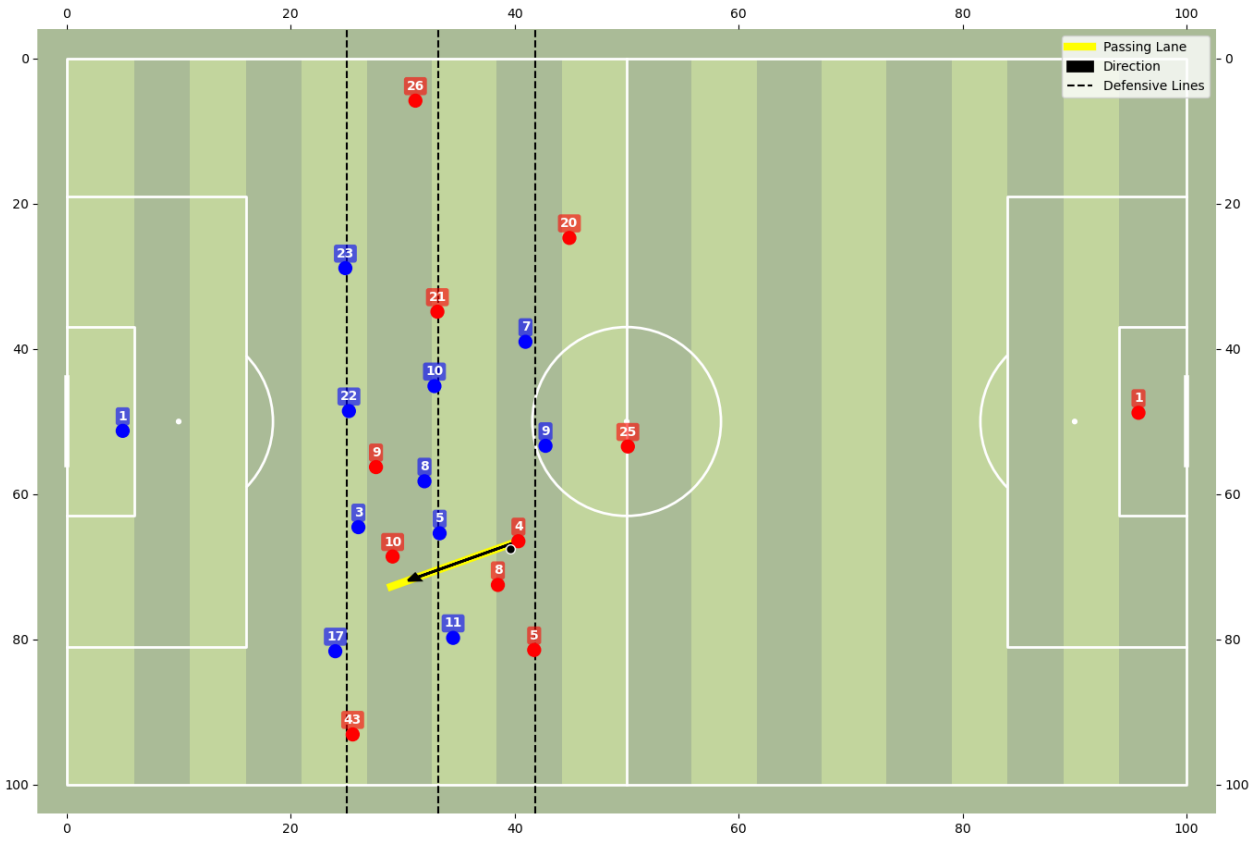

Expanding the Scope of Football Analytics by Integrating Tracking Data and Utilizing Statistical Learning

A collaboration with PlaymakerAI

Master's thesis in Complex Adaptive Systems - MPCAS

ALBIN STEEN

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Expanding the Scope of Football Analytics by Integrating Tracking Data and Utilizing Statistical Learning

A collaboration with PlaymakerAI

ALBIN STEEN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2025

Expanding the Scope of Football Analytics by Integrating Tracking Data and Utilizing Statistical Learning
A collaboration with PlaymakerAI
ALBIN STEEN

Supervisor: Jesper Haglöf, PlaymakerAI
Examiner: Rebecka Jörnsten, Department of Mathematical Sciences

Master's Thesis 2025
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Tracking Data frame for a passing event from the game Real Madrid vs Borussia Dortmund 22nd October 2024.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Expanding the Scope of Football Analytics by Integrating Tracking Data and Utilizing Statistical Learning
A collaboration with PlaymakerAI
ALBIN STEEN
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

The recent decade has seen a data revolution within football as data analytics and statistical learning have become established vital tools. The data revolution has also seen the emergence of a new type of football data called tracking data. This thesis first explores how information from tracking data can be integrated with established event data and improve an established statistical learning model providing an expectancy metric for passes called xP. Secondly, the thesis explores how it can used to create a new type of statistical expectancy metric for player playability previously unattainable with only event data.

Using event data and tracking data from 28 real football games, these separate datasets have been synchronized to extract new information and context for passing events. This information was used to train and compare a statistical learning model for the xP metric with a model only trained on the previously known event data. The results indicate that the added tracking data information provides a significantly improved xP model especially in terms of understanding passing events and therefore making more realistic pass probability predictions. Despite clear improvement, there exist possibilities to further improve the xP model in regards to for example a more accurate data synchronization process as well as further improved feature engineering.

Moreover, the synchronized tracking and event data in combination with the improved xP model were used to develop a metric that describe player playability expectancy called xPlay. The new metric provides a simple and elegant way of measuring player playability and results of various implementations indicate that the metric can serve as a great tool in both player and team evaluation. Although promising results the metric is in need of more evaluation on a bigger scale.

Keywords: Event Data, Tracking Data, Statistical Learning, xP, xPlay

Acknowledgements

I would like to start of by thanking my primary supervisor at PlaymakerAI, Jesper Haglöf, who throughout this project has been able to regularly give valuable input and encouragement which helped massively in the execution of this thesis. Furthermore, I would like to massively thank Alex Jorge at PlaymakerAI for both providing relevant support and discussions whenever I needed it, as well as actually giving me the opportunity to write this thesis on one of my biggest interests and hobbies.

Lastly, I would like to thank my academic supervisor and examiner at Chalmers, Rebecka Jörnsten at the department of mathematical sciences, for both sparking my interest in data science with her course *Statistical Learning for Big Data*, and for providing relevant feedback and advice in regards to the thesis work.

Albin Steen, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BSS	Brier Skill Score
CATBoost	Categorical Boosting
ED	Event Data
FN	False Negatives
FP	False Positives
KPI	Key Performance Indicator
ROC-AUC	Receiver Operator Curve - Area Under the Curve
SVM	Support Vector Machine
TD	Tracking Data
TN	True Negatives
TP	True Positives
TTI	Time To Intercept
UMAP	Uniform Manifold Approximation and Projection
XGBoost	Extreme Gradient Boosting
xG	Expected Goals
xP	Expected Passes
xPlay	Expected Playability

Contents

List of Figures

List of Tables

1
Introduction

This section provides an introduction to the project in regards to what the important background to the project is and why the project is relevant. Furthermore, the overarching aim and limitations that has surrounded the project from start to finish is also declared and described. Lastly, the concrete research questions that the project attempts to answer are established.

1.1 Background

This section will outline the necessary information needed to understand the area that this project is carried out within, namely data analytics for football. The section also outlines why this project is carried out and what relevance it serves within the space of football data analytics.

1.1.1 Football and it's influence

Football is considered to have been the biggest sport in the world for a considerable period of time both in terms of the amount of people partaking in it as well as amount of people watching the sport. The biggest tournaments like the FIFA World Cup are in modern times regularly getting billions of viewers. The interest in the sport has naturally resulted in that there exist an impressive economic market surrounding it. Professional clubs in the top leagues can today expect to produce incredible multi-million sums in revenue every season just for competing at a high level.[28] This aspect has made the sport more competitive than ever as clubs are obsessed with finding ways to compete at the highest level and the most influential and prolific way that has appeared in the latest decade is using data.

1.1.2 Data analytics within football

In the last decade the usage of data within football has exploded. The usage of data has seeped into almost every aspect within a football club, whether it be using it for analyzing games, preventing injuries, scouting new players and so on. What has happened to modern day football is best described as a data revolution.

This revolution has provided several new ways to use data to gain insights and new information. The most popular and widely accepted *data usage* has come in the form of the development of so called ***Key Performance Indicators***, which henceforth

will be called ***KPIs***. A KPI is a type of metric that often in a single value tries to capture some aspect within football. The easiest way to further explain KPIs will be through examples and there are several KPIs that have become known and used not only within football clubs but are now also available and used by general football fans all over the world.

1.1.2.1 Influential KPIs

It would be wrong to not start this section with talking about the KPI ***xG - Expected Goals***. xG is the KPI that symbolizes the entire data revolution as it's a metric that has become wildly popular. The KPI xG is a metric that tries to measure the probability of a shot resulting in a goal. It is a simple metric calculated using data collection from historical shot measurements in combination with statistical learning methods. From the historical data certain features can be attained, these features usually regard things like the position of the shot, the angle towards goal, the distance to goal and which body part was used just to name a few. These features for all shots are then fed into a statistical learning model together with a binary target regarding if the shot turned into a goal or not. The statistical learning model will learn the relationship between shot features and the binary target and will after training be able to give a number between 0 and 1 representing it's confidence, in regards to the viewed historical data, that a certain shot results in a goal. This can then be further interpreted into a probability value that describes how likely it is to score that certain shot. A shot with 0.9 xG value is a shot that you would expect to turn into a goal 9 out of 10 times.[36]

Furthermore, another KPI that is very similar to xG is **xP - Expected Passes**. This KPI works in the same exact fashion as xG but instead of looking at successful shots it looks at successful passes, namely, xP uses historical data and statistical learning to determine the probability of a pass reaching the intended teammate.

1.1.2.2 Usage of KPIs

The KPIs that the data revolution has provided provides quite interesting information and can be used in several ways. KPIs like xG and xP that describes *expectancy* are particular useful in giving more context in analyzing performance both at team level and player level. A usual approach is to simply sum the probability values for all the shots or passes made from a team or a player during a game to get a goal or pass expectancy for the game. When analyzing a football match that finished 3 - 1 to the home team it would at surface level seem that the team who scored 3 goals played better. In contrast, the total summed xG values for the teams' shots might show 1,3 - 3,5 . This would suggest that the team who only scored one goal actually in total had chances and shots that *should* give at least 3 goals and furthermore the home team only had shots worth at least 1 goal. Now does this mean that the result is unfair or that the losing team actually played really well? The simple answer is not necessarily, but the xG score gives ***more context*** when analyzing the performance.

The same method can be used for xP but it can also be approached from a player perspective. If two players from a game both completed 40 passes but lets imagine the first player had a total summed xP value of 41.2 and the second player had an xP of 32.8. This suggests that the first player performed *as expected* regarding their passes while the second player actually performed *better than expected* and most likely were able to complete some very difficult passes.

All in all, the KPIs are very powerful in providing more context and more information that previously might not be known unless someone directly watched a whole game or player performance. Further, the KPIs allow for this complex context to be encapsulated into a single value. This is extremely useful when trying to make accurate analysis of performances of both players and teams in certain games as well as judge performances over a whole season.

1.1.3 Modern day challenges

There are several challenges regarding data analytics within football but most of them surround how to extract more and new information from the data. In the last years there has been advancements within artificial intelligence and specifically computer vision which has resulted in new data collection methods for football games.

There are two main divisions of data in football. The first and classical division is based on ***Event Data*** and will henceforth be called ***ED***. This data consists of usually manually annotated events from a football game and these events usually consists of shots, passes, tackles or interceptions, namely interesting events from a match. These events are manually tagged with several features like position, destination, if they were successful and so on. Moreover, as mentioned, the advancements in computer vision has opened up a new division of data that consists of spatiotemporal ***Tracking Data*** which henceforth will be called ***TD***. This data describes all players' as well as the ball's position at very small intervals during a complete football game. This entails that the data is big in size as well as being quite unstructured and complex. But this also entails that it is rich in information, specifically in information that the established ED does not contain.

In modern times TD has become of huge interest as it can aid in providing new and rich information that can both aid already established KPIs and analytics but also potentially prompt the creation of completely new KPIs and other analytical tools. This is precisely what this project will be concerning. In collaboration with the leading Swedish company for football data analytics PlaymakerAI, this project and master thesis will explore TD and how it can be integrated with their already established ED to improve existing KPIs and possibly create a new one.

1.2 Aim

The overall aim of this thesis is to conduct research in collaboration with the company PlaymakerAI of spatiotemporal tracking data from real football games and

use it to produce relevant and useful knowledge. The thesis explores this aim by two main objectives; *can spatiotemporal tracking data be utilized to improve already established statistical modeling in football* and *can new tracking data be utilized to develop completely new statistical modeling in football*. The primary objective of this thesis is the former of the mentioned objectives which is executed by exploring how to use the new spatiotemporal tracking data in order to enrich established information from ED. Further the thesis explores using the enriched data in combination with statistical learning to provide an improved KPI metric for predicting pass success namely an xP-model. To achieve this, the project further aims to thoroughly investigate and compare how an xP-model using the new data as well as a model using only the previously known event data compare in their predicted probability performance. Specifically, this thesis investigates how previously unknown aspects within the event data concerning different types of pressure, distances and pass intersections of defensive lines as well as distances and angles to opponents effect the statistical modeling of predicting pass outcomes. Lastly, the thesis also investigates the secondary main objective by viewing how the tracking data itself in combination with the newly developed xP model from the first thesis objective can be used to produce a completely new statistical KPI for measuring player playability.

1.3 Limitations

There are several established limitations in regards to the project that will be defined below.

1. ***Quantity of Data:*** The quantity of data was a limitation as the TD that PlaymakerAI supplied for this project was in fact data they themselves have acquired from an external source. There is a limited amount of games for which PlaymakerAI had acquired tracking data, more specifically 28 games. The TD itself was very large in size and contained a lot of different information, but when integrated with the ED that is a lot smaller, only smaller parts of the tracking data becomes relevant. There is only a certain amount of events within a game of football and therefore to train a model that is specified for a certain type of event it is in general good to have data for a lot of different games so that as much data as possible for that specific type of event can be extracted.

   As the amount of games that had tracking data was limited, this therefore limited the type of events that became relevant for modeling. For example goals are quite rare in football and therefore it would require a lot of games to train a statistical learning model for a metric like xG. Passes, on the other hand occurs a lot and therefore requires less amount of games. All in all, as the TD was limited this also limited which type of metric can be enriched and improved in a successful way for the project which is why xP became the metric of attention for this thesis.

2. ***Type of Data:*** The data used for the project was limited to certain leagues

and tournaments. Although differences between tournaments are not necessarily huge as the sport played is still the same, there are slightly different trends and patterns in terms of playing style, tactics and level within the different leagues therefore for more accurate modeling purposes it would've been better to have all games from the same tournament. In contrast, having data from varied tournaments gives the modeling more objectivity even though the accuracy might suffer slightly.

3. ***Time Frame:*** The project was constrained within a certain time frame which means that the amount of time that went into the different sections of the project have had to be fitted into that time frame. This entails that the aim and main objectives of the thesis were constructed in such a way that it ensured interesting and reasonable results within that time.

4. ***Thesis Assumptions:*** The thesis has employed several assumptions and generalizations in terms of certain aspects and hyperparameters employed in various sections of the methodology. This naturally limits the exact accuracy of certain methods but was also essential in ensuring that the thesis could be completed in the expected time frame.

1.4 Specific research questions

As mentioned in section 1.2, the main purpose for this project is to work with and research TD in regards to whether it can be integrated with ED to provide a better modeling of the xP KPI as well as provide the basis for a new type of playability KPI. Therefore, this project will attempt to answer the following research questions:

- ***RQ1:*** Can information from spatiotemporal tracking data be integrated into established event data to improve the xP KPI that describes the probability of completing a pass?

- ***RQ2:*** How does an xP model utilizing tracking data information differ in the specific outputted passing probabilities in comparison to a model only utilizing event data?

- ***RQ3:*** Can new knowledge in the form of a new type of player playability KPI be achieved by using information from tracking data?

2
Theory

In the following chapter the important background theory regarding this thesis is presented. The theory primarily aims to as clearly as possible outline and explain the important theoretical aspects needed to understand the implementations, motivations and possible assumptions presented within the project methodology. The focus theory-wise will be to cover the core concepts and methods used from statistical learning but also present some established related work to this thesis.

2.1 Statistical Learning

In this section the core methods used from areas such as unsupervised learning, supervised learning, feature importance and model evaluation will be presented and explained. The explanations mainly focuses on describing what each method does while also presenting the most relevant and vital mathematics without going into too much unnecessary detail.

