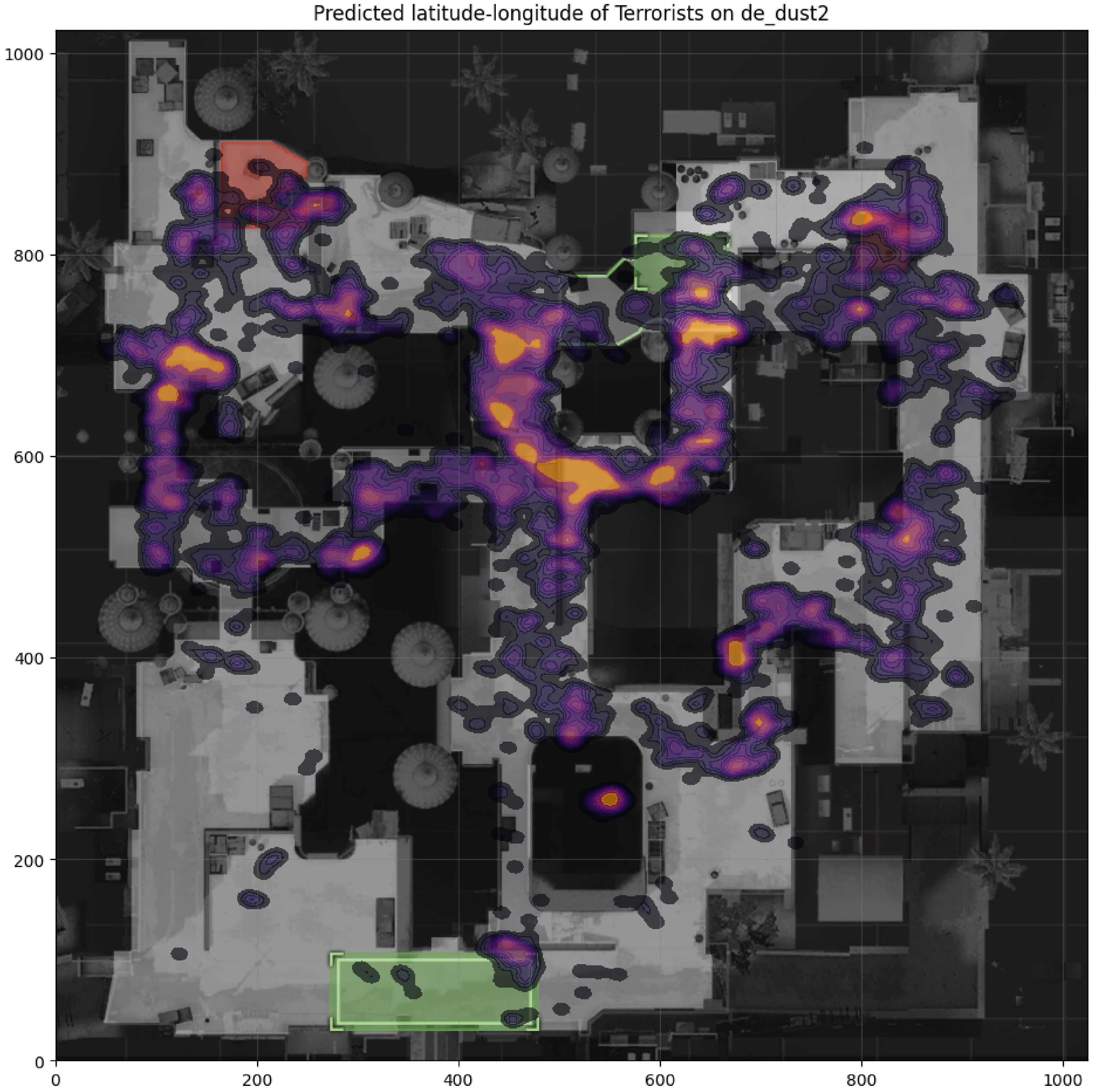


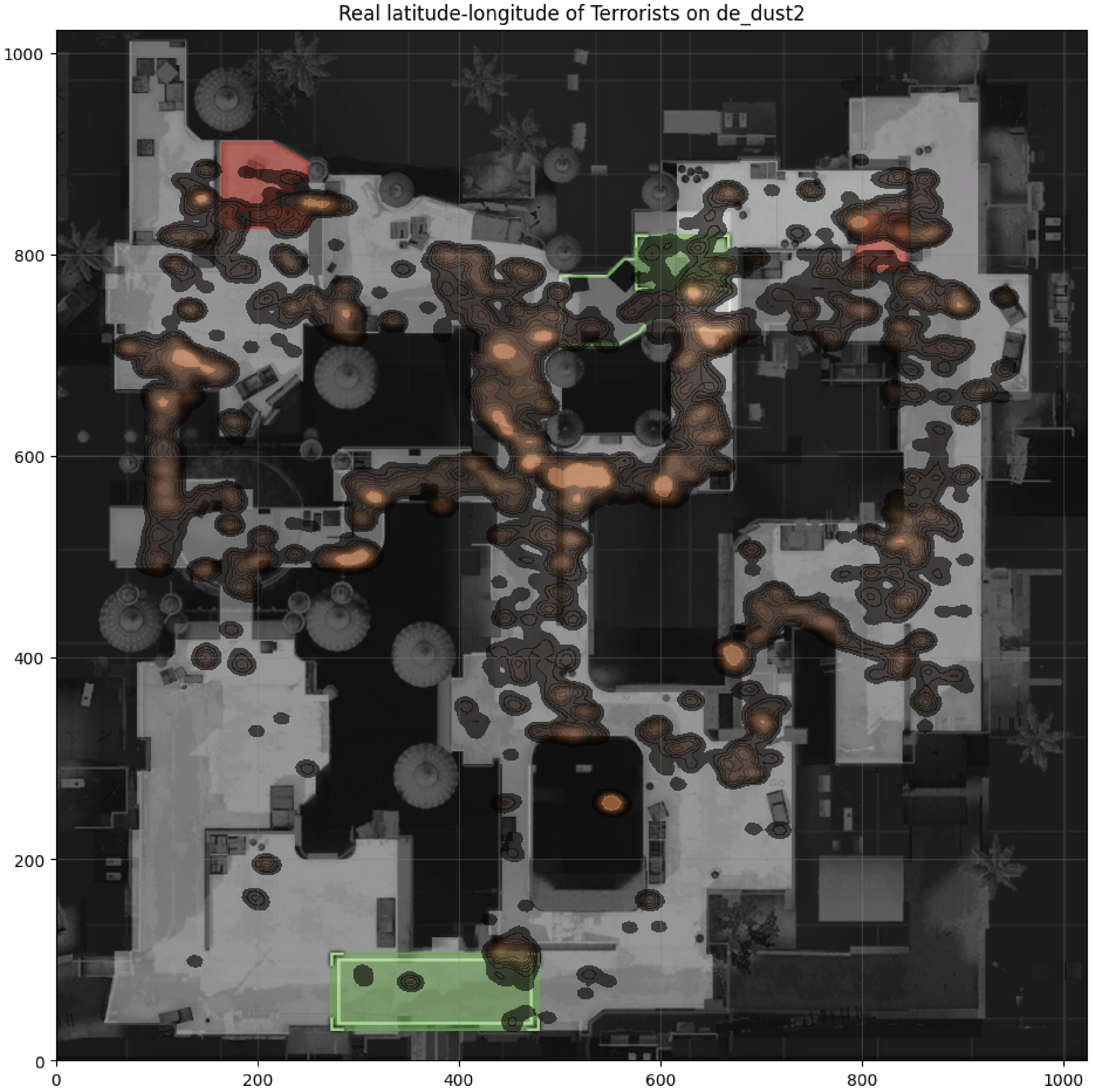
BONUS

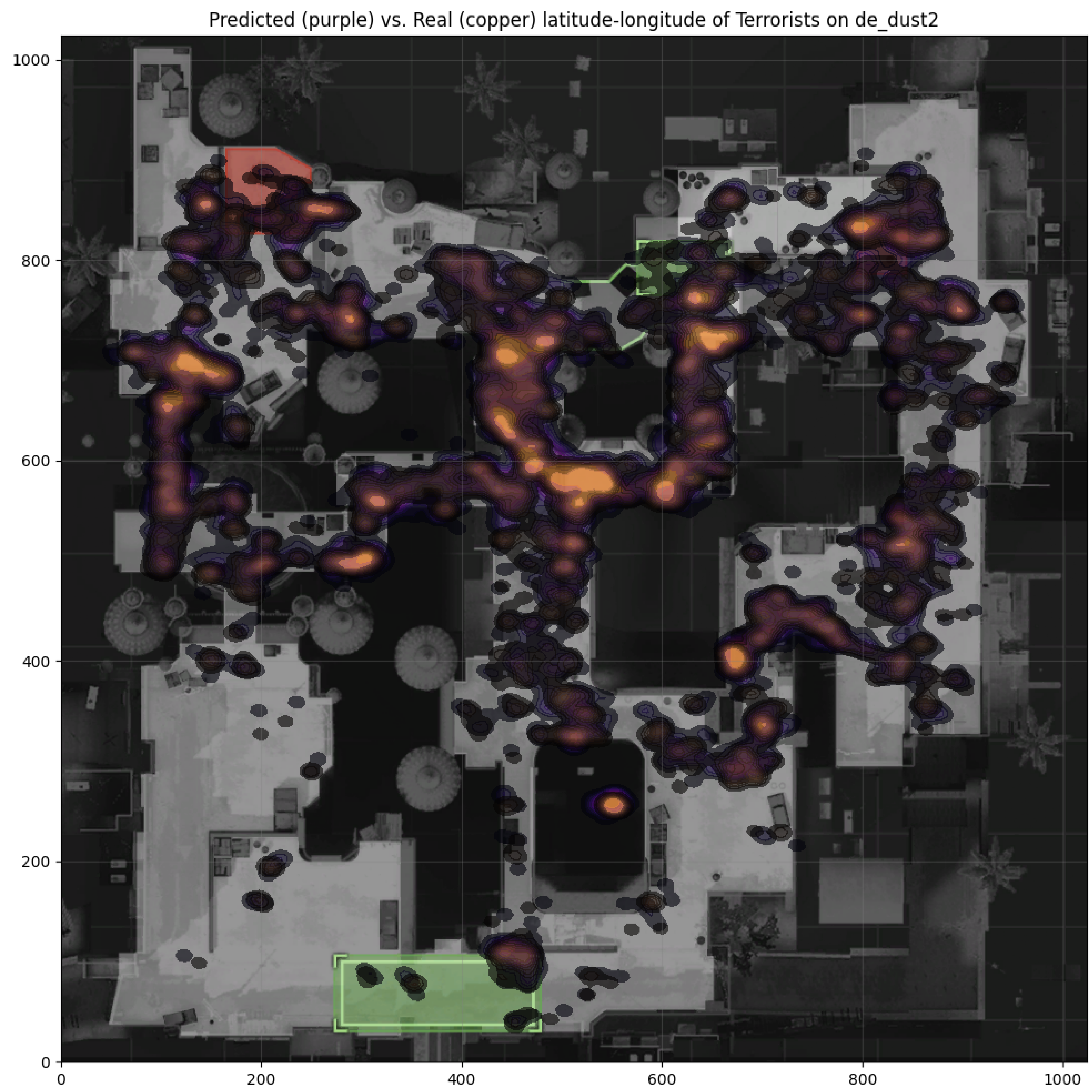
Just for demonstration, taking the longitude and latitude data of Terrorist attack patterns on de_dust2 and feeding it to a Long Short Term Memory model (LSTM) which is a variant of a Recurent Neural Network (RNN), we can predict the flow of a game if the data exists.

This is the output from 4 games of Terrorist attack patterns on de_dust2.

BONUS







BONUS

Superimposing the predicted and the real data on the same plot shows us the regions where the neural network fails model appropiately.

However, this could be scaled and applied to any game and any movement with coordinates.

Thank You!