



provided via 115 features. In the work presented here, these 115 features are selectively narrowed down to 14 key features for subsequent analysis and modelling as presented in Table 2. In this study, we focus on event locations, durations, and events like passes, carries, and shots. This approach enables an attentive examination of in-possession events through analysing the data and mapping each event with the home or away team involved in the match.

TABLE I. STATSBOMB COMPETITION DATASET

Competition Dataset Columns		
Column Name	Data Type	Description
Match_id	Int	The unique identifier for the match
Competition	String	The competition or league the match was apart of
Match_week	Int	The week of the season the match took place in
Home_team	String	The team that played at their home venue
Away_team	String	The team that travel to the home team's venue
Home_score	Int	The number of goals the home team scored in the match
Away_score	Int	The number of goals the away team scored in the match

Each record in the competition dataset refers to a match or fixture that is played between a home team and an away team. The relevant event (events with the same match id) can be mapped to a team in possession as either the home team or the away team from the corresponding record in the competition table.

TABLE II. STATSBOMB EVENT DATASET

Event Dataset Columns		
Column Name	Data Type	Description
Match_id	Int	The unique identifier for the match, corresponds to the match_id provided in the Match Dataset
Index	Int	The number indicating the chronology order of the labelled event in the match
Possession_team	String	The team that is in possession of the ball when the event occurs
Type	String	Type of event (e.g., Shot, Carry, Pass)
Duration	String	Duration of the event (in seconds)
Location	Pair of Floats	The horizontal (1 <sup>st</sup> value) and vertical (2 <sup>nd</sup> value) of where an event happens on the pitch. E.g (60,40 is the center spot in the middle of pitch)
Dribble_outcome	String	Details the outcome of the dribble. E.g. complete or incomplete. Note: a completed dribble is when the player is not tackled after taking on an opponent
Pass_outcome	String	Details the outcome of the pass. E.g. incomplete, out, etc. note: nan is a complete pass.
Shot_outcome	String	Outcome of a shot event (e.g., Goal, Saved)
Shot_statsbomb_xg	Float	Expected goals (xG) for shot events, valued between 0 and 1. This is StatsBomb's calculated and recorded value
Carry_end_location	Pair of Floats	The horizontal (1 <sup>st</sup> value) and vertical (2 <sup>nd</sup> value) of where a carry event ends on the pitch.
Pass_end_location	Pair of Floats	The horizontal (1 <sup>st</sup> value) and vertical (2 <sup>nd</sup> value) of where a pass event ends on the pitch.
Shot_end_location	Pair of Floats	The horizontal (1 <sup>st</sup> value) and vertical (2 <sup>nd</sup> value) of where a shot event ends on the pitch.
Pass_length	Float	A calculated length of the pass from location to pass_end_location. This is StatsBomb's calculated and recorded value

Fourteen key features in the dataset have been engineered to develop over 20 significant match-related features to use as inputs to machine learning algorithms for predicting match outcomes. Within the StatsBomb event dataset, unnecessary data are first filtered out by focusing on important event types like "Pass", "Carry", and "Shot", which provide essential spatial information around in-game events. This data selection process filters the data to a manageable subset, resulting in over 3 million rows from a total of 1823 professional football games in various European leagues.

The StatsBomb dataset maps pitch locations with precise coordinates: horizontally ( $x$ ) using a range from 0 to 120 and vertically ( $y$ ) using a range from 0 to 80. Fig. 2 provides a visualization of the key locations on the pitch and their corresponding  $x$  and  $y$  coordinates. These coordinates can be used to delineate the pitch into thirds: defensive ( $x < 40$ ), midfield ( $40 \leq x \leq 80$ ), and attacking ( $x > 80$ ). We can also detail specific zones like the attacking penalty box ( $x > 102$ ;  $18 \leq y \leq 62$ ), thereby facilitating accurate event mapping and subsequent analysis.

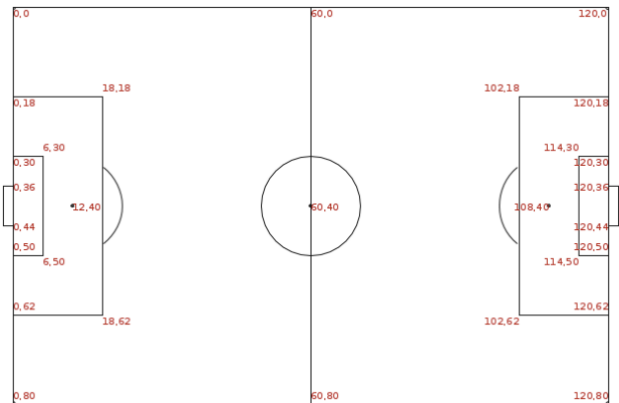


Fig. 2. Statsbomb pitch labels of key locations. The defending goal is always on the left ( $x$  value of 0) and the attacking goal is always on the right ( $x$  value of 120) when considering event data.