2.1.1 Unsupervised Learning

Statistical learning is often divided into two separate but similar types, the first one is called *Supervised learning* and the second is called *Unsupervised learning*. Consider data consisting of observations $i = 1, \ldots, n$ where each observation has a related vector of features x_i. Supervised learning adds a further variable y_i which is usually referred to as a *target* or *response variable*. Unsupervised learning concerns itself with finding relationships and patterns in the data itself without any regards to a target variable.[21]

2.1.1.1 K-Means Clustering

K-means clustering is a type of unsupervised learning method that based on observation similarity sections the observations into different types or clusters. Within a cluster the observations have similar feature vectors but between different clusters the observations should ideally have as different feature vectors as possible. The exact execution to achieve this utilizes an iterative algorithm that has the objective of minimizing the feature variation within the clusters based on equation 2.1[21]:

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{2.1}$$

Here equation 2.1 describes a minimization of the sum of the average squared euclidean distances between cluster centers and the observations for that cluster. For a cluster C_k, the observations assigned to cluster k will be x_{ij} and the cluster center or rather centroid will be $x_{i'j}$. Further, here i indexes all the observations belonging to C_k and j is indexes the features for each observation.[21]

The iterative algorithm optimizing equation 2.1 has a simple procedure of starting with randomly assigning a set number of cluster centroids to random observations. Next each observations is assigned to a centroid based on which centroid is closest which creates clusters. Within these clusters the average feature vector is extracted which is used as new centroids. The cluster assignment is redone for the new centroids. These steps are iterated until some stopping criteria is reached usually based on a iteration number threshold.[21]

The main disadvantages with K-means is that it does not find the best number of clusters to section the data itself and the cluster number must be specified before running the algorithm. Further, the local optimization is dependent on the initial centroid assignment which means that the final result can vary between trials. Lastly, the method is distance based which means that it is sensitive to outliers, noisy data and data with features of different types or scales. Furthermore, it assumes spherical cluster shapes which might not be ideal for certain data types that could be clustered better using clusters of more complex shapes. In contrast, the advantages are that K-means is simple, quick and given that the number of appropriate clusters is known beforehand in combination with data that has numerical features on the same scale, the method will usually provide very good results.[21]

2.1.1.2 Uniform Manifold Approximation and Projection - UMAP

Uniform Manifold Approximation and Projection or UMAP for short is a method that projects high dimensional data into a lower dimension while keeping a similar structure to the data in the higher dimension. The method falls under dimension reduction techniques and within that area it serves more as a visualization technique rather than being something used for data pre-processing like principal component analysis (PCA).[25]

UMAP is based on a method that models the data structure in the high dimension using a type of graph structure and then it tries to create a similar graph in a lower dimension by optimizing a loss function that measures the difference in graph structures. UMAP does this by extending a radius in the high dimensional space for each observation and connects observations which radius overlap. By doing this UMAP creates so called *simplices* that are k-dimensional objects based on the connection of k+1 observations and these simplices can together model the data topology. Further, UMAP does not use and extends the same exact radius for all observations, by looking at the distance of each observation to a specified neighbor number, UMAP can see if a point is more or less isolated which prompts the use of a bigger or smaller radius and UMAP also because of this attaches a probability to each connection between observations. Next, UMAP focuses on the created simplices that are either 0

or 1-dimensional as they can be used within the lower dimensional graph. Lastly, UMAP creates a first initialization of a lower dimensional graph and uses a cross entropy loss between the connected simplices in initial graph in the high dimensional space and the simplices in the new projected graph in lower dimensional space to optimize the latter graph to resemble the same structure and topology as for the high dimensional graph.[15][16] For more information on the exact structure and detailed mathematics of the UMAP algorithm the reader is referred to [25].

The disadvantages of UMAP is that it is dependent on tuning hyperparameters that regulate the method execution. The most important hyperparameter is "n_neighbor" which regulates if the method should capture more of the local or global structure of the data. A low value of the hyperparameter suggest that the method will only consider a smaller number of neighbors in total which therefore entails that neighbors situated further away will not be used when creating the graph. A higher value will instead do the opposite. The advantage is that the method, if correctly tuned, can give a very good approximate visualization of the data structure which aids in understanding the data better.[16]

2.1.2 Supervised Learning

Supervised learning concerns as previously mentioned working with data that has a target or response variable. Specifically it concerns methods that can use the feature vectors to learn information that can be used to predict the targets. This entails that supervised learning has a ground truth which can be used directly to quantify how well methods work. Relevant methods within the area will be presented in coming subsections with a lot of focus being put on decision tree based models. [21]

Before outlining the different models there exist an important aspect of supervised learning that needs to be outlined. Within supervised learning there exist a concept called *bias-variance trade-off*. Bias describes when supervised methods make too simple assumptions about the data which results in that the methods cannot learn the relevant information which results in overall poor performance when predicting the targets. Variance in contrast concerns models that learn the the data information *too well* which results in that the models memorize the data and predictions instead of learning the underlying relationships. Variance further entails that when the model later encounters completely new data it cannot generlize to it and therefore performs poorly. High bias is also called *underfitting* to the data while high variance is called *overfitting* to the data. Lastly, the trade-off between these is based on when the supervised learning method complexity increases, in terms of making less simple assumptions and better capturing complex relationships, the bias will decrease until a certain point where the variance instead starts to increase. This means that within supervised learning an important aspect is to balance these two to ensure as little under- and overfitting as possible.[21]

2.1.2.1 CART

Classification and Regression Trees or CART is a supervised learning model used to predict both continuous (regression) or integer (classification) targets. The model is based on the concept of decision trees which are a type of method that splits the high dimensional feature space into different regions, a bit similar to clustering, and then within each region either uses the average target for regression or the majority target for classification or the proportion of targets for probability outputs. The creation of these regions is based on a set of rules regarding different feature values, for example a rule could be to section the data into two regions based on if observations has smaller or bigger value than a certain threshold value for a certain feature. Next, the two new regions can employ further rules of their own to create even more defined regions. Every region created is called a node in the tree with the nodes being used for splitting called internal nodes while the final last nodes and regions used for prediction being called leaf nodes. The strategy to pick splitting rules that creates regions that provides good predictions is based on *recursive binary splitting* which chooses a certain feature split that minimize a loss function. One common loss function called *gini index* is structured according to 2.2 [21]:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2.2}$$

Equation 2.2 is specified for classification trees and uses the region m proportion of target k to measure the the total variance over the different targets. A low gini index value indicates that the region or node is pure meaning that it almost exclusively holds observations with one target type. The gini index can therefore be used to compare different splits in regards to which node purity they create and the feature split that gives the lowest gini index will be the chosen split. Further, when building the trees this splitting procedure is usually iterated for all regions until some stopping criterion is met, examples are a certain tree depth or a threshold for the gini index.[21]

The disadvantages with the CART models is that they can tend to become too detailed which causes overfitting. There exist so called regularization strategies which helps against overfitting like for example set restrictive tree depth thresholds or remove created branches in the tree that does not necessarily contribute that much. The advantage is that the tree structure is stable for data using features of different types and scales, furthermore they are quite intuitive and interpretable while providing good prediction performance.[21]

2.1.2.2 Random Forest

Random Forest is a supervised learning model that utilizes decision trees in an ensemble namely uses several decision trees at the same time. Further, the model also uses a strategy called *bagging* which entail using a random sample of the data where each observation was sampled with equal probability and with replacement. The strategy is executed several times to create several samples $b = 1, 2, \ldots, B$ and

each of the samples is used to train a separate decision tree. Lastly, when predicting an observation's target it is run through all decision trees and the tree predictions is averaged according to equation 2.3 [21]:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x) \tag{2.3}$$

In equation 2.3 $\hat{f}^{*b}(x)$ is the prediction of a specific tree within the ensemble train on the specific sample b. For a classification task the equation is instead substituted with a majority vote between the trees and for probability outputs the average of all tree target proportion is used. Further, Random Forest also utilizes a subset of features from a random sample when building the trees by creating new splits and usually the sample size m is defined by $m \approx \sqrt{p}$ where p is the number of features. The bagging strategy and the feature subset sample both serve the purpose of dealing with overfitting that decision trees can suffer from. By using an ensemble of trees trained on different parts of the data the ensemble consists of several trees with low bias and perhaps higher variance. By then using them in an averaged fashion this variance is decreased while keeping the low bias. Further by only consider random samples of features when building the trees it is made sure that the trees are different and do no correlate with each other which further improves generalization when averaging over the trees.[21]

The main disadvantages of Random Forest is that it's less interpretable than a single decision tree and the model can also be more computationally intensive because of training several trees for the ensemble. Random Forest also usually require some hyperparameter optimization and tuning in relation to parameters for number of trees used, the maximum tree depth, loss function used and how many samples to use to name some. The advantages is that the model keeps the advantages of CART models with added protection against overfitting. Moreover, the ensemble approach also protects against outliers and noisy data. While less interpretable than a single CART model it still from a statistical learning perspective is one of the more interpretable models and there exist several methods for extracting feature importance scores from the model.

2.1.2.3 Extreme Gradient Boosting - XGBoost

Extreme Gradient Boosting or XGBoost is a supervised learning model that also falls to the ensemble category by, similar to Random Forest, utilizing several decision trees. In contrast, XGBoost does not use the bagging strategy and instead uses random sampling without replacements for each new tree as well as building the trees sequentially. This strategy is called *boosting* or in XGBoost's case *gradient boosting* where each new tree is a weak learner, namely a shallow decision tree, that is trained to correct the mistakes of the trees built before.[21]

Gradient boosting in detail tries to build an approximate function $\hat{F}(x_i)$ by building en ensemble of sequentially trained weak learners that using the observation features can predict the target. These weak learners are built be minimizing a certain loss

function $L(y_i, \hat{F}(x_i))$. The approximate function and the ensemble is structured according to a weighted sum of the weak learners according to equation 2.4 [11]:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h_m(\mathbf{x}) \tag{2.4}$$

Here, an approximate function using m weak learners is the sum of the previous weak learners and a the new weak learner that is also weighted by ρ_m. Each new weak learner as well as weight for that weak learner is developed to minimize a loss function which is given by equation 2.5 [11]:

$$(\rho_m, h_m(\mathbf{x})) = \arg\min_{\rho,h} \sum_{i=1}^{N} L\left(y_i, F_{m-1}(x_i) + \rho\, h(x_i)\right) \tag{2.5}$$

The exact loss function L used depends on the supervised learning problem, there exist several different functions that can be implemented. For example when predicting a binary target probability the model uses the log loss function and equation 2.4 then predicts the log odds for the positive target which is fed into a logistic function to get a probability of an observation belonging to the positive target class. Moreover, gradient boosting solves the optimization problem proposed in equation 2.5 by training the new weak learners on the mistakes or rather errors by the previous approximate function and ensemble. These errors are given by the pseudo-residuals of the previous ensemble according to equation 2.6[11]:

$$r_{m,i} = -\left[\frac{\partial L(y_i, F(\mathbf{x}))}{\partial F(\mathbf{x})}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})} \tag{2.6}$$

Here, equation 2.6 describes the pseudo-residual $r_{m,i}$ for observation i which the new weak learner m should fit to. Further the weight ρ for that new weak learner can be optimized by minimizing the result of equation 2.5 using the new weak learner. A problem with the regular gradient boosting is that it can easily suffer from overfitting if the residuals are captured too well by the weak learners. XGBoost therefore makes some additional tweaks to the regular gradient boosting by introducing another type of loss function in equations 2.7-2.8 [11][14]:

$$L_{\text{xgb}} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \tag{2.7}$$

$$\Omega(h) = \gamma T + \frac{1}{2}\lambda ||w||^2 \tag{2.8}$$

Here the new loss function uses a new term that regulates how complex the weak learners can become. T and w is the number of leaf nodes and the predicted leaf node values respectively and therefore γ becomes a hyperparameter for regulating the weak learner size while the λ is a coefficient that regulates strong leaf node predictions. Another regularization hyperparameter called *shrinkage* is also employed by substituting it with the ρ_m in equation 2.4 and therefore effectively acting as a constant learning rate.[11][14] XGBoost also uses a couple of strategies to more quickly optimize the loss function as well as building trees faster. First, instead of

using first order gradient of the loss function as residuals it uses the second order Taylor expansion around $F(x_i)$ in equation 2.7. This approximation can be used for quick and accurate optimization while also being used as a metric for getting the best feature splits when building trees by looking at splits that decrease the loss the most.[14] Moreover, when finding the best feature splits XGBoost does not iterate over all possible features for each split but instead, similar to Random Forest, uses a subset of features. In contrast, these features are not selected randomly but the features are suggested together with a splitting value based on percentiles in the feature distributions.[11] For more detailed and mathematical implementations of XGBoost the reader is referred to the original paper [14].

The disadvantages of XGBoost is that it requires proper tuning and optimization of different regularization hyperparameters in regards to how many observations to sample for each weak learner as well as how many features to sample for each split, the shrinkage and λ parameters, restrictive parameters for tree depth and complexity and minimum reduction in loss needed to split an internal node just to name a few. In contrast, XGBoost provides state-of-the-art accuracy while being computationally fast and also combating overfitting. Like Random Forest it also provides built in methods for extracting clear feature importance scores which provides interpretability. As it uses decision trees and an ensembling approach, it also gets the benefits of being robust against tabular data with features of different types and scales as well as robust against overfitting, noise and outliers.

2.1.2.4 Categorical Boosting - CATBoost

Categorical Boosting or CATBoost is a supervised learning model that works in a very similar fashion to XGBoost with some added tweaks in *ordered target statistics* and *ordered boosting*. The former is a method for internally dealing with categorical features and the latter is a method for improving the gradient boosting methodology. Categorical features concerns features that does not use numerical values but categories and to be able to use this type of features the categories need to be represented numerically. As the categories does not hold numeric properties it is naive to replace them with random integers. One method is target statistics or target encoding which means to replace a category with the average target value for that category. Although target encoding in this form gives so called *target leakage* which means that the training data contains information that a model is supposed to predict. To restrict this the following form of target encoding is usually employed as seen in equation 2.9 [29]:

$$\hat{x}_k^{(i)} = \frac{\sum\limits_{x_j \in D_k} \mathbf{1}_{\{x_j^{(i)} = x_k^{(i)}\}} \cdot y_j + ap}{\sum\limits_{x_j \in D_k} \mathbf{1}_{\{x_j^{(i)} = x_k^{(i)}\}} + a} \tag{2.9}$$

Here $x_k^{(i)}$ is the categorical feature type for feature i and observation k, $\hat{x}_k^{(i)}$ is new encoded value for feature i and observation k, D_k is a subset of the entire data excluding x_k, p is a prior value which is usually the mean response variable over the

entire dataset and a is a smoothing factor which controls the weight of the prior. The use of the prior is to smooth the encoding and prevent overfitting especially for cases which have a very rare feature type. Even though this equation helps against target leakage it still persists somewhat as all response variables are used for each other which mean there exists some interdependence and subtle target leakage.

CATBoost instead uses ordered target statistics that is inspired by online learning as it treats the observations in a categorical feature in a sequential manner. For each new boosting iteration it creates a random permutation σ, namely reordering of the observations, and it then uses $D_k = \{x_j : \sigma(j) < \sigma(k)\}$ in equation 2.9 which entails only using observations placed before the current encoding observation for the target encoding. This mimics real-time test scenario better by using and learning from the data in a sequential manner which prevents target leakage better and therefore also prevents overfitting.[29]

Moreover, CATBoost employs ordered boosting which mitigates data leakage that exist within the regular gradient boosting methodology. As outlined in section 2.1.2.1, predictions in decision trees are done using all the observations within a certain region. Further, this entails that when training gradient boosting models, the predictions used for calculating the residual for an observation i in equation 2.6 uses all observations within the regions that observation i belongs to including observation i itself. As the observation residuals in the next step are used as new targets for the next weak learner this causes biased residuals that in turn results in interdependence between predictions and targets for the weak learners which hurts the generalization capabilities. To combat this the ordered boosting method makes a slight change to the standard gradient boosting method by utilizing the same permutation strategy as from the ordered target statistics. For each boosting iteration, namely when a new weak learner is trained, a new permutation σ of the observations is extracted. Next, in equation 2.6, instead of using $F_{m-1}(\mathbf{x})$, an approximate function $F^k_{m-1}(\mathbf{x})$ is used which only utilize a subset of observations $D_k = \{x_j : \sigma(j) < \sigma(k)\}$. This means that for an observation k it's residual is only based on a prediction utilizing solely the observations placed before itself in the permutation σ. This ensures non-biased residual targets for the weak learners which improves generalization. Lastly, CATBoost also only use so called symmetric trees which are trees that use the exact same splitting rule at the same node level in the tree which results in that the tree has a balanced and symmetric structure. These trees are weaker than regular decision trees in terms of often having slightly higher bias although slightly lower variance. Using symmetric trees helps further against overfitting and they also make predictions faster. For further details on the CATBoost algorithm the reader is referred to the original paper [29].