Event durations indicate team possession times, allowing for possession percentage calculation. Passes and carries, recorded with start and end locations, contribute to team possession through successful completions.



Fig. 3. A visualisation of the probability of expected threat provided by Singh [13]. The attacking team are targetting the goal on the right.

Singh demonstrated that expected Threat (xT) [13] quantifies the strategic value of attacking actions by assigning

higher xT to moves that bring play closer to the opponent's goal. This involves dividing the pitch into grids, with xT values adjusted based on ball movement across zones, providing a nuanced understanding of possession dynamics and goal-scoring potential. The xT probability of each location is visualized in Fig 3. Events illustrations can be generated using the `soccermatics` [14] Python library [15].

Expected Threat (xT) reflects the dynamic chances of scoring, increasing with actions closer to the opponent's goal. Since not all forward movements increase xT, highlighting the strategic play through passes and carries leading to shots. xT evaluates the potential of sequences of events (also called possession chains) ending in shots, aiding in assessing team performance by their positioning and threat generation. Shots are assessed by StatsBomb's expected goals (xG) metric [9], factoring in shot location, defensive pressure, ball height, player contact, and goalie positioning. This pre-calculated xG, integral to assessing goal likelihood, requires no further calculations for this study. Fig. 4 illustrates an example of positive and negative xT values resulting in a goal. In this possession chain, possession was first gained from a throw in. Passes are illustrated using purple arrows, carries using orange arrows and the shot is represented using the gold arrow.

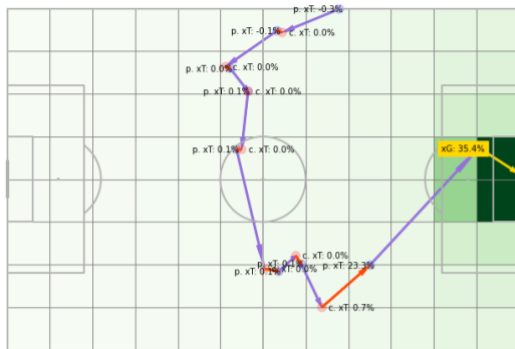


Fig. 4. A series of events in a possession chain that involved passes, carries and a shot which turned out to be a goal.

In this study we focus on statistical features from 1823 games, rather than using historic player or team reputation. Insights from event data, alongside game knowledge and previous research, have led to the ability to identify features which can be categorized into four distinct groups, detailed in the next section.

### III. FEATURE ENGINEERING

Utilising historical data from the StatsBomb dataset, we engineers features to predict match outcomes, focusing on in-possession events. These features are categorised into **General, Dribble/Carry, Passing, and Shooting**, quantifying team performance. Table 3 details the features used. The utility of these features is assessed by analysing their correlation with match results and goals scored, considering the actions of both home and away teams.

In Table 3 the match ID distinguishes the match the event is in relation to, and the possession team name distinguishes the team the event benefits. These features all produce numeric values. The features, centred around ball possession,

offer insights for deeper data analysis. Using Seaborn [16], a correlation matrix is used to highlight key feature relationships.