The main disadvantage with CATBoost is that the internal ordering principles makes it slower than other gradient boosting methods like XGBoost. In contrast, the ordering principles makes the model robust against overfitting from the start and therefore requires less hyperparameter tuning. Although relevant hyperparameters for the method are very similar to XGBoost in number of boosting iterations, max

depth for weak learners as well as the shrinkage and λ parameters to name some. The ordered target statistics also makes it one of the best supervised models in terms of dealing with different features types without pre-processing. Lastly, similar to other decision tree ensemble methods it has built in feature importance scores, is very robust against noise and outliers.

2.1.2.5 Logistic Regression

Logistic Regression is a supervised learning model that uses the features x_i for an observation through a linear combination of them in $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$. Next, the model passes the linear combination to a logistic function which maps the combination to a value between 0 and 1 giving a probability of the observation belonging to a certain target. In a binary classification setting a threshold of 0.5 is used to determine if an observation will be predicted to the targets 1 or 0. The logistic function for a binary classification is given by equation 2.10 [21]:

$$P(y = 1|\mathbf{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} \tag{2.10}$$

The coefficients β used for the linear combination of the feature values are calculated by maximizing the probability of observing the actual targets by using a maximum likelihood estimation (MLE).[21] In practice logistic regression takes the logarithm of the likelihood instead and minimizes according to equation:

$$\begin{aligned}\ell(\beta) = -\sum_{i=1}^{n} \Bigg[y^{(i)} \log \left(\frac{1}{1 + e^{-(\beta_0+\beta_1 x_1^{(i)}+\cdots+\beta_p x_p^{(i)})}} \right) \\ + (1 - y^{(i)}) \log \left(1 - \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1^{(i)}+\cdots+\beta_p x_p^{(i)})}} \right) \Bigg]\end{aligned} \tag{2.11}$$

Here the equation describes the sum of the log-likelihood for all observations i which is minimized to get the best coefficients that maximizes the likelihood of observing the correct targets. Usually a regularization term that regulates the size of coefficients is also added to the negative log-likelihood sum and this term is chosen as a hyperparameter for the method. The problems with logistic regression is that it according to equation 2.10 assumes a linear relation between the log-odds of the response variable and the features as mentioned which creates difficulties when the data has complex and non-linear relationships with the response variable. Furthermore, the assumption also makes it sensitive to multicollinearity (when several features correlate linearly with each other) in terms of model interpretability as the coefficients becomes less trustworthy. The linear assumption for the log-odds also makes the model sensitive to outliers that can distort the linear decision boundaries significantly. Usual hyperparameters for the method concerns the type of regularization employed, the strength of the regularization as well as which optimizer to use.

2.1.2.6 Support Vector Machines - SVM

Support Vector Machines or SVM is a supervised learning model that uses a hyperplane to separate observations of different targets and making predictions based

on which side of the hyperplane an observation lies. For data that lives in the p dimensional space a hyperplane would be a specific decision boundary in the p-1 dimensional space defined by[21]:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{2.12}$$

Further, imagine binary targets $y_i \in \{-1, 1\}$, a hyper plane separating these classes would be defined by the following equation[21]:

$$y_i\Big(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\Big) > 0 \quad \text{for all } i = 1, \ldots, n \tag{2.13}$$

Given this hyperplane a new observation x^* can be classified using the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^*$. To find this hyperplane the minimum margin M to the hyperplane for the closest observations of each target class called the support vectors is maximized according to[21]:

$$\begin{aligned} &\max_{\substack{\beta_0,\beta_1,\ldots,\beta_p,\\ \epsilon_1,\ldots,\epsilon_n,\, M}} M \\ &\text{s.t.} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \\ &\qquad y_i\Big(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\Big) \geq M\left(1 - \epsilon_i\right), \quad i = 1, \ldots, n, \\ &\qquad \epsilon_i \geq 0, \quad i = 1, \ldots, n, \\ &\qquad \sum_{i=1}^{n} \epsilon_i \leq C. \end{aligned} \tag{2.14}$$

Here ϵ_i are called slack variables that describes the placement of observation i in relation to the margin M as well as the hyperplane. As two target types rarely are perfectly separable the slack variables in combination with the tuning hyperparameter C determines how many observations that are allowed to break the margin or the hyperplane separation.[21] This optimization is solved using techniques involving reformulations, taking the lagrangian and estimating the coefficients and slack variables using derivatives. These steps goes outside of the relevant theoretical bounds needed for this thesis but for more detailed information the reader is referred to [12]. In the end the resulting hyperplane function can be given in the so called dual form by[21]:

$$f(x^*) = \beta_0 + \sum_{i \in S}^{n} \alpha_i \langle x^*, x_i \rangle \tag{2.15}$$

Here equation 2.15 models a decision function using the weighted sum of feature vector similarity between a new test observation x and support vector observations x_i as well as the hyperplane interception term β_0. The weights α_i are langragian multipliers incorporating the targets y_i. When solving the optimization only the support vectors end up with non-zero weights which is why they solely contribute to

the final decision function. Further, similar to before using the sign of the decision function gives a classification of the observation x^*. The problem with equation 2.15 is that it is a linear classifier and therefore struggles when the observations are not linearly separable. To adjust for this the SVM use something called kernels that generalizes the inner product in equation 2.15 by $K(x, x_i)$. The following polynomial kernel generalization $K(x, x_i) = \left(1 + \sum_{j=1}^{p} x_j x_{ij}\right)^d$ with $d > 1$ allows equation 2.15 to work in a higher dimensional space where a linear decision boundary may be found. There exists several different types of kernels that allows different amount of flexible decision boundaries for the support vector classifier.[21]

SVMs have good prediction performance for non-linear data with high dimensionality as well as being adaptable with different types of kernels. The disadvantages are that it requires proper hyperparameter tuning to get the best kernel as well as make sure it does not overfit by optimizing hyperparameters regarding the kernel choice and regularization parameters used within the optimization of the weights. As it is distance based the model becomes sensitive to features with different scales as well as noise and outliers. They also does not provide much interpretability.

2.1.3 Feature Selection and Importance

The area of feature selection and importance concerns quantifying feature correlation and relevance for the statistical learning task. There exists several ways to investigate that can look at both if feature information is relevant in regards to the response variable but also if the information provided is unique within the data. The relevant methods to this are explained in coming subsections.

2.1.3.1 Filtering Methods

Correlation Coefficients: Correlation Coefficients are numbers that describe strength and direction in terms of how two features correlate in term of the information they contain. ***Pearson correlation*** is a type of correlation that is focused on linear information correlation between features and assumes normality of the feature distributions as well as that they only contain the numerical continuous feature type. Moreover, ***Spearman correlation*** is another type of correlation that instead looks at monotonic relationships and further makes no distribution assumptions although the features are expected to be numerical continuous or ordinal in type. Both correlation types give values in the -1 to 1 range where very high or very low values indicates a strong correlation in either positive or negative direction.[33] Both correlation types can be used to also find correlation to a target or response variable but can suffer, especially Pearson correlation, if the target is unbalanced namely one type of target dominates the observations.

Mutual Information: Mutual Information is another method for quantifying correlation. Mutual information measures how much information in one feature can help describe the information in the other features. Mutual information can capture correlations that may not be either linear or monotonic while making virtually no

significant assumptions regarding the involved features. Mutual Information utilizes the following equation 2.16 [22]:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{2.16}$$

The equation describes the mutual information ($I(X;Y)$) between features X, Y by looking at the entropy difference between the sum of the entropies of the respective feature distributions and the entropy of their joint distribution. The distribution entropy in itself is a measure that describes the spread of the feature, namely a feature that has uniform distribution would be considered to have high entropy. Equation 2.16 describes the theoretical implementation of mutual information but the practical calculation of the entropy terms usually uses some type of generalization or estimation with one alternative being a k-nearest neighbor approach which can estimate the density of feature values. The mutual information takes the value 0 if the features are not correlated at all while an increasing value from 0 indicates stronger correlation. For more exact mathematical implementation of mutual information and the k-nearest neighbor approach see the paper [22].

2.1.3.2 Embedded Methods

Embedded methods concerns methods that gives feature importance directly related to how a statistical learning utilizes the features. The most common model that has embedded methods for feature importance are tree-based models but there also exist other models where some type of feature importance can be extracted like for example the coefficients from Logistic Regression.

The tree-based feature importance methods usually relate to some type of score for each feature in terms of how important or how much information it provided during the tree building process. There are different types of metrics to measure this and they all have their advantages and disadvantages in what they tell and do not tell about the features. One important example for this thesis is the CATBoost built in feature importance score ***PredictionValuesChange***. PredictionValuesChange is a measure for how much the prediction in a tree changes if a value of a specific feature changes. The principle of the method is that when a specific feature is used in a split it will separate the training observations into two temporary or final leaf nodes. Remembering how CART models work, these leaf nodes will have internal predictions which are the average response variable for the training observations. If these predictions are compared between the new leaf nodes and the difference is large the feature can be considered to be important. The exact equation used is given by equation 2.18 [1]:

$$\text{feature\_importance}_F = \sum_{\text{trees, leaves using } F} \Big[(v_1 - \text{avr})^2 \cdot c_1 + (v_2 - \text{avr})^2 \cdot c_2\Big] \tag{2.17}$$

$$\text{where avr} = \frac{v_1 \cdot c_1 + v_2 \cdot c_2}{c_1 + c_2} \tag{2.18}$$

Here equation 2.18 describes that the feature importance score for a feature F is the sum for all splits in all trees using feature F for a split where the sum is over the

given the squared difference between the average response variable in the left leaf node v_1 and the weighted average response variable in both nodes avr multiplied with the number of observations in the left leaf node c_1, this is then added with the same calculation for the right leaf node using v_2, c_2. Further, all the feature importance scores are then normalized so they sum up to 100.[1]

The advantages of embedded methods like PredictionValuesChange is that they describe features effect directly tied to the inner workings of the model. Most embedded methods are usually global meaning that they are averaged over all several splits and trees and can therefore not necessarily be used to explain individual predictions. Another problem with several embedded methods including PredictionValuesChange is that they can both overestimate and underestimate feature importance. Consider two features that describe similar important information, if one of them randomly happens to be chosen early in the tree there is a bigger possibility of driving a bigger prediction change, which gives that feature more importance even though the other feature potentially could have been used. Furthermore, if they are correlated then the need for the other feature also becomes less which underestimates the information and importance for that feature even more. All in all, the embedded methods provide great insight but not the whole picture and therefore other methods should complement the embedded methods for complete feature importance.

2.1.3.3 Model Agnostic Wrapper Methods

The Model Agnostic Wrapper Methods is a type of wrapper method, namely a feature importance/selection method utilizing model test performance as a scoring metric, that is agnostic to the type of statistical learning model.

One example of a model agnostic wrapper method is ***Permutation Importance*** which takes a trained statistical learning model as well as a test set with observations. Next, by predicting the test observations response variables a specified baseline score for a certain evaluation metric is established. It then permutes a certain feature's values among the observations and recalculates the same metric and views potential decrease in performance. It does this for all features usually several times and then averages the performance results.[3] The advantages of the permutation importance is that it directly correlates features with effect on test performance. Further, permutation importance will not provide high scores for features that have information that other features also provide either alone or in combinations with each other, namely a feature that is permuted might lose it's information but if the same information is available through other features ultimately it might not matter that much and the model might still perform well. This can help with providing perspective in relation to the possible over- and underestimation that embedded methods can give. In contrast, the same principle can also be a disadvantage if permutation importance alone is used for feature importance as the principle will score features providing similar information very low even though the information the features hold is important. The method is also very slow especially on large data sets and also only works globally like embedded methods and therefore cannot explain feature importance for individual predictions.

Another wrapper method is ***Shapley Additive Explanation or SHAP-values***. A SHAP value is specific to a feature and an observation and measures the contribution that the specific feature value made to the prediction of the observation across all possible feature subsets. This is done according to the following equation 2.19 [23]:

$$\phi_i(f,x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} \left[f_x(S \cup \{i\}) - f_x(S) \right] \tag{2.19}$$

Here $\phi_i(f,x)$ is the SHAP-value for a feature i based on a model f and an observation x. Further, it calculates this by using variables for a subset of features not including feature i by $S \subseteq F \setminus \{i\}$ where F is the set of all features. Moreover M is the total number of features available, $f_x(S)$ is the model prediction of observation x with only the features in the subset S and $f_x(S \cup \{i\}$ is the model predication of observation x when feature i is added to the feature subset S. In simple terms equation 2.19 aggregates and sums the difference in prediction when feature i is added to a subset of features S and also weights this difference depending on the size of the feature subset S.[23]

SHAP-values are powerful in terms of feature importance as they similar to permutation importance are model agnostic and directly translates importance in terms of test predictions. In contrast, it works on both global and local scale as each value is assigned to a single observation and then the values can be aggregated to give the globally most important features. This also provides the possibility through plot techniques to view the top features while also seeing the effect on individual predictions. Adding to this, the SHAP-values also provide direction, namely in which direction the observation prediction is dragged by the feature which neither permutation importance or embedded methods usually does. Lastly, as it operates over all possible subsets it is less prone to suffer from correlated features as permutation importance does and therefore generally better can capture features with important information even though the same information might appear elsewhere. The disadvantages is similar to permutation importance as it can be computational heavy for larger datasets and can still underestimate feature importance if a very large number of features correlates with each other.

2.1.4 Model Evaluation Theory

Model Evaluation concerns methods for analyzing the performance of statistical learning models. Specifically this section outlines the most relevant supervised learning evaluation metrics for this thesis. There are two main groups of supervised learning evaluation methods where the first one is threshold based and the second one is non-threshold based.

2.1.4.1 Threshold Based Metrics

The general and most simple metrics in terms of classification are based on a threshold and the prediction results from a statistical learning model. Usually supervised model predictions in a binary setting can be turned into a target positive target probability between 0 and 1. Using a threshold, usually set to 0.5, the model makes a prediction for one of the binary targets by assigning a prediction to the positive target 1 if the probability is higher than the threshold and negative target 0 otherwise. Using the predicted targets four important aspects can be extracted in *True Positive or TP - the predicted positive target is correct, False Positive or FP - the predicted positive target is incorrect, True Negative or TN - the predicted negative target is correct and False Negatives or FN - the predicted negative target is incorrect.* By sectioning all test observation predictions into these sections they can then be used to provide the following evaluation metrics [30]:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \tag{2.20}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.21}$$

These equations outlines precision, recall and F1 for the positive target, it is possible to calculate the same metrics but for the negative target class as well. Here precision measures how precise or accurate a model is when predicting a certain target class, Recall measure how much of a target class that the model correctly identifies. Further, Accuracy simple shows the general percentage of correct predictions and F1-score is the harmonic mean of the precision and recall scores giving a somewhat general sense of how the model balances these. All of the scores takes values between 0 and 1 where scores closer to 1 is better.[30] Moreover, there also exist so called *macro* versions of precision, recall and F1 scores which are just the non weighted average of these scores over all possible target classes.

2.1.4.2 Threshold Independent Metrics

The threshold independent metrics does in contrast not utilize and require a set threshold for predictions and one important example of threshold independent metrics is the ROC-AUC score. This score is calculated by comparing the true positive rate (recall) and the false positive rate $= \frac{FP}{FP+TN}$ over several different thresholds. For each threshold a point can be plotted with the recall value on the y-axis and the false positive rate on the x-axis and the curve that is produced is called the ROC or receiver operating curve. The curve in simple terms explains how well the model differentiate between the different targets and the closer the curve is to the top left corner the better. Further, to measure how well the curve is to that corner the area under the curve - AUC can be used as it will give a value between 0 and 1 where 1 indicates that the curve is perfectly in the top left corner. Further, the final ROC-AUC value can be interpreted as the probability of a randomly chosen positive target observation being ranked higher in predicted probability than a randomly chosen negative target observation.[30]

Another threshold independent metric is the Brier Skill Score or BSS. BSS uses the regular Brier Score which measure the mean squared error between target and the predicted probability according to the following equation in a binary classification setting [32]:

$$B(y,p) = n^{-1}\sum_{i=1}^{n}(y_i - p_i)^2 \tag{2.22}$$

In this equation y_i is the observation target and p_i is the predicted observation probability of belonging to the positive target class 1. The brier score takes a value between 0 and 1 where a lower value is better.[32] Moreover, by establishing the brier score for a so called baseline model, usually a model that simply predicts a constant probability equal to the target class 1 proportion, the brier score for the actual trained model can be divided with the baseline score and then subtract this value from 1. This gives a score between 1 and negative infinity that shows if the trained model performs better or worse that a baseline model in terms of brier score and is called the BSS. The best value would be 1 while a value below 0 indicates that the trained model performs worse than the baseline. The brier score and the BSS measures how accurate in terms of prediction as well as how confident a model is in the predicted probabilities and the BSS also shows how a trained model compares in regards to this to a baseline model.

2.1.4.3 Probability Scoring and Calibration

Probability scoring or calibration is an area that concerns making sure that the predicted probabilities from a statistical learning model are reasonable and realistic. Neither threshold dependent or some threshold independent metrics quantifies if the predicted probabilities used for determining the metrics are actually accurate. A predicted probability of 0.53 and 0.99 is the same through a classification perspective but if the actual probabilities themselves matter it becomes important to quantify how well the individual probabilities reflect reality.

One possible method to get some sense of this aspect for a binary classification problem is by using a calibration curve also called a reliability diagram. By using probability ranges or rather bins which the probability predictions can be divided into it is then possible to compare the average positive target probability prediction in each bin with the proportion of the positive target in each bin. These values can be plotted using the average predicted probability on the x-axis and the average positive target proportion on the y-axis. The line created by the values will be a straight increasing diagonal line with slope 1 if the probabilities are well calibrated. [26]

2.2 Related Work

The football data revolution has been underway for some time which has resulted in established published work regarding football data analytics. For this thesis that concerns itself with the usage of TD in the realm of passing metrics as well as

the creation of a playability KPI, there exist relevant related work covering similar aspects.

2.2.1 Related work on using TD for passing metrics

To name some of the most relevant and interesting related work one example is the "xPass 360: Upgrading Expected Pass Models" by Sharpe.C for the analytics company Hudl Statsbomb. The article outlines an approach for improving the xPass KPI metric, which is the same as the xP KPI, by incorporating positional information from TD. It specifies that by utilizing feature engineering to create features capturing positioning of opponents in close proximation to the pass as well as other "soft pressure" metrics, they can create new features that in combination with statistical learning modeling improves an xP model both in terms of raw performance but also in calibration of probabilities.[34] Moreover, Robberechts.P et al in their paper "un-xPass: Measuring Soccer Player's Creativity", shows another approach of using TD regarding passing metrics. Similarly this paper uses statistical learning with a handcrafted feature dataset from TD although the purpose and end result is to create a new KPI measuring pass creativity.[31]

Specifically for this thesis the most related work is the paper by Anzer.G and Bauer.P - "Expected passes".[9] The main part of this paper regards the exact same main subject that this thesis also explores in using TD to create a better performing xP model and therefore this thesis has used that work as a main source of inspiration for the exploration of this subject. The methodology by Anzer.G and Bauer.P is based on four main sections:

1. ***An algorithm for synchronization of ED and TD***
2. ***A target estimation system for unsuccessful passes***
3. ***An xP model utilizing XGBoost and a hand crafted feature dataset***
4. ***A blocking model giving probabilities of a pass being immediately intercepted***

There are many important differences and distinctions between the work in "Expected passes" and the work in this thesis regarding the methodology, structure and aims of the two works which are outlined below in the contributions that this thesis makes:

- The settings and data used are completely different. The "Expected passes" paper produced for research purposes utilizes a passing TD dataset of 840 386 passes from 918 German first division matches where they exclude so called "fair-play" passes (where the player willingly gives up the ball), throw-ins as well as immediately blocked passes.[9] This thesis was developed in an industrial setting in collaboration with PlaymakerAI in a request to improve the data analytics on their platform, furthermore this thesis uses TD from a mere 28 games from different leagues and utilizes all type of passes.

- As the exact synchronization algorithm is not defined in precise detail by

Anzer.G and Bauer.P, the inspiration for this thesis has been the overall principles the algorithm uses. The final algorithm utilized for this paper has some distinct differences in the exact structure which are explained in detail in the section 3.1.2. Furthermore, the synchronization for this thesis utilizes post processing for further refinement which is different.

- Although this thesis also employs the physics based ball trajectory model proposed by Spearman et al, the overall method for intended pass location this thesis use is based on a very different and simplified approach in comparison to the one developed in the second main section of the work by Anzer.G and Bauer.P.[9]

- The information used from the TD to create the features and therefore the actual final feature dataset is different and novel. This thesis puts more focus on features tending to the concept of pressure for a passing situation and tries to capture this in a more varied and complex way utilizing several different pressing features. Furthermore, this thesis also explores the concept of defensive lines and their possible effect on the probability in a given passing situation.

- This thesis presents an exploratory data analysis of the feature dataset. Furthermore, this thesis utilizes the dataset in a different way by employing cross validation procedures when executing hyperparameter optimization, model selection and final xP model training.

- This thesis provides a more extensive feature importance evaluation and also a feature importance comparison between an xP model only using ED and a model using both ED and TD.

- The work by Anzer.G and Bauer.P mostly tends to research question 1 in this thesis. In contrast this thesis answers research question 1 in more detail by not only consulting general metrics but capturing more detailed performance differences by for example reviewing class specific performances and probability distributions. Further, research question 2 gives even more in depth comparison by explaining the model's general behaviors in probability outputs for specific types difficult of passes as well as what type of passes the models seems to struggle with the most. All in all, the evaluation methodology used in this thesis is different and adds new perspectives.

The concrete inspirations from their work and therefore the specific similarities between the methodology of this thesis the methodology in the "Expected passes" paper are the following:

- This thesis employs a synchronization algorithm for TD and ED mainly inspired by the idea and theory behind the synchronization algorithm outlined in the first section of the methodology by Anzer.G and Bauer.P.

- This thesis similar to Anzer.G and Bauer.P also utilize the ball trajectory model proposed by Spearman et al within a method for extracting intended pass location, although the method itself is as mentioned very different. [35]
- Some less complex TD features used in the feature dataset for this thesis was inspired by features also used by Anzer.G and Bauer.P.

- The threshold dependent and independent performance metrics utilized in this thesis were inspired by metrics also used by Anzer.G and Bauer.P.

2.2.2 Earlier work related to new playability KPI

The new KPI proposed in this thesis called xPlay for *expected playability*, tries to quantify a player's playability. To the knowledge of this thesis, the xPlay KPI is quite novel in it's structure and idea, but the KPI makes use of the theory of hypothetical or artificial passes and their probability of being accurate which is an area previously explored.

Dick et al in their work "Who can receive the pass? A computational model for quantifying availability in soccer" tries to capture something quite similar to the xPlay KPI but in a very different manner. This work looks at how available a receiver potentially is by computing the likelihood that the receiver gets to the ball if its played within a zone around the receiver. This is done in a complex manner using physics-based modeling of potential passes and also incorporates recurrent graph neural networks to capture player and ball interactions.[18]

In the "Expected Passes" by Anzer.G and Bauer.P it is discussed briefly that a potential use for their improved xP model would be to make hypothetical passes to each teammate in a passing situation and based on each pass artificial xP value determine if the passer chooses a safer or more dangerous pass for a passing situation.[9]

The common theme of these related works is as mentioned that they explore the area of hypothetical pass success. The new KPI proposed by this thesis xPlay also explores this subject but in a slightly different manner and application. By using the idea of hypothetical pass probabilities derived using an xP model much like the suggestion and idea proposed by Anzer.G and Bauer.P in [9], xPlay build upon this theory by in contrast contributing the pass probabilities to potential receivers that are only within a certain distance to the pass situations. Further, by using the artificial receiver xP values a transparent and interpretable measure for the game average probability of a player being within close proximation to a passing situation as well as being successful in receiving the ball can be extracted. This measure can be used for interpreting a player's playability. Because of this methodology and usage of artificial pass probabilities there exists to the knowledge of this thesis plenty of novelty surrounding the methodology behind proposed xPlay KPI.

3
Methods

This chapter presents the entire methodology used within the thesis to reach the defined aim and objectives. It is divided into three main sections based on initial data preparation, statistical xP modeling and modeling of the new playability KPI.

3.1 Data Preparation and Integration

This section outlines all the initial work that was done with both the TD and ED which in the end could produce a new enriched dataset for the xP modeling. This process was planned in detail from the start and there were several key stages for the data preparation that were established and these stages will be described in the following subsections.

3.1.1 Initial data exploration and processing

The first stage was an initial data preparation or processing stage where the both the TD and the ED was researched to understand how to structure the TD for a successful integration with the pass events from the ED. The most important detail and difference between the two datasets that requires mentioning is that they were using different coordinate systems. First of the TD was using a more standard type of football coordinate system by using the pitch length as the x-axis extending from 0 to 105 while the width in the y-axis extends from 0 to 68 in an inverted form. The ED was in contrast using a coordinate system where both the width and height were scaled to be between 0 and 100, see fig 3.1.

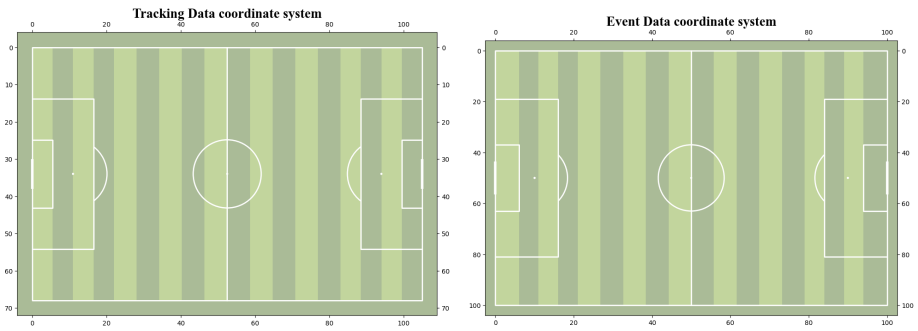

Figure 3.1: Coordinate system for the TD in the left subplot and ED in the right subplot.

Furthermore, the TD was a rich and big data file with a lot of information. The format of the data was in a *jsonl* file namely a very large type of json file. An example of a TD truncated row, namely a row with only some of the players, can be seen in A.1 in the appendix. The ED was formatted in csv files and an example of row of ED information can be seen in A.2.

As the ED was already structured in a tabular format there was no real work required at that end. In contrast, the TD was processed in a manner where the ball and player data was a lot more accessible to ease indexing player and ball information as well as the overall work with the TD at later stages. Further, it was also essential to reduce the size to be able to efficiently work with data. All this was achieved by processing the TD data files into two new separate csv files, one with the ball data and one with player data.

3.1.2 Synchronization of Event and Tracking data

The next stage in the methodology was to synchronize the events in the ED with a correct positional frame from the TD. Furthermore, synchronization of ED and TD at first glance may seem as a trivial task as it should theoretically just be based on matching a frame from the TD with the same timestamp as the event in the ED. But this simply is not the case as often ED and TD comes from different providers that might use different internal clocks for measuring time. Furthermore as presented by Anzer.G and Bauer.P the ED is usually manually annotated data and therefore there is a risk of human error in the annotation of the time of the event. [8]

The strategy and algorithm developed for this project was heavily inspired by especially both of the works by Anzer.G and Bauer.P as well as Van Roy.M et al. [9][8][37] The method builds on the theory of correcting two types of time offsets that can exist between ED and TD as presented by Anzer.G and Bauer.P.[9][8]

The first type of offset will be called ***static time shift*** for this thesis and this offset describes if there is on overarching constant offset in time between the TD and the ED. This static time shift is the first thing that needs to be identified and accounted for and if it exists the whole TD must be adjusted by this shift as to align it with the time in the ED. The second type of offset will be called ***stochastic time shift*** and describes the random offset in time for an event that most likely is a result of human inaccuracy in the ED.

Moreover, the different halves within the game needs to be treated as *separate games* as it is not always the case that the static time shift apparent in the first half is the same in the second half. This is in most cases due to that both the ED and TD simply continue the time tracking from the point where it was stopped in the last moment of the first half. It is more likely than unlikely that their internal clock stopped at slightly different times which induces a new static time shift for the second half.

3.1.2.1 Synchronization Algorithm

The essential steps and the general workflow of the algorithm is outlined in Algorithm 1, but below these steps will be explained in more detail. The concrete strategy to deal with these different time shifts are in principle the same and is based on the strategy from Anzer.G and Bauer.P by using time windows in the TD that extends around the timestamp of an event from the ED.[9][8] The static time shift correction must be done first but is only done once by simply iterating through the passing events of the ED in order and if a matching event can be detected in the TD the possible static time shift is extracted and used to adjust the possible static time shift in the TD.

Unlike the work of Anzer.G and Bauer.P, in this thesis the time window for the static time shift correction is defined by putting a static lower limit on the time window, 0 seconds for first half while the second half uses the last recorded time for the first half in the TD. A varying upper limit of $lower\_time\_limit + 3 * event\_index$ seconds is then used, where the $event\_index$ is the index of the event for each half within the ED. This means that the first event for each half, namely the kick offs, get a 3 second window while the next event would get a 6 seconds window and so on. All tracking data regarding the player involved in the passing event and the ball is the extracted as well within the defined time window. Furthermore, out of the time window frames the selection is further refined by extracting the frames where the ball and the player are within a distance of 2 meters from each other as these are the frames where the player can be considered to be in possession of the ball. [9][8] If no possession frames can be identified the event is dropped and the algorithm moves on. This is iterated until an event for which TD possession frames is identified.

In contrast, to correct the stochastic time shifts a constant time window is instead used. If the passing event has the annotated time t_{ED} a time window of ± 5 seconds around t_{ED} is utilized. The same procedure as with the static shift is then followed by extracting possession frames within that window or alternatively dropping the event and moving on if no such frames are found.
Furthermore, after relevant possession frames for an event has been found the exact same procedure is applied both for the static and stochastic time shift synchronizations. The procedure is based on calculating different types of features for the possession frames similar to the algorithm by Anzer.G and Bauer.P. [9][8] The exact features used in their work is not know but for this thesis for each possession frame, these features entails:

- **Change in ball speed value from current frame to next**
- **Percentage increase in ball speed from current frame to next**
- **Distance between TD ball location and ED pass event location**
- **Distance between TD passing player location and ED pass event location**
- **Change in distance between player and ball from current frame to next**
- **Change in ball traveling direction from current frame to next**

Using the above features each frame got a feature score which then was used to sort and rank each frame within each feature where the most relevant frames were sorted into first position. For example a frame which has a high change in ball speed from itself to the next frame will get ranked high within that feature and therefore be positioned early within the frame sorting for that feature. When all of the frames have got these different feature scores and been ranked, a final aggregation is done by iterating over the feature scores and giving each frame a final value within each feature according to the following formula:

$$F_{ij} = w_j(n_j - rank_{ij}) \tag{3.1}$$

Here F_{ij} is the final feature value for feature j for frame i, w_j is a weight for feature j, n_j is the amount of frames within the feature j and lastly $rank_{ij}$ is the rank or position of frame i for the feature j. The best frame is then selected by then summing the final feature values for each frame and then extracting the frame with the maximum score. Furthermore, as mentioned earlier for the selected frame in the static time shift case that is done first, we also extract the difference in time between the annotated event time t_{ED} and the time for the selected frame from the TD. This difference is then used to adjust all the time values in the entire TD for that half before moving on with the stochastic time shift synchronization. One important aspect to mention is that the idea of using weights to aggregate the feature scores is inspired by Anzer.G and Bauer.P, although in contrast for this thesis the scoring weights w_j were tuned manually instead of using optimization.[9][8]

Algorithm 1 Complete synchronization algorithm of ED and TD for one half period

Require:
Passing event dataset for a half E
Ball tracking dataset for a half T_b
Player tracking dataset for a half T_p
Feature scoring weights W

Perform initial static synchronization:
Identify lower limit of time window

for each event $e \in E$ **do**
    Get upper limit of time window
    Extract possession frames within time window using T_b and T_p

    if possession frames exist **then**
        Get feature scores for all possession frames
        Aggregate feature scores using weights W
        Extract best-ranking frame and static time shift t_{diff}
        Remove event e from E
        break

    else
        Remove event e from E
    end if
end for

Offset all time values in T_b and T_p by t_{diff}

Perform refined stochastic synchronization:
for each remaining event $e \in E$ **do**
    Let t_{ED} be the time of event e in ED
    Define time window $[t_{\text{ED}} - 5\text{s},\ t_{\text{ED}} + 5\text{s}]$
    Extract possession frames within this window using T_b and T_p

    if possession frames exist **then**
        Get feature scores for all possession frames
        Aggregate feature scores using weights W
        Extract best-ranking frame and stochastic time shift t_{diff}
    end if
end for

3.1.2.2 Processing and Refining Synchronization Results

Synchronizing ED and TD perfectly by always finding the correct TD frame is not usual as seen in the results from earlier work.[8][37] For the algorithm developed for this thesis it does produce some events which get *wrongly* synchronized. Seem-

ingly the errors as well as the frames that does not get synced at all appears mostly randomly although there were some situations which seemed to increase the risk of synchronization errors slightly. These types of situations were usually if a passing player made many similar passes in a short time span from a very similar positions, if the passing player made some type of extensive pushing or dribbling action on the ball before passing or the TD simply had some type of error in recording either the ball or player position accurately.