TABLE III. CALCULATED FEATURES

Calculated Features Columns				
	Feature Name	Columns from Event Dataset used	Description of their calculation	Associated Features
General Features	Possession	Duration, Location	Takes a sum of the duration of each event for each team. This will give a total value of how long a team was in possession of the ball in a match.	Possession Defending third, Possession Attacking third, Possession Percentage Share
	Events count	Location	Takes a count of the number of events each possession team has.	Events count Defending third, Events count Attacking third, Events count in the box
	Events distance from team goal	Location	This takes the mean distance of the events for that team from their own goal (location value (0,40)). And the opponents goal, (120,40)	Events distance from opponents' goal
Dribble / Carry Features	Dribbles count	Type, Dribble_outcome	Counts the total number events of type "Dribble" for each team.	Dribbles Completed, Dribbles Completed Percentage
	Carries count	Type, Location, Carry_end_location	Counts the total number events of type "Carry" for each team.	Carries Length, Carries Mean Length, Carries count Defending third, Carries count attacking third
	Expected Threat (xT) from Carries	Type, Location, carry_end_location	Accumulates the total value of xT calculated from each carry a team has in the match.	Total xT from Passes and Carries
Passing Metrics	Passes count	Type, Location, Pass_outcome, Pass_length, Pass_end_location	Counts the total number events of type "Pass" for each team. Field tilt Compares each team's calculated passes in the final third and works out the percentage relevant to each team. FT of A = Att 3 <sup>rd</sup> passes of A / (Att 3 <sup>rd</sup> passes of A + B) %.	Passes Completed, Passes Completed Percentage, Passes Length, Passes Mean Length, Passes Defending third, Passes Attacking third, Field Tilt
	Expected threat (xT) from Passes	Type, Location, Pass_end_location, Pass_outcome	Accumulates the total value of xT calculated from each completed pass a team has in the match.	Total xT from Passes and Carries
Shooting Metrics	Shots count	Type	Counts the total number events of type "Shot" for each team.	Shots on-target count, Shots on-target percentage
	Expect Goals (xG) from Shots	Type, Shot_statsbomb_xg	Taking a sum of the Statsbomb provided xG values	Mean xG

The matrix differentiates between home and away team features, correlating them with match outcomes (win, draw, loss) to uncover impactful trends. Fig. 5 shows a selection of the most interesting and relevant features based on their correlation values. The top half of the correlation matrix includes features relevant to the home team and the bottom

half includes the features relevant to the away team. The results are indicated by H for home team win, D for draw, and A for away team win. The non-diagonal elements represent the correlation between pairs of different variables. Correlation matrices were created for all the categorised features and displayed in Fig. 5.

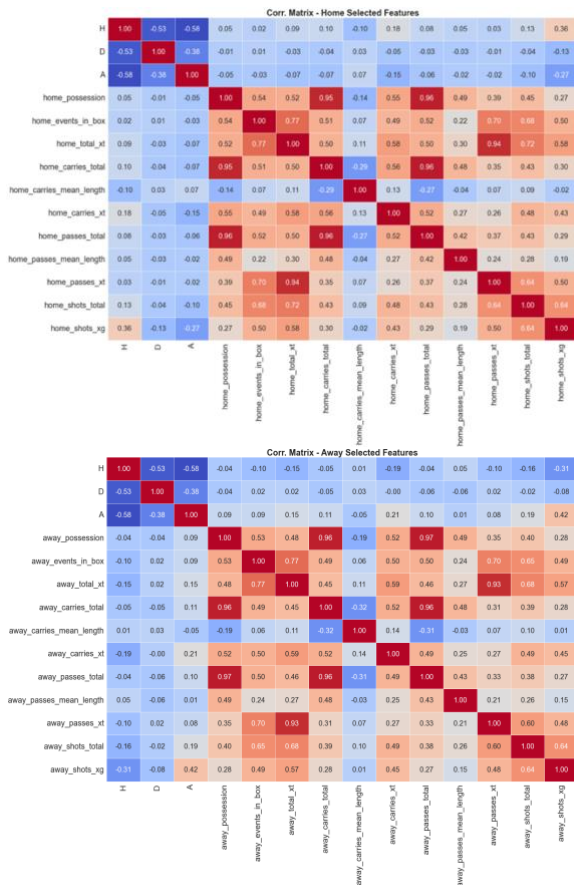


Fig. 5. Correlation matrices showing the relationship between results and goal scores using selected features.

Home team data shows a strong link between possession and expected threat (xT) with a correlation measure of 0.52, with the highest correlation (0.96) found between possession and carries xT. Home shots xG also correlates significantly with home wins (0.36), while longer carries negatively correlate with frequency (-0.39). Similarly, away team features mirror home trends, with away shots xG (0.42) and carries xT showing strong positive correlations with winning. Despite common beliefs, possession alone doesn't correlate directly with winning, highlighting the need for nuanced metrics like xT for predictive accuracy. These insights form a robust foundation for developing features for use with machine learning models.

Other opportunities for feature engineering include possession chain analysis as well as looking at other metrics such as blocks, saves, fouls, free kicks, corners, and many others. Whilst these have been excluded in the work presented here, it demonstrates many opportunities for further work on engineering other features.