To deal with these potential errors the results after the synchronization were refined through post processing. By looking at the distribution of the stochastic time shifts for each half one could deduce that they always are quite normally distributed although a bit skewed, with fat tails and large kurtosis, see figure 3.2. Consulting the distributions in combination with manual evaluation revealed that wrongly synced events usually had stochastic time shifts distinctly different from the average stochastic shift for the events of the synced half period. Using this fact the IQR method was used as a robust technique to identify and remove outliers which most likely were wrongly synced events. The IQR-method uses the quantiles namely cut off points where a certain proportion of the data falls below each cut off. Using Q1 representing the cut off for the first 25% of the data and Q3 representing the cut off for the first 75% of the data the interquartile range or IQR according to $Q3 - Q1$ is extracted. The IQR is then used to identify and remove outliers extending outside Q1 - 1.5*IQR and Q3 + 1.5*IQR within the distributions. This ensured better precision in terms of correctly chosen frames for the synchronized events.

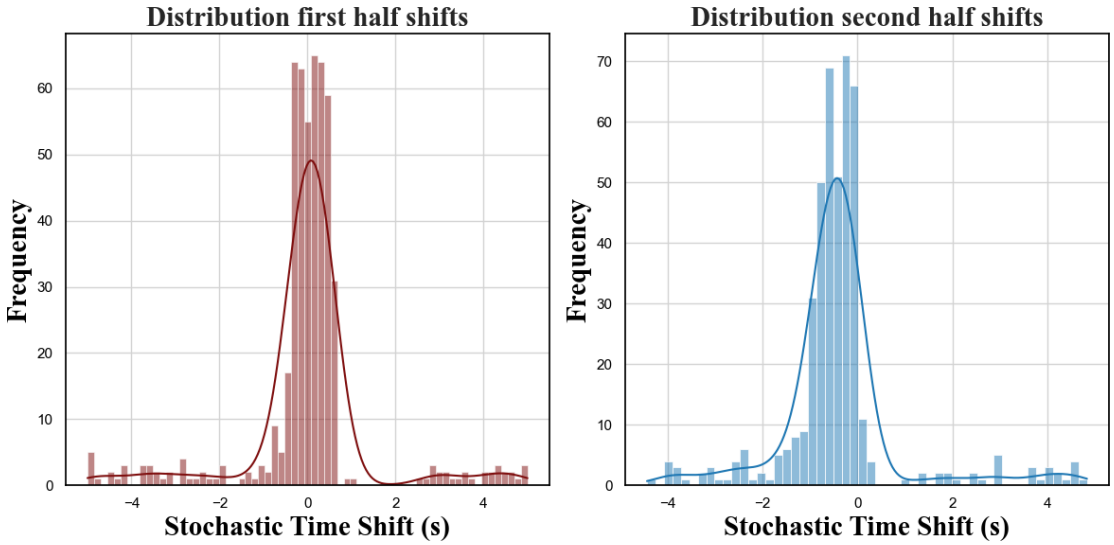

Figure 3.2: Example of the distribution of stochastic time shifts for passing events from a game after synchronization.

3.1.3 Tracking Data feature engineering

The stage that follows the synchronization stage was the TD feature engineering stage. The TD feature engineering put a lot of emphasis on looking at aspects that

regular ED completely misses. These aspects relates to the spatial context around passing situations in regards to distances, pressures and angles to opposition players. To be able to extract some of these features it was also important to understand the specifics of the passing situation in regards to the ***intended*** direction of the pass as well as the actual passing lane itself. In the next subsections the method for first extracting the pass direction and lane will be outlined and after the concrete features developed are introduced.

3.1.3.1 Extracting Passing Lane and Direction

Extracting the passing lane and direction is in most cases a trivial task for accurate passes but for inaccurate passes this quickly becomes non-trivial as a lot of the time the actual passing lane and direction were not usually the intended one. Furthermore, some inaccurate passes actually do not even have an intended target like clearances or immediately intercepted passes. This thesis utilized a simple method for getting acceptable intended pass directions and passing lanes for pass events by sending each pass event through different checks.

1. ***Check 1 - Is the pass successful and have a tagged receiver?:*** This check looks at accurate passes with a tagged receiver. If a pass event qualifies for this check the method then uses the same possession window strategy used for the ED and TD synchronization algorithm but looks at a possession window for the tagged receiver with a lower time limit of the time of the pass time stamp and an upper time limit of 5 seconds. If no relevant possession frames are identified the event is dropped as this indicates a likely wrongly synced pass event. In contrast if there exists possession frames, the receiver position of the first frame within the possession window is set to the intended pass location.

2. ***Check 2 - Is the pass inaccurate and have a tagged receiver?:*** This check looks at non-accurate passes with a tagged opposition receiver. The procedure is the same as for check 1 with the difference being that after extracting the first possession frame of the receiving opposition player, it uses the time stamp of that frame and extracts all the positions of the players on the same team as the passer. Next, by checking if there were teammates within 8 meters of the receiving opposition the closest teammate's position is set as the intended pass location. Although, if no teammates are found within 8 meters the pass is treated as having no intended teammate and therefore the receiving opposition position is set as the intended pass location.

3. ***Check 3 - Dealing with remaining cases:*** This last check takes all the passing events that does not pass any of the previous checks and simulates the ball trajectory based on the physical trajectory model proposed by Spearman et al.[35] The model proposed uses the following formula for the ball

acceleration:

$$\ddot{\vec{r}} = -g\hat{z} - \frac{1}{2m}\rho C_D A \,\dot{r}\,\dot{\vec{r}} \tag{3.2}$$

Here g = 9.8 is the gravitational acceleration ($\frac{m}{s^2}$), $\hat{z}$ a unit vector pointing straight down, m = 0.42 is mass of the ball (kg), $\rho = 1.22$ is the assumed air density ($\frac{kg}{m^3}$), $C_D = 0.25$ is drag coefficient and A = 0.038 is cross-sectional area of the ball (m^2). Using an initial ball velocity and position from the synced TD frame and utilizing the scipy function "*solve_ivp*" in combination with the Runge-Kutta method, this differential equation is solved for a fixed time span of around 2 seconds to get a possible pass trajectory and the last ball position is set to the intended pass location.[5]

3.1.3.2 Direct Press Feature

Pressure in football is defined by through specific movement and positioning influencing the opposition team's capability of making decisions in a negative way. Quantifying pressure is non-trivial as there exist several factors which effect how *pressured* a player becomes. One type of pressure that this thesis defines as ***direct pressure*** concerns opposition that is actively moving straight towards a passing player to close the distance and therefore create pressure. To accomplish quantifying this type of pressure the work "Pressing Intensity: An Intuitive Measure for Pressing in Soccer" by Bekkers.J was used. The method presented uses approximations of the *time to intercept* an opponent to calculate the probability of interception. [10] The exact implementation of the method within the context of this thesis is outlined below.

Time to intercept (TTI): The first main step was to calculate the TTI the passer for opposition players which is done using equations from Bekkers.J adjusted for the context of a single passing player [10]:

$$T_i = \tau_r + \tau_i + \tau_i^{\beta}, \tag{3.3}$$

$$\tau_i = \frac{\left\| \vec{r}_p + \vec{v}_p - \left(\vec{r}_i + \tau_r \, \vec{v}_i \right) \right\|}{v_{\max}}, \tag{3.4}$$

$$\tau_i^{\beta} = \frac{\|\vec{u}_i\| \; \beta_i}{\pi}, \tag{3.5}$$

where

$$\vec{u}_i = \vec{r}_i + \vec{v}_i - \vec{r}_i, \tag{3.6}$$

$$\vec{m}_i = \vec{r}_p + \vec{v}_p - \vec{r}_i, \tag{3.7}$$

$$\beta_i = \arccos\left(\frac{\vec{u}_i \cdot \vec{m}_i}{\|\vec{u}_i\| \; \|\vec{m}_i\|} \right). \tag{3.8}$$

Equation 3.3 describes the TTI the passing player for a direct pressing opposition player i T_i. This TTI is dependent on the summation of three terms where τ_r is the

opposition reaction time, τ_i is the time for the opposition to reach the passer and lastly τ_i^{β} is a term that gives a time add on if the opposition presser has to make a significant angular change in movement. Other important variables are $\vec{r_p}, \vec{v_p}$ and $\vec{r_i}, \vec{v_i}$ which are starting position and velocity vector of the passer and the presser i respectively. Further, $\vec{u_i}$ is a vector that describes where the opposition is moving while $\vec{m}_i$ is the direction from the presser to where the passer is moving. These are used to calculate β_i which looks at how similar the direction of the presser movement is to the direction of where the passer is moving which captures if the opposition is moving in a correct direction to intercept the passer. Lastly, these equations are only applied to opposition pressers moving above 2 m/s which captures the direct pressure aspect namely active movement to exercise pressure.[10] Within the equations there exist two constants in τ_r and v_{max} that describe human reaction time as well as maximum velocity of an average football player. The reaction time was set to 0.5 while the maximum velocity was set to 8 m/s based on the work in the following papers [35][17].

Turning TTI values to a probability: The TTI values was used to get a single interception probability according to the following equations[10]:

$$p_i\Big(T_i,\, T^{intercept} \mid \sigma\Big) = \left[1 + \exp\left(-\frac{\pi}{\sqrt{3}\sigma}\left(T^{intercept} - T_i\right)\right)\right]^{-1}, \tag{3.9}$$

$$P = 1 - \prod_i \Big(1 - p_i\Big) \tag{3.10}$$

Equation 3.9 gives an estimated probability of presser i intercepting the passer by plugging the presser TTI value T_i into a logistic function. Further, the specific logistic function utilizes two hyperparameters where $T^{intercept}$ regulates the TTI threshold for when there is a 50% probability of interception. This hyperparameter was set very low to 0.08 seconds as the pressure calculation is done for the exact moment of the pass. Next, the σ regulates the steepness of the logistic curve which in simple terms defines how steep the probability transition is from low to high probability. As mentioned the pressure is calculated at the moment of the pass which entail that if a presser is not able to intercept within the threshold then there is a very low probability of interception which prompts a larger σ value. Specifically for this thesis the value of 0.75 was used which is larger in comparison to the value used by Bekkers.J in 0.45 although in their work the method implementation is not specified to pressure at the exact moment of a pass. In the end equation 3.10 models the probability of at least one opposition presser manages to intercept the passer naively assuming independence which provide a final approximate pressure value P between 0 and 1.[10]

Incorporating sideline and goalline pressure: This thesis makes a new contribution to the direct pressure model by considering the increase in pressure when the passer is close to either the sidelines and/or the goallines on the pitch. This consideration was a suggestion from Bekkers.J in regards to possible improvements and future work of the model.[10] The thesis incorporates this aspect according to the following equations:

$$M_{SL}, M_{GL} = \begin{cases} 1 + \alpha \left(1 - \frac{d}{10}\right), & \text{if } d < 10 \\ 1, & \text{if } d \geq 10 \end{cases} \tag{3.11}$$

$$P_{final} = P \cdot M_{SL} \cdot M_{GL} \tag{3.12}$$

These equations describe variables M_{SL}, M_{GL} that are multipliers for potential sideline and goalline pressure giving a final pressure value P_{final} and they are dependent on the minimum distance d to the sideline for M_{SL} and to the goalline for M_{GL}. We only consider distances closer than a 10 meters threshold as it would be unreasonable to have a constant pressure from the different lines. Furthermore, α is a constant that determines the maximum added pressure in percentage, this constant was put to 0.15 by a process of arbitrarily manually trying different values and looking at which ones yielded reasonable results. In figure 3.3, the result of the method can be viewed where the equations are used to calculate a direct pressure distribution grid around opposition pressers.

Important to mention regarding the direct pressure model and methodology is that equation 3.10 very naively assumes independence of the different interception probabilities which is not the case. Pressing in football is not done individually and how a player decides to press is therefore very much dependent on how his teammates are pressing. Furthermore, multiplying the total probability with arbitrary multipliers also diminishes the aspect of the pressure metric being represented as a probability. Therefore, P_{final} was viewed as well as used as a general *direct pressure value* for the thesis.

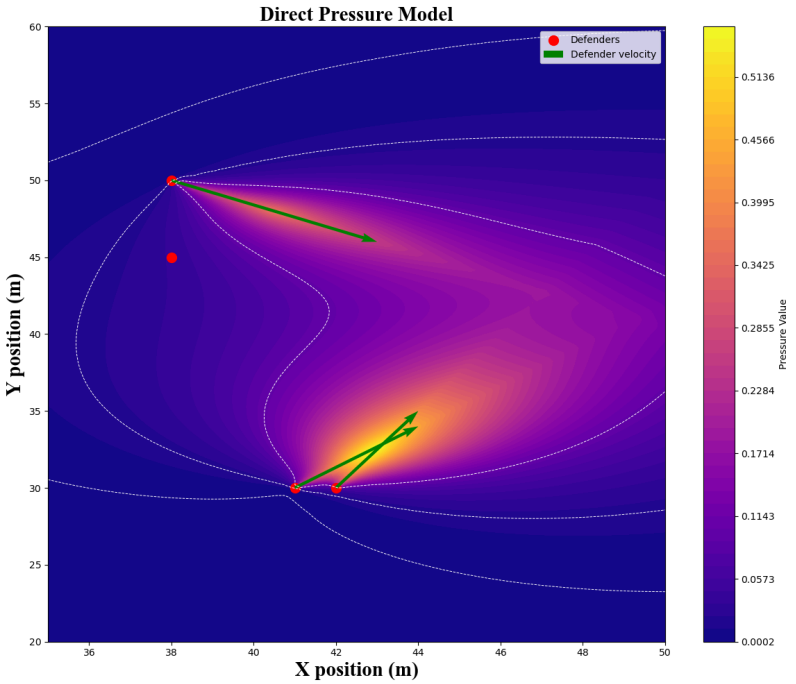

Figure 3.3: Direct pressure distribution using equations 3.3-3.12.

3.1.3.3 Indirect Angular Press Feature

Another important aspect of pressure within football is *where* the pressure is coming from in relation to the pressured player. The thesis views this pressure aspect by utilizing the model proposed by Andrienko.G et al in their work "Visual Analysis of Pressure in Football". The model describes the pressure asserted onto a player as a numerical value based on the relative positions and angles between the opposition pressing players and the player with the ball.[7] The model is based on three main aspects:

Threat direction: Andrienko.G et al defines something called *threat direction* as the direction in which the player with the ball could cause the most threat. Most implementations of this model uses a threat direction pointing towards the opposition goal. This thesis implements this slightly different as the threat direction is viewed as the direction in which pressure would pose the most threat for the player with the ball. Further, for the work of this thesis that regards passes, the threat direction naturally becomes the direction in which the passing player intends to pass the ball.

Pressure zone: A pressure zone is defined by a boundary for where pressure can be exerted and this is calculated using the following equations [7]:

$$L = D_{\text{back}} + \left(D_{\text{front}} - D_{\text{back}}\right) \frac{z^3 + 0.3\,z}{1.3}, \tag{3.13}$$

$$z = \frac{1 + \cos\Theta}{2}. \tag{3.14}$$

Here, L is the pressure boundary dependent on the constants D_{back} and D_{front} as well as the variable z which is dependent on the cosine similarity of the angle θ between the threat direction and the direction from the passer to the presser. The constants are put to $D_{back} = 3$ and $D_{front} = 9$ as Andrienko.G et al describes these values as reasonable and in line with recommendations from experts.[7] Equation 3.13 creates a pressure boundary in an oval shape where the long axis points exists along the threat direction.