#### IV. MACHINE LEARNING ALGORITHM VARIATIONS

In the work presented here, various machine learning models were considered. Using the scikit-learn Python library [17], the data from 1823 matches was split into 1459 training- and 365 testing-sets. The selected machine learning models include Support Vector Machine, Random Forest, and XGBoost. Each model is trained to predict home wins, draws, and away wins using the engineered features as inputs. The modelling process involves one-hot encoding of categorical outcomes and evaluating model accuracy based on prediction probabilities across these categories. Other models considered but not included in this paper include Logistic regression, Decision Trees, K-nearest neighbour, and Naive Bayes, but have been omitted due to their poor performance.

TABLE IV. MODEL PREDICTION RESULTS

Calculated Features Columns				
Model	Accuracy (H)	Accuracy (D)	Accuracy (A)	Overall Accuracy
Support Vector Machine (SVM) Linear Kernel	0.773	0.740	0.814	0.658
Support Vector Machine (SVM) RBF Kernel	0.553	0.789	0.699	0.488
Random Forest (RF)	0.696	0.781	0.767	0.625
Extreme Gradient Boosting (XGBoost)	0.748	0.712	0.781	0.611

Table 4 illustrates the performance of the selected models. All results were all obtained using the default scikit-learn parameters. In the case of the SVM model with a linear kernel there is no limit to the maximum number of iterations used on whilst training, significantly increasing the runtime of model fitting model compared to the other approaches.

To evaluate model performance, each algorithm was assessed using a confusion matrix and classification report. The confusion matrix details true positives, false positives, true negatives, and false negatives, while the classification report provides metrics such as precision (correct positive predictions relative to total positive predictions), recall (correct positive predictions relative to all actual positives), and F1-score (harmonic mean of precision and recall). These metrics offer insights into each model's accuracy and its ability to handle class imbalance, with macro and weighted averages highlighting performance across different classes.

The SVM model with a linear kernel was the most accurate at predicting match outcomes, achieving a 65.8% accuracy rate. This suggested definitive linear boundaries generated within the data, which also implies that the key features of football matches can directly affect the outcome matches. However, its performance dropped to 44.9% when limited to 1,000,000 iterations, indicating that while effective, its suitability for larger datasets or real-time predictions may be limited due to potential efficiency issues. Adjustments to the data normalisation process and model parameters, like the hyperplane tolerance and regularization parameter C, could further enhance model efficiency and effectiveness.

The Random Forest model ranked second in accuracy at 62.5% and performed well in predicting draws. Despite its strengths, the model faced challenges, indicated by lower precision, and recall rates. As an ensemble of decision trees, Random Forest's effectiveness can be fine-tuned by adjusting

the number of trees, or `n_estimators`, using `scikit-learn`'s `GridSearchCV`. The balance between too few trees, risking underfitting, and too many, causing overfitting and increased computational demands, is crucial. Optimal tree counts were found to be 250 for home wins, 500 for away wins, and 2500 for draws, with diminishing returns beyond these numbers.

Fig. 6 illustrates the optimal number of trees. This reinforces the fact that the random forest's precision and recall are low for Draws and the number of trees just keeps increasing without ever finding an optimal value. Using the new parameters for the model results in higher accuracies for predicting just Home Wins (71.5%), Away Wins (77.8%) and Draws (79.5%) in isolation but this resulted in a small decrease to overall accuracy (61.3%).

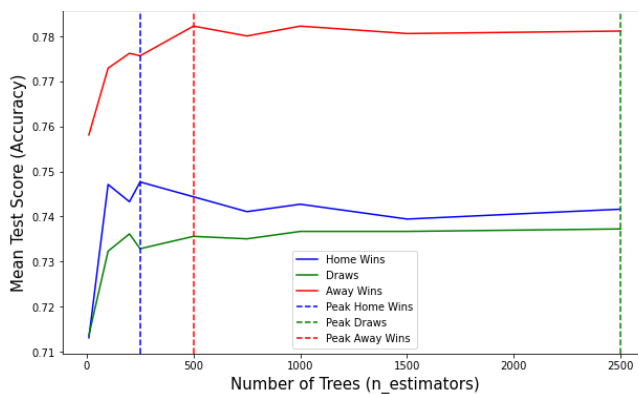


Fig. 6. A line graph plotting the Mean Test score of various `n_estimators` values to help decide the optimal hyperparameters for the RF model.

XGBoost achieved a 61.1% accuracy, featuring capabilities like handling missing data and regularization to curb overfitting. It allows fine-tuning through parameters such as `n_estimators` (tree number), `learning_rate` (to control learning speed and reduce overfitting), and `max_depth` (limiting tree depth). Default settings are 100 trees, a 0.3 learning rate, and a max depth of 6.