Pressure calculation: After the zone for pressure is established each potential pressuring opposition player can be classified into being within the pressure zone or outside. For each opposition player within the pressure zone a pressure value is calculated according to[7]:

$$P_r = \left(1 - \tfrac{d}{L}\right)^q \times 100\%, \tag{3.15}$$

Here a pressure value P_r between 0 - no pressure and 100 - maximum pressure is calculated for each opposition presser based on their distance to the passer d, their pressure boundary from equation 3.13 L and a constant q set to 1.75 in accordance with Andrienko.G et al that determines how fast the pressure should decrease with the distance away from the passer. The total pressure on a passer is given by summing all individual opposition pressure together.[7] An example of a pressure

zone with different pressure values calculated using the above equations can be seen in figure 3.4.

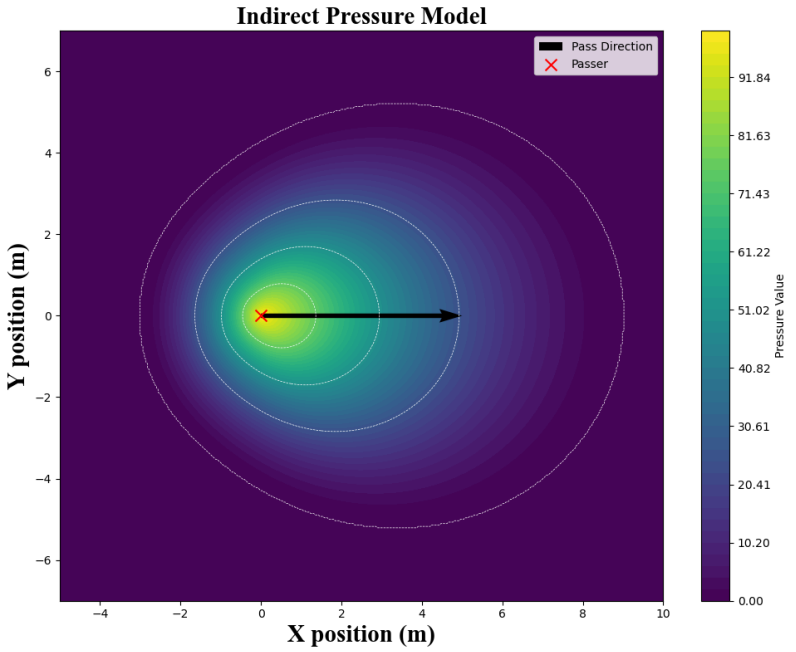

Figure 3.4: Pressure zone and values for an example passer calculated using equations 3.13-3.15.

3.1.3.4 Passing Lane Press Feature

The final pressure feature is looking at an aspect not covered as extensively when it comes to pressure within football and this is ***pressure put on the passing lane***. This concept looks at how opposition intercepts or block passing lanes, while this type of pressure is a bit more abstract and is not exerted directly onto the passer it is an important aspect as it can considerably effect the outcome of a pass. Bekkers.J outlined in their work "Pressing Intensity: An Intuitive Measure for Pressing in Soccer" how the TTI pressure model, used earlier for the direct pressure feature, can be adapted to calculate pressure on passing lanes by looking at the TTI of an opposition to their closest point on the passing lane.[10] This thesis explores that established idea and implementation to develop a passing lane pressure feature with some innovative tweaks according to the following steps:

Calculate closest points on the passing lane: For each pass there exists an intended passing lane according to section 3.1.3.1, using that passing lane a calculation of each opposition player's closest point, cp_i for opposition i, on the passing lane is done in accordance with the suggestion from Bekkers.J. [10]

Dynamic pressure zone along passing lane: The next step takes inspiration from the pressure zone concept defined for the indirect feature. A pressure zone in the shape of a 45 degree *circle sector-like* shape around the passing lane is defined. The inspiration to use a specific shape like this was from earlier work by Sharpe.C

for the company Hudl Statsbomb where they used a similar shape when trying to quantify presence along a passing lane.[34] This zone is extracted by the following equation:

$$d_{thres}(d) = d_{\min} + \left(d_{\max} - d_{\min}\right)\frac{d}{d^p_{max}}\,, \tag{3.16}$$

Here, d_{thres} is the calculated dynamic pressure zone distance threshold to the passing lane for an opposition presser that depends on the opposition distance d to the passer. Here to simulate the 45 degree sector this thesis utilized $d_{min} = 0$, $d^p_{max} = 10$ and $d_{max} = tan(\frac{\pi}{8}) * d^p_{max}$.

TTI calculation: For each opposition presser i their distance to the passing player is used in equation 3.16 to get a threshold distance to the passing lane. Comparing distance to cp_i with d_{thres} it can be determined if an opposition presser is within the pressure zone. Next, if that is the case then equations 3.3-3.8 from Bekkers.J are used to calculate the TTI for cp_i.[10] Furthermore, it can be assumed that the closest point on the passing lane will always be static and therefore $\vec{v}_p$ can be put to 0. Important to mention is that the model for this thesis does not filter out opposition players not having a speed above 2 m/s as just standing on the passing lane will still put pressure on it.

Dynamic interception threshold: Next step is to calculate a dynamic and changing interception threshold $T_i^{intercept}$ passed to the logistic function in equation 3.9. This addition makes the passing lane pressure model more realistic as depending on the distance to the passing player, an opposition player will have more or less time available to intercept the passing lane. Therefore, the threshold for when there is essentially a 50% chance of intercepting the passing lane needs to be dynamic. This is accomplished by a similar equation to equation 3.16 according to:

$$T^{intercept}(d) = T^{intercept}_{\min} + \left(T^{intercept}_{\max} - T^{intercept}_{\min}\right)\frac{d}{d^p_{\max}}\,, \tag{3.17}$$

Here, $T^{intercept}(d)$ is the dynamic interception threshold depending on the distance to the passer d. The constants are put to $T^{intercept}_{\min} = 0$, $T^{intercept}_{\max} = \frac{d^p_{\max}}{speed_{pass}}$ and $d^p_{\max} = 10$. The $T^{intercept}_{\max}$ is therefore in itself dynamic and depends on the specific pass in question.

Pressure value calculation: This last stage follows equations 3.9-3.10 from Bekkers.J for each opposition presser within the pressure zone using their individual interception thresholds and TTI values. [10]

Important note is that the passing lane feature has similar behavior as the direct pressure feature as they share the same main principles. This means that the passing lane feature makes use of probabilities using the naive assumption that all interception probabilities are independent. Therefore, similarly to the direct pressure feature, the passing lane pressure cannot be treated as an actual probability and is

rather used as a general press value. In figure 3.5 a grid of possible pressure values are visualized for an opposition with speed $0\frac{m}{s}$ along a passing lane as well as a more detailed example with three possible opposition pressers.

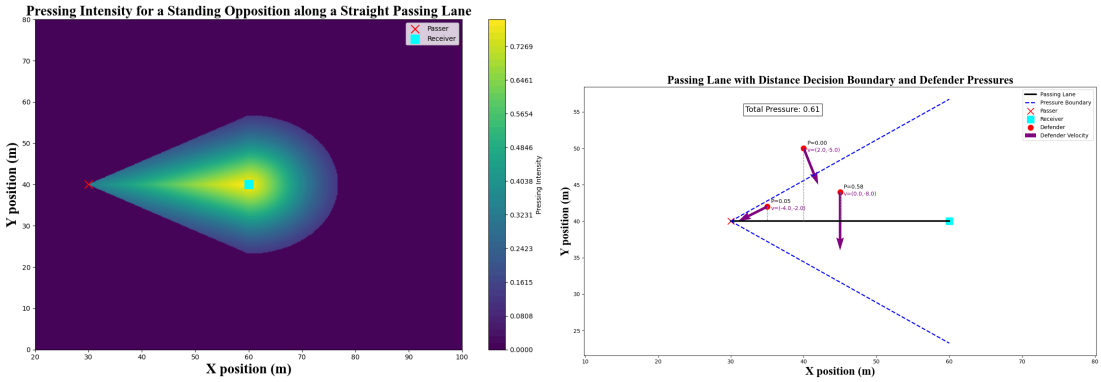

Figure 3.5: Left subplot shows pressure on passing lane grid where the opposition would be standing still using an example passing lane of 30 meters, ball speed 20 m/s. Right subplot shows pressure on a passing lane using example opposition players with their own individual direction and speed on the same passing lane of 30 meters with ball speed 20 m/s.

3.1.3.5 Defensive Line Features

Defensive lines concerns the three lines of defense which a defending team usually employs when defending. The concept of these lines are relevant in terms of passing and as features to an xP model as a majority of football teams focus on preventing their lines from being played through by passes. The method for extracting the defensive lines was heavily drawn from Holmström.M's thesis "Playing Through the Lines: statistical Learning for Analysing Build Up Play in Football". Holmström.M uses K-Means clustering with three designated clusters for the opposition players, with the exception of the goalkeeper, based on their location on the x-axis and uses the cluster centroids to estimate the defensive line positions.[20] This method proved useful in identifying reasonable defensive lines and was therefore also employed within this thesis using the "KMeans" function from Scikit-learn library.[2] Example of clustering and extracted defensive lines can be seen in the example pass situation in figure 3.6.

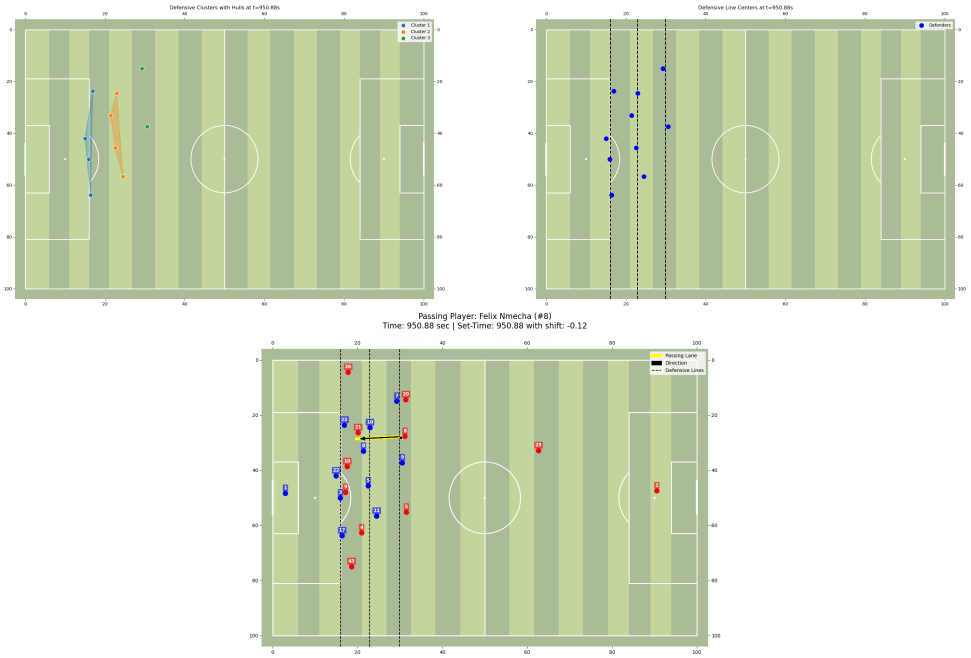

Figure 3.6: For a specific pass event the upper left subplot shows the player clusters, the upper right subplot shows the defensive lines using the cluster centroids and the lower middle shows the lines incorporated into the complete event frame which shows an attempt at intersecting the first two lines.

After the defensive lines had been identified relevant features can be extracted using them. First of, this thesis utilized ***three binary features describing attempt of breaking lines***. These three features, one for each line, describe if the intended pass *attempts*, in the moment of the pass, to break the lines. Secondly, the thisis utilized ***three numerical features describing passer distances to the lines***. These three numerical features describes the distances from the passer to each defensive line calculated by subtracting the passers x coordinate from the specific line x coordinate.

3.1.3.6 Other Less Complex Features

There were also some less complex although new features that could be extracted from the TD and added to the ED, these features are outlined below:

- ***Bypassed opponents:*** This feature looks at how many opponents that are between the passer and the receiver, in terms of x dimension, at the moment of the pass.

- ***Smallest angle between opponent and passing lane:*** This feature looks at all the opponents within the pass distance from the passer and then gets the smallest angle between one of these opponents and the passing lane.

- ***Smallest distance between an opponent an the passing lane:*** This feature is inspired by the feature dataset used by Anzer.G and Bauer.P, furthermore it simply is a value describing the smallest distance between an

opponent and the passing lane.[9]

- ***Smallest distance between an opponent and the passer:*** This feature is a value simply describing the smallest distance from the passer to an opponent at the moment of the pass.

- ***Passing player's speed:*** This feature, also inspired by the feature set used by Anzer.G and Bauer.P, is the passing player's speed at the moment of the pass.[9]

- ***Height of pass:*** This feature describes the height of the pass which is calculated using the average coordinate in the z dimension from the TD for a couple of frames after the moment of the pass.

- ***Speed of pass:*** This feature describes the speed of the pass which is calculated using the average recorded speed values from the TD for a couple of frames after the moment of the pass.

- ***Ball height at pass:*** This feature, also inspired by the feature set used by Anzer.G and Bauer.P, is a value for the z coordinate of the ball at the moment of the pass.[9]

- ***Smallest sideline distance:*** This feature gives the value for the smallest distance from the passer to either of the sidelines at the moment of the pass.

- ***Smallest goalline distance:*** This feature gives the value for the smallest distance from the passer to either of the sidelines at the moment of the pass.

- ***Number of opponents closer to passer than to their own goal:*** This feature gives a value for how many opposition players that have a smaller distance to the passing player than their own goal at the moment of the pass. It is inspired by an inverted version used by Anzer.G and Bauer.P.[9]

3.2 Statistical modeling of xP

This stage describes the modeling of the xP KPI and follows a rigid structure based on a classic data science workflow. The data science workflow consists of 5 main steps going from initial set up and problem definition to the final testing and evaluation of the statistical modeling. In the following subsections the rigid data science workflow will be explained step by step.

3.2.1 Set up and Problem definition

The first action within the workflow was to set up what the modeling is supposed to achieve and try to formulate what the mission is and most importantly define how success is measured. The xP KPI is developed using a statistical learning

model trained within a supervised binary classification setting, namely the model using a feature data set with labeled target data, 1 - accurate pass or 0 - inaccurate pass, is trained to predict these labels. In contrast, the xP model is not used as a binary classifier directly in an applied setting as they are utilized through their outputted probabilities. What this means for the modeling is that what actually matters and measures the success of an xP model is the quality and realism of the pass probabilities. While classical threshold dependent and independent metrics are not completely irrelevant and provides some information, they do not describe in depth the aspects of an xP model that matters the most.

3.2.2 EDA - Exploratory Data Analysis

For the EDA the dataset was analyzed and prepared to ensure effective use of statistical learning techniques. By looking at distributions, cleaning and removing possible outliers and noise, visualizing correlations both between the features themselves and between the features and the target, knowledge about the dataset structure was extracted which prompted certain modeling choices. The concrete steps to achieve this will be outlined below.

3.2.2.1 First inspection and cleaning

When the final pass dataset, see table A.1 for feature structure, was imported it was inspected for possible missing missing values as well as possible outliers and strange behaviors for certain features using feature distributions and boxplots. The final feature distributions after cleaning, see figure A.2 and figure A.1, importantly revealed the imbalance in regards to the dataset target variable with around 84% of the data being accurate passes. In general there was also imbalance for several of the binary features such as throw-in, freekick and corner passes. Furthermore, after cleaning, the final size of the dataset was 19,521 passes with 34 features where 20 of these features are added information from the TD. This data was compiled from a total of 27 games which means that one games was not included. It was decided that for evaluation purposes set aside one of the 28 games as it could be used to get test results that could easily be manually judged and evaluated see coming section 3.2.5.5.

3.2.2.2 Dataset correlations

After the initial cleaning and inspection the correlations between the features themselves as well as between the features and target was evaluated. Figure 3.7 shows the Pearson and Spearman correlation matrices for the features excluding the target. It is possible to see some correlations especially for the distance metrics where the defensive line distance features shows the most correlation with each other which is expected. All in all, there did not exist heavy multicollinearity that prompted feature selection or further feature engineering. Important caveat to recognize is that there exist several binary features which violates the assumptions of both Pearson and Spearman correlation. Further several numerical features are not normally

distributed which further violates the assumptions of Pearson correlation. These violations does limit the range and interpretability of the correlation values but in practice both methods can still provide very useful estimations of correlation as well as reveal some patterns that could show multicollinearity and they can therefore still serve as valid EDA methods.

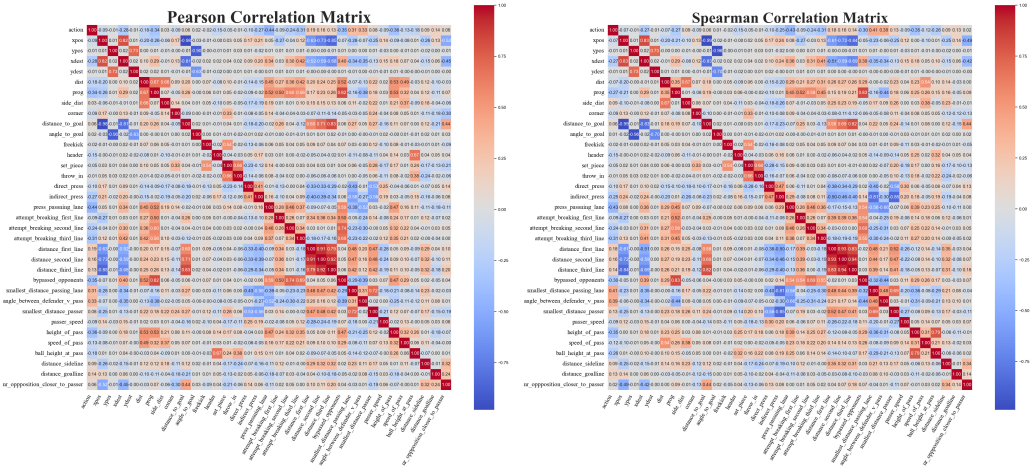

Figure 3.7: Left subplot shows Pearson correlation for final dataset while right subplot shows Spearman correlation.

Moreover, in terms of target correlation, figure A.3 and figure A.4 shows the change in target probability, namely making an accurate pass, based on different feature values. Several features seem to correlate and provide information about the target with features as *dist, prog, indirect_press, press_passing_lane and bypassed_opponents* showing very clear trends. Regarding the binary features it's harder to make strong inference as their distributions were extremely unbalanced. They seem to show a slightly lower success probability for corners and headers which is reasonable from a football perspective. Furthermore, the success probability seemingly decreases with the defensive line that is broken, namely breaking the third defensive line seems to entail generally lower success probability in comparison with breaking the first defensive line which once again would be reasonable within a football perspective.

The final check of the features looked at the mutual information score between the features and the target as well as the Spearman correlation coefficients, see figure 3.8. The mutual information scores ranks *smallest_distance_passing_lane, press_passing_lane and angle_between_defender_v_pass* as top descriptive features while *freekick, set_piece, throw_in and corner* does not provide that much information at all. Moreover, the Spearman correlations seems to confirm and strengthen the mutual information scores. Important to note is that because of the target imbalance the Spearman correlations becomes less accurate and trustworthy, therefore the results should only be interpreted in a wider sense as to which features at large gives higher or lower correlations.

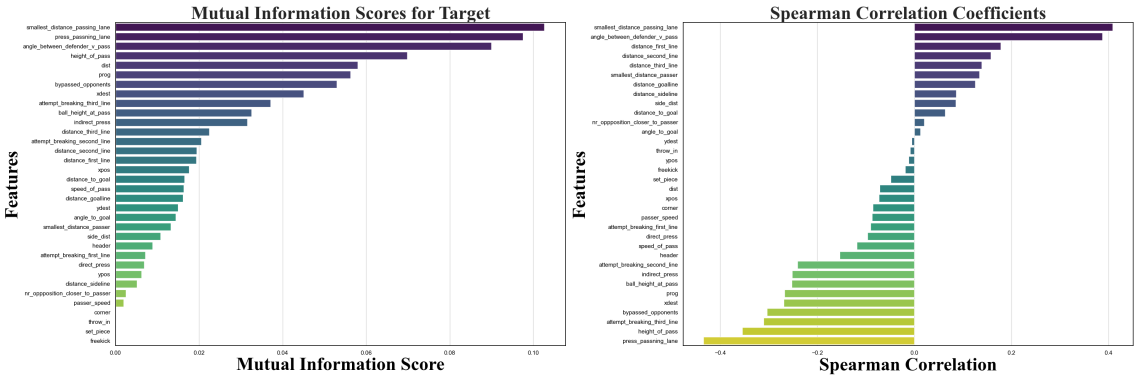

Figure 3.8: Mutual information scores between target and features in the left subplot and the right shows spearman correlation coefficients between target and features.

3.2.2.3 Dataset structure in higher dimension

The final step in terms of first inspection and preparation was to visualize the data within the feature dimension. For this UMAP was used. As the UMAP method is dependent on several hyperparameters, most importantly the "n_neighbor" parameter that controls wether UMAP focus on capturing more local or global structure, two visualizations were generated first for 20 neighbours - looking more locally and one for 100 neighbours - looking more globally. This was also done first only using ED features and then using all features. As seen in figure 3.9 and 3.10, there is not much difference between the local and global structure and there were tighter distances between data points and a bit more separation for the ED-only data. Furthermore, the added TD information added dimensionality and context, it's therefore no surprise that the UMAP embedding was more spread out. Importantly, it would seem that the inaccurate passes is better sectioned for the data using the added TD features as the majority of these passes does not mix into the big cluster of accurate passes. For the ED only data, although areas with higher inaccurate passes exist, these areas seem to be more intertwined with the accurate pass data in comparison to the data using all features.

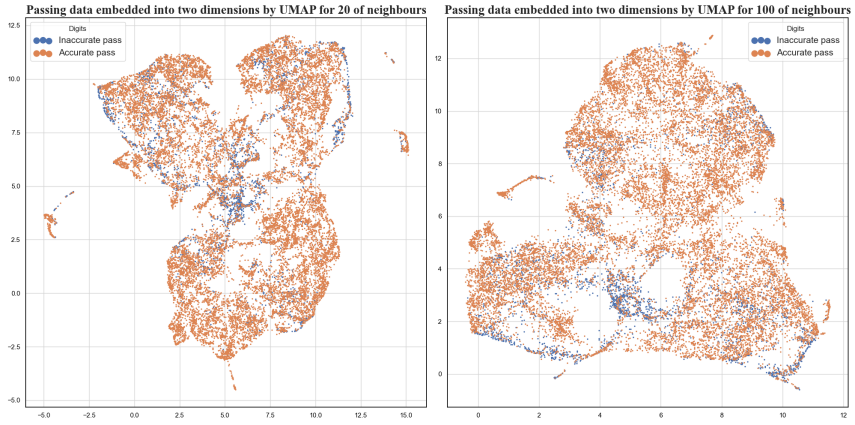

Figure 3.9: UMAP visualization of approximate data structure in higher feature dimension for dataset with only ED features.

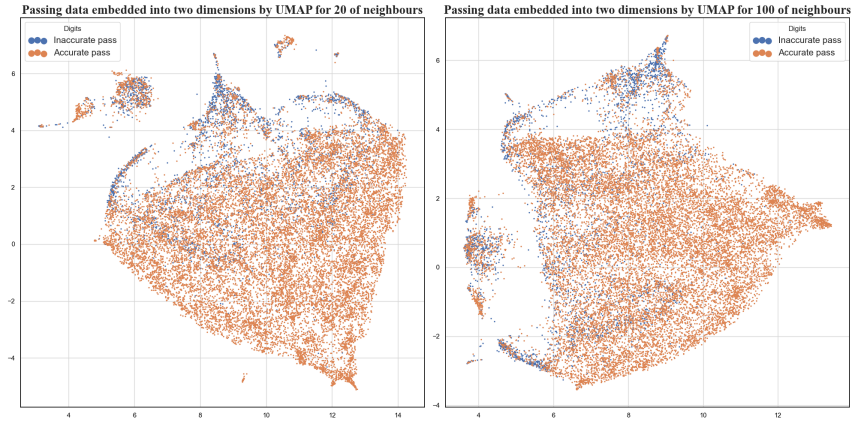

Figure 3.10: UMAP visualization of approximate data structure in higher feature dimension for dataset with ED and TD features.

3.2.3 Feature Selection

Using the results from the initial inspection and cleaning it was decided that no initial feature selection should be executed because of several reasons. First, the datasets that will be compared, namely ED features only as well as all features, have the dimensionality of 14 and 34 respectively. This is not considered very high dimensional data that immediately would prompt reduction. Adding to this, the dimension of the datasets, almost 20000 samples to 14 and 34 features, means that there exists plenty of data in relation to the number of features which also prompts that feature selection is less necessary in terms of the *curse of dimensionality* where distance and relations break down in higher dimensions. Moreover, removing features always entails the risk of removing information, even if the information is not the best it is still information. Furthermore, weak signals or features can still provide information in combination with others. As established in the set up and problem definition, the xP model will have to evaluated thoroughly in regards to it's realism. Within this aspect the interpretability is a big factor and to analyze the

interpretability it is positive to have as much context and information surrounding the passing events as possible. Lastly, many statistical learning models, for example tree based models, have built in procedures that makes them robust against features that possibly do not add information. Therefore using a robust statistical learning model would deal with the aspect of feature selection itself.

3.2.4 Selecting a Statistical Learning Model

To select a statistical learning model for the final xP modeling a comparative analysis was done. The initial models selected for comparison was three decision tree and ensemble based models in ***Random Forest, XGBoost and CATBoost***. The decision tree ensemble based models were the main focus and initial favorites as they work very well with tabular data with mixed types, are less sensitive to feature scales, robust against noise and outliers and have built in feature importance and selection which provides interpretability. This is highly attractive attributes as the data have potential smaller noise level and outliers because of synchronization errors between the ED and TD, see section 3.1.2.2. Furthermore, the dataset is tabular using a varied type of features and the interpretability adds support in answering research question 2 of this thesis. Moreover, to add to the comparative analysis it was relevant to provide a baseline and therefore ***Logistic Regression*** was also included in the model selection. Lastly, purely for comparative purposes another model using a different method to the other models was included in a ***Support Vector statistical - SVM*** model.

The set up for performing the model selection used a workflow that each statistical learning model was put trough, visualized in figure 3.11.

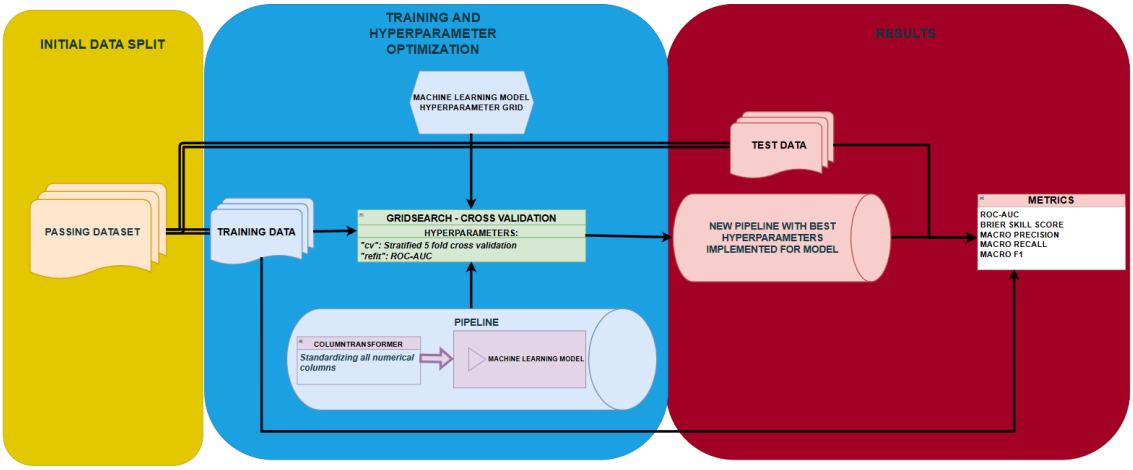

Figure 3.11: Visualization of workflow for training and testing statistical learning models.

Initial data split: The initial data split sections a shuffled and stratified 80% chunk of the passing dataset for training and sets aside the remaining 20% chunk for testing.

Training and hyperparameter optimization: This step used the Scikit-learn function "Pipeline" which allows to create a pipeline with several transformer instructions for pre-processing as well as a final predictor model.[4] This pipeline used two steps for first standardizing the numerical and ordinal features and then sending the data to the statistical learning model. The pipeline was used in combination with the training data and a specified hyperparameter grid within a stratified 5-fold gridsearch cross-validation to execute hyperparameter optimization. The gridsearch extracts all possible hyperparameter combinations and then splits the training data into 5 stratified and shuffled sections. Lastly, it trains the model on 4 of the 5 sections and tests the trained model on the last section and this is done for all 5 combinations of these 5 sections. Using a "refit" metric the hyperparameter combination that got the best average validation score across the 5 combinations for this metric is picked as the best hyperparameters. For the gridsearch used in this thesis ROC-AUC was used as a refit metric as it is not threshold dependent and not heavily effected by target imbalance making it a suitable choice for an xP model.

Results: From the gridsearch cross validation a new pipeline is received that has a fully trained statistical learning model inside with the best performing hyperparameters. This model is then finally tested against the hold out test set in regards to the following metrics; ***ROC-AUC, Macro-F1 Score, Macro-Recall, Macro-Precision and Brier Skill Score***. The choice of metrics reflect the important aspects when assessing classification performance on an unbalanced data set and were also inspired by the metrics used by Bauer.P and Anzer.G for evaluating xP models.[9]

This whole process was executed first using only ED data and then ED+TD data for each of the five picked statistical learning models and then cross-validated by repeating the whole workflow for 3 different shuffled and stratified train and test splits. The model selection process gave the following results shown in table 3.1 and table 3.2. The tables show that the CATBoost model has the best performance across all metrics both for the ED only data and the complete new dataset.

Table 3.1: Test set performance (ED only) (mean ± std)

Classifier	BSS	Macro-F1	ROC-AUC	Macro-Precision	Macro-Recall
Logistic Regression	0.211 ± 0.011	0.643 ± 0.008	0.809 ± 0.005	0.746 ± 0.009	0.616 ± 0.006
SVM	0.210 ± 0.014	0.648 ± 0.007	0.832 ± 0.004	0.784 ± 0.008	0.618 ± 0.005
Random Forest	0.359 ± 0.014	0.746 ± 0.013	0.877 ± 0.004	0.807 ± 0.010	0.714 ± 0.016
XGBoost	0.349 ± 0.020	0.749 ± 0.009	0.875 ± 0.005	0.794 ± 0.012	0.722 ± 0.007
CATBoost	**0.379 ± 0.016**	**0.762 ± 0.006**	**0.884 ± 0.005**	**0.812 ± 0.009**	**0.732 ± 0.005**

3.2.5 Final Training, Testing and Evaluation

The final last stage of the xP modeling was the final training, testing and evaluation of the chosen statistical learning model - CATBoost. The workflow was in principle the same as the one described within the model selection and visualized in figure

Table 3.2: Test set performance (ED and TD) (mean ± std)

Classifier	BSS	Macro-F1	ROC-AUC	Macro-Precision	Macro-Recall
Logistic Regression	0.356 ± 0.011	0.750 ± 0.005	0.891 ± 0.006	0.794 ± 0.008	0.723 ± 0.004
SVM	0.336 ± 0.011	0.733 ± 0.007	0.883 ± 0.003	0.799 ± 0.006	0.700 ± 0.008
Random Forest	0.411 ± 0.012	0.764 ± 0.003	0.908 ± 0.005	0.812 ± 0.004	0.735 ± 0.003
XGBoost	0.412 ± 0.021	0.778 ± 0.006	0.908 ± 0.007	0.803 ± 0.006	0.759 ± 0.006
CATBoost	**0.436 ± 0.018**	**0.788 ± 0.006**	**0.913 ± 0.007**	**0.821 ± 0.009**	**0.765 ± 0.006**

3.11 although with some key differences. First of, the hyperparameter optimization employed a slightly more thorough hyperparameter grid. Secondly, the results stage extracted more results by not only looking at the metrics defined in figure 3.11 but also extracted the actual predicted target probabilities as well as the correct target response variable for each testing observation. Moreover, from the outputted optimized pipeline the built in CATBoost feature importance scores were extracted.

Further, the whole workflow was cross validated 5 times namely the whole workflow was repeated for 5 different initial 80/20 train-test splits, although the 5 different train splits was the same *between* the ED-only and ED+TD models. During the cross validation the ROC-AUC score on the test set was tracked and the model with the best ROC-AUC score during the process was extracted and saved. Furthermore, that model was used to get permutation importance scores in terms of effect on the ROC-AUC metric as well as SHAP-values. The motivation behind this is that integrating a permutation importance and SHAP-values calculation for all cross-validated models would increase the already long computation time considerably. Furthermore, the information provided by these feature importances will still be very much relevant even though they were not cross validated.