Random Forest and XGBoost both excel in football score prediction through iterative decision tree construction, each addressing the previous errors for enhanced accuracy. Random Forest uses parallel trees with bagging and feature randomness to mitigate overfitting, diversifying its model by sampling data and features. Conversely, XGBoost builds trees sequentially, focusing on correcting past mistakes to minimize bias and loss, thus gradually improving predictions.

Optimizing XGBoost for the dataset involved adjusting several parameters: increasing `n_estimators` to 1000, decreasing the learning rate to 0.01, and reducing max depth to 3. This was achieved using an element of trial and error as `scikit-learn`'s `GridSearchCV` and `RandomizedSearchCV` provided little or no significant improvements to overall accuracy. These changes led to an approved 64.4% accuracy. The optimized results are detailed in the confusion matrix and classification report in Table 5. The XGBoost model achieves 64% accuracy in predicting football match outcomes, with high F1-scores for home (0.75) and away wins (0.69), but a lower score (0.26) for draws, suggesting difficulty in this area due to possible class imbalance or feature issues. Improvements might include class balancing or enhanced feature engineering. This model, particularly effective for

definitive outcomes, shows promise for predicting future matches based on historical data.

TABLE V. XGBOOST CONFUSION MATRIX AND CLASSIFICATION REPORT

Predicted Values	Actual Values		
	H	D	A
H	141	27	9
D	35	18	24
A	9	27	141

Class	Precision	Recall	F1-Score	Support
H	0.72	0.80	0.75	177
D	0.31	0.23	0.26	77
A	0.70	0.68	0.69	111

Macro Average	0.57	0.57	0.57	365
Weighted Average	0.62	0.64	0.63	365
Accuracy			0.64	365

Table 1. The XGBoost confusion matrix and classification report.

## V. PREDICTING MATCHES USING TEAM AVERAGE AS THE SEASON PROGRESSES

Simulated match predictions utilise engineered features, aggregating each team's previous match features to calculate mean values. This involves combining home and away team data from past games to form new average feature values for upcoming matches, enhancing prediction accuracy. Predictions start from a team's second game, utilizing mean values from previous match feature categories **General**, **Dribble/Carry**, **Pass**, and **Shot**. With each subsequent game, these averages update, incorporating all prior games' data. The XGBoost model, trained on matches up to the current one, uses this evolving dataset for predictions, requiring at least one past game for accurate forecasting. The dataset covers 1823 matches across 171 dates, with a median of 6 and a mean of 10 games per date, ranging from 1 to 28 games. Collecting the match data into **Match Weeks**, provided better sequential analysis of the results. Fig 7 shows the accuracy for each of the relevant match weeks. Rolling predictions start from a team's second game, utilising mean values from previous match feature categories **General**, **Dribble/Carry**, **Pass**, and **Shot**. With each subsequent game, these averages update, incorporating all prior games' data. The XGBoost model, trained on matches up to the current one, uses this evolving dataset for predictions, requiring at least one past game for accurate forecasting. As the dataset grows by 49 games per match week, it enhances the algorithm's training, notably improving draw prediction accuracy by 0.33% per week as the season progresses. This incremental data input boosts overall prediction accuracy by 0.32% per week, illustrating the algorithm's potential as a long-term predictive tool for football match outcomes.

The algorithm's overall accuracy peaked with an improvement of 0.53% in the final match week of the season, resulting in a notable 90% prediction success rate for games in the final week of the Spanish La Liga. Detailed results and team performances are presented in Table 6. Predictions were made for all 5 leagues considered.

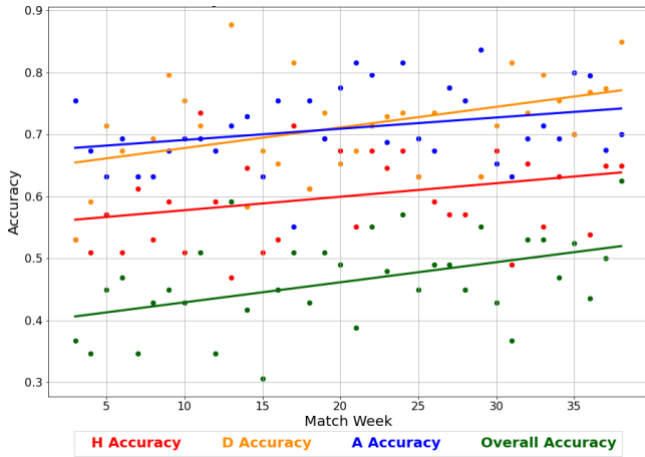


Fig. 7. Line graph plotting the Home (H), Draw (D) and Away (A) outcome accuracy per match week using XGBoost.

TABLE VI. THE FINAL MATCH WEEK OF LA LIGA PREDICTION

Competition	Home Team	Away Team	Result	Prediction
Spain - La Liga	Valencia	Real Sociedad	A	A
Spain - La Liga	Atlético Madrid	Celta Vigo	H	H
Spain - La Liga	Athletic Club	Sevilla	H	H
Spain - La Liga	RC Deportivo La Coruña	Real Madrid	A	A
Spain - La Liga	Granada	Barcelona	A	A
Spain - La Liga	Espanyol	Eibar	H	H
Spain - La Liga	Málaga	Las Palmas	H	H
Spain - La Liga	Sporting Gijón	Villarreal	H	A
Spain - La Liga	Rayo Vallecano	Levante UD	H	H
Spain - La Liga	Real Betis	Getafe	H	H

Table 2. The Table showing the Home Team, Away Team and the Result and the Algorithms prediction for Match week 38 in La Liga.

## VI. CONCLUSION AND FUTURE WORK

This research has made significant strides in predicting football match outcomes using event data and machine learning. By meticulously analysing football match event data and engineering relevant features, the research has demonstrated that certain on-pitch actions and strategic manoeuvres significantly correlate with match outcomes. The insights gained from match events, alongside the benefits of engineered features and machine learning algorithms demonstrate how optimized predictive solutions can be found. Despite the achievements presented, this research opens many potential avenues to explore further, through expanding on the feature engineering aspect or optimising the machine learning.

## ACKNOWLEDGMENT

Thanks to StatsBomb who provided sufficient data to conduct the research presented in this paper.

## REFERENCES

- [1] StatsBomb, "The 2015/16 Big 5 Leagues Free Data Release." Dec. 2023. <https://statsbomb.com>
- [2] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001.
- [3] XGBoost Documentation. Accessed: Dec. 01, 2024. <https://xgboost.readthedocs.io>
- [4] C. A. Casal, et al, "Possession zone as a performance indicator in football. The game of the best teams," *Front Psychol*, vol. 8, no. JUL, Jul. 2017, doi: 10.3389/fpsyg.2017.01176.
- [5] C. A. Casal, et al "Possession in football: More than a quantitative aspect - A mixed method study," *Front Psychol*, vol. 10, 2019: 10.3389/fpsyg.2019.00501.
- [6] M. Klemp, et al, "In-play forecasting in football using event and positional data," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-03157-3.
- [7] Q. Zou, K. Song, and J. Shi, "A Bayesian in-play prediction model for association football outcomes," *Applied Sciences*, vol. 10, no. 8, 2020, 10.3390/APP10082904.
- [8] StatsBomb Open Events Structure and Data Specification v4.0.0, *Github*. May 08, 2019. [github.com/statsbomb](https://github.com/statsbomb)
- [9] Statsbomb, "What Are Expected Goals (xG)?" Dec. 01, 2023.
- [10] Statsbomb, "What is Expected Threat (xT)? Possession Value Models Explained." Dec. 01, 2023.
- [11] Towards AI, "Sports Analytics 101—Expected Threats (xT)", Accessed: Dec. 01, 2023. Available: <https://towardsai.net/p/sports-analytics-101-expected-threats-xt>
- [12] StatsBomb, "statsbombpy." *Github*, 2023. Accessed: Dec. 19, 2023. [github.com/statsbomb/statsbombpy](https://github.com/statsbomb/statsbombpy)
- [13] Karun Singh, "Introducing Expected Threat (xT)." Accessed: Aug. 01, 2023. [karun.in/blog/expected-threat.html](https://karun.in/blog/expected-threat.html)
- [14] A. A. (code) David Sumpter, "soccermetrics." 2022. Accessed: Dec. 1, 2023. [Online]. Available: <https://soccermetrics.readthedocs.io/en/latest/>
- [15] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput Sci Eng*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.
- [16] M. Waskom, "seaborn: statistical data visualization," *J Open Source Softw*, vol. 6, no. 60, p. 3021, Apr. 2021, doi: 10.21105/joss.03021.
- [17] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.