For training the models no consideration to the class imbalance was implemented and that also applies to the models in the model selection stage. The motivation behind this is based on trials using class weights the calibration curves, see coming section 3.2.5.1, which describes how well the models predicted probabilities line up with real probabilities significantly worsened. Using class weights or other techniques like up sampling usually just shifts the decision boundary more in favor of the minority class then before which results in that the model better find and classifies the non accurate passes, namely better recall, but the precision significantly worsens. In the end as the predicted probability realism is the most important success metric for an xP model and not the classification performance, it makes a lot more sense to let the model learn the natural distribution of the data.

Lastly, the evaluation for the final CATBoost models goes beyond the general metric and feature importance scores. To investigate the xP model probability reliability several strategies were employed to try and understand their behavior which are outlined in the coming subsections.

3.2.5.1 Probability behavior analysis

The first step was a analysis of the general behavior of the predicted probabilities from the model using only ED and the model using ED+TD information. For the each of the 5 cross-validated models their predicted test probabilities were extracted and then used to generate a calibration curve and these 5 calibration curves were then averaged into a final calibration curve. Furthermore, a more general exploration of the models' probability behavior was explored by viewing the probability distributions, average predicted probabilities and comparing average paired differences in probability prediction. This was done for all test passes then only accurate test passes and lastly only non accurate test passes.

3.2.5.2 High dimension visualization with predicted probabilities

The test observations were plotted in two dimensional space together with their respective predicted probabilities using UMAP embedding. As the EDA seemingly showed that there was little difference between 20 and 100 number neighbors for the UMAP plot, this stage only plotted the UMAP embedding for 20 neighbors. This evaluation and method can visualize the structure of the data and where the model decision boundaries may lie which in turn can possibly show if there exist clear boundaries and probability gradients which could entail better probability predictions.

3.2.5.3 Evaluating specific pass types

Some difficult pass situations were defined based on domain knowledge and certain extreme feature values within the numerical feature distributions. For each of these difficult pass types a paired permutation test was utilized to compare the probability distributions of the ED and ED+TD models. Further, the test used the test statistic of the mean paired differences between ED+TD models xP values and the ED-only models xP values for the paired samples. The null hypothesis was that the difference was 0, namely that the samples are from the same distribution, against the one sided alternative hypothesis that it's less then 0 indicating a smaller mean xP values from the ED+TD models. Lastly a low significance level of 0.01 was used to ensure reasonable inference. The permutation test was executed using the "permutation_test" method from the Scipy library.[6] For more information on the theory behind the permutation test the reader is referred to [19].

The difficult pass situations which were chosen to be evaluated were:

1. Passes with a distance above 60 meters.
2. Passes with a progressive distance above 50 meters.
3. Passes with direct press value above 0.4.
4. Passes with indirect press value above 100.
5. Passes with passing lane press value above 0.9.

3.2.5.4 Evaluating feature distributions within misclassifications

As the ED-only model did not have access to the TD features it was interesting to view whether feature distributions differed between the ED only and ED+TD model within the misclassifications. A model having a higher direct press value among the false negative predictions arguably makes *better* false negative predictions and judges passes better. Importantly, a Wilcoxon rank sum test, was conducted on the specific feature distributions between the models, first on their false positives and then on their false negatives. The test looked at whether the models' false positive and false negative predictions occupies different regions within the specific feature space by utilizing the p-value of the test in relation to a significance level of 0.01. Furthermore, the rank sum test is non parametric but assumes similar distribution shapes which could be assumed based on consulting the feature distributions for both the false positives and false negatives, see figure A.5 for examples. This assumption means that the result of the rank sum test was interpreted as comparing the medians of the different models' feature distributions with the null hypothesis that the medians were the same compared against a two sided alternative hypothesis. For more theoretical information about the rank sum test the reader is referred to [24].

3.2.5.5 Manual evaluation

The last evaluation was a manual evaluation using the saved best models as well as the set aside game mentioned in section 3.2.2.1. The best performing models gave xP values for all the passes in that game which were then manually assessed using a recording of the set aside game.

3.3 Developing a playability KPI

This section presents a football analytics statistic called **xPlay** which measures the *expected playability* of a player during a game. The section will in the coming subsections explain the methodology and assumptions behind the KPI as well as how it can be applied in football data analytics.

3.3.1 Defining playability and xPlay

Playability is as with many aspects in football hard to precisely define but a fitting overall description of playability could for example be *how much a player make himself available for passes*. This thesis defines the playability of a player based on two aspects:

How easy is it to successfully pass to a player?: This first aspect describes that for a specific player in a game, given a pass situation, how easy is it in general for the passer to find this specific player with a pass. Although, because of positional play in football certain players are not meant to provide *easy* passes in every passing situation as they need to stay in their designated position. Therefore this aspect is

mostly interesting to evaluate for players in closer proximation to the passing player. This first aspect can be quantified by the following pass probability for a player i over a game: $P^i(A\ pass\ to\ player\ i\ would\ be\ successful|Player\ i\ is\ within\ Q\ meters\ of\ passer)$, where the variable Q determines what the close proximation constitutes.

How often does a player position himself in close proximation to passes?: As the first aspect describes how easy it is to reach a player with a pass when the player is positioned in a close proximation to a pass, the natural next step is to look at how often a player position himself in this close proximation. A player showing great playability will not only be very easy to reach when in close proximation but will also *regularly position themselves to be in close proximation.* The second aspect can be quantified by the following probability for a player i: $P^i(Player\ i\ is\ within\ Q\ meters\ of\ passer)$.

These definitions of the two main playability aspects can then be used to develop the definition of the xPlay KPI. Let S = A pass to player i is successful and C = Player i is within Q meters of passer then xPlay is defined by the following joint conditional probability under the chain rule:

$$xPlay^i = P^i(S \cap C) = P^i(S \mid C) \times P^i(C) \tag{3.18}$$

xPlay is a probabilistic metric that the describes the joint probability that for a pass situation in a game that a player is both within a certain playable distance, and that the pass to the player would be accurate. This metric captures in an elegant and simple way on a player level both *how often* and *how easy* in a single value.

3.3.2 Implementation of xPlay utilizing TD and xP model

To be able to actually use equation 3.18 using data involved several steps. The xPlay KPI is sanctioned on the availability and use of ED and TD in synchronized combination as well as an accurate xP model. Below the concrete steps will be outlined.

ED and TD synchronization for passing situations: The first step was to get TD regarding player positions for each pass event during a game which requires either TD with passing event information already integrated or a synchronization of ED and TD. Furthermore, for the xPlay KPI the only interesting pass events are where a potential receiver has to make their own decision regarding which space to occupy. Therefore it becomes natural to disregard pass events in the ED that are corners, free kicks or other set pieces. To see the method for synchronizing pass events between ED and TD see section 3.1.2.

Creating artificial passes: The next step was to create artificial passes using the synchronized ED and TD. For each real pass event within a game teammates to the passer that was within the close proximation threshold Q was extracted. In discussion with supervisors at the collaboration company PlaymakerAI a threshold value of $Q = 20$ meters was used for the implementation of xPlay for this thesis. For each

of the teammates that were within the close proximation threshold a hypothetical real pass was created, see figure 3.12.

Figure 3.12: All images show tracking frames for real pass moments with artificial passes to teammates within a 20 meters distance threshold.

Moreover, for each created artificial pass to a teammate, pass features were extracted. These pass features corresponds to the features used for a specific xP model. For this thesis the natural choice becomes to use the new xP model utilizing both ED and TD data and therefore the features of interest are the same features as in table A.1. Most of these features regards passer or receiver information and relations which are trivial to extracting from the TD or if they already exist in the ED. Although, there does exist two features that proved harder to implement as they both regard specific information about how the pass is executed. The first feature is the "height_of_pass" feature and this was dealt with by assuming each artificial pass must be played on the ground. The next feature is the "speed_of_pass" and in contrast to the pass height this feature is actually more fair and objective if the value depended on the specific pass distance. Therefore to ensure fairness between artificial passes and fair comparable xP values, the speed of the pass feature was calculated by first dividing the passes from the final dataset from the 27 games in section 3.2.2.1 into 50 sections based on their pass distance. Further, the average pass speed of each distance section was calculated and as visualized in figure 3.13. There exists a quite clear linear correlation in the region of around 0 to 40 meter pass distance. As only passes between 0 to 20 meters are interesting this correlation was used to for each artificial pass place it in a distance section and then using the average pass speed of that distance section as the pass speed for the artificial pass.

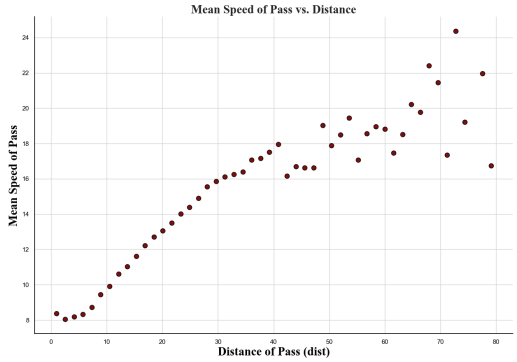

Figure 3.13: Dataset mean of feature "speed_of_pass" plotted against sections of feature "dist". Linear correlating behavior between 0-40 meters can be seen.

Predicting artificial passes and calculating xPlay: The last step was to use the artificial passes as input to an xP model that predicted the success probability and these xP probabilities could then be turned into xPlay values. First step was to estimate

$$P^i(\textit{A pass to player } i \textit{ would be successful} | \textit{Player } i \textit{ is within } Q \textit{ meters of passer})$$

Using the law of total probability for conditional probabilities and let S = A pass to player i is successful, C = Player i is within Q meters of passer and $\{B_n^i \mid n = 1, 2, 3, \dots\}$ be the set of pass events where player i was within 20 meters[38]:

$$P^i(S \mid C) = \sum_n P^i(S \mid C \cap B_n^i) P^i(B_n^i \mid C) \tag{3.19}$$

Here $P^i(S|C \cap B_n^i)$ can be estimated by the xP model's predicted probabilities for the artificial passes within 20 meters range for player i as each of these artificial xP probabilities is in principle stating *what is the probability of a successful pass to player i* ***given*** *that they is within 20 meters for a specific pass event.* Furthermore it is then a bit naively assumed that each pass event where player i was within 20 meters, namely $P^i(B_n \mid C)$, was equally likely. Further, let N^i denote the number of pass events where player i was within 20 meters of the passer. Then, using the mentioned approximations, the probability of player i successfully receiving a pass conditioned on being within 20 meters is given by:

$$\hat{P}^i(S \mid C) = \frac{1}{N^i} \sum_{n=1}^{N^i} P^i(S \mid C \cap B_n^i) = \frac{1}{N^i} \sum_{n=1}^{N^i} \mathrm{xP}_n^i \tag{3.20}$$

Lastly, let L^i denote the number of unique pass events that his team made ***during player i's specific time on the pitch***.

$$P^i(C) = \frac{N^i}{L^i} \tag{3.21}$$

By for each player calculating equations 3.20 and 3.21 and then implementing them into equation 3.18, the xPlay KPI was calculated for each player of a game.

Important aspect of the xPlay KPI is that it's not weighted by the time a player spent on the pitch which causes unreliability for players that comes on very late in games. Further, for these types of players, their team might only be able to do 5 to 10 passes and just by being within 20 meters of all of them or perhaps none of them, the xPlay could be extremely high or extremely low. Therefore, another hyperparameter in complement to the playability threshold Q is established to regulate if a player's xPlay should not be calculated based on the amount of passes that the player's team has been able to do during their time on the pitch. The threshold value used within this thesis during the implementations described in the next subsection is 50 total team passes while on the pitch. This was a manually tuned threshold motivated by conversations with supervisors within the collaboration company PlaymakerAI. Both of the xPlay hyperparameters can be changed and tuned depending on user preference of the KPI which align with several other analytical KPI structures within the PlaymakerAI platform.

3.3.3 Implementation and evaluation of potential xPlay applications

Three possible applications of the xPlay KPI are proposed and all of them were evaluated using the set aside game mentioned in section 3.2.2.1. The first possible application is to use it in a *standard* format by simply calculating xPlay for each player for a game. The players' xPlay values can then be sorted for each team which shows players who provided the highest playability within each team. Furthermore, it is also possible to compare full game player xPlay scores between the different teams.

The second potential application is ***xPlay evolvement over time*** during a game. This would involve not calculating the xPlay for an entire game but instead consider several subsets of the game. First a moving time window of 10 minutes that moves one minute forward across the entire game is established. Then every player's xPlay within this window is calculated for each step in time and by doing this it is possible to see how players' xPlay evolve during the game.

Lastly, a third application is to instead look at a spatial context by considering ***xPlay within different zones of the pitch***. The implementation used the "Juego de Posicion" zones which are positional zones popularized by the infamous Spanish coach Pep Guardiola and the zones are identified as important zones for creating structure within a game.[13] By treating each zone as its own smaller pitch and only using the subset of pass events that occurred for a player's team within those zones, the xPlay KPI was calculated for each of the Juego de Pocision zones. This implementation can show how players differ in which zones the players are more or less playable.

4
Results

This chapter presents the major relevant results from the project relating to first the overall data preparation in terms of processing and synchronizing TD with ED. Secondly, xP modeling and comparison of a model using only established ED information against a model using ED information with integrated TD information. Lastly, results and evaluations regarding the applications of the proposed xPlay KPI.

4.1 Synchronization and TD processing results

The synchronization algorithm in combination with post processing as well as some minor refinement employed during TD feature engineering and the EDA resulted in 19,659 synchronized passing events from 27 total games. Added to this was a final 28th game that was used for manual evaluation. The 27 games that constituted the the final complete dataset in total consisted of 26,287 passes originally, meaning that the whole synchronization and processing process yielded a result of a $\approx$ **74.79**% success rate in attaining TD information for ED pass events. Moreover, the actual accuracy of these synchronizations is hard to estimate precisely but thorough manual evaluation of around 100 pass events for each half from each of the total 28 games indicated a synchronization accuracy of **90**% at least. The total stochastic time shifts for all 28 games are shown in figure 4.1, further the figure shows a distribution with the appearance of a normal distribution around the value of 0 which would seem as very realistic.

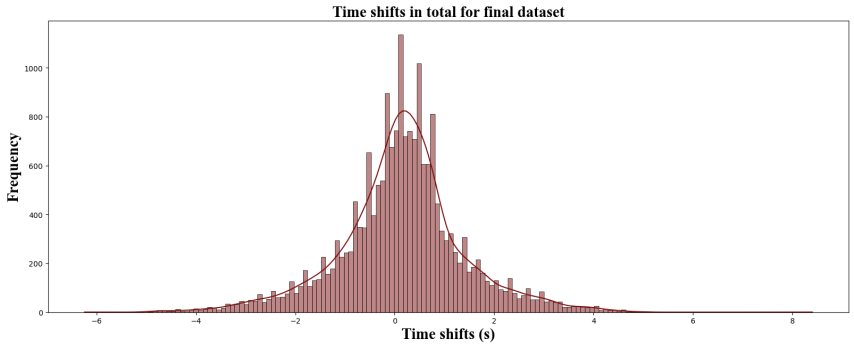

Figure 4.1: Distribution of timestamp shifts for passing events between ED and TD.

4.2 xP modeling and comparison

In the following subsections the results of the different general and in depth evaluation strategies are presented as well as shortly analyzed.

4.2.1 General metric performance and Feature importance

Table 4.1 shows the cross validated metric performance between the models trained on ED only and models trained on ED+TD. Seemingly the models using ED+TD data performs better within each metric both on a general level but also on a class specific level. The only metric that the ED-only model performs just slightly better on is the test recall for accurate passes but it is minimal. At first sight comparing the general test and train results for Brier skill score, F1, Precision and Recall metrics seems to indicate some overfitting for the models. Further, consulting the class specific performance tables shows that the potential overfitting is caused by the non-accurate class.

Table 4.1: Train and Test set performance (ED-only vs. ED + TD)

Train set – General metrics					
Dataset	BSS	Macro-F1	ROC-AUC	Macro-Precision	Macro-Recall
ED only	0.4853 ± 0.0116	0.8065 ± 0.0065	0.9178 ± 0.0042	0.8685 ± 0.0067	0.7692 ± 0.0061
ED + TD	**0.6522 ± 0.0586**	**0.8882 ± 0.0282**	**0.9635 ± 0.0119**	**0.9313 ± 0.0294**	**0.8564 ± 0.0269**
Train set – Class-specific metrics					
Dataset	Prec. Non-accurate	Prec. Accurate	Recall Non-accurate	Recall Accurate	
ED only	0.8200 ± 0.0115	0.9171 ± 0.0020	0.5633 ± 0.0108	0.9751 ± 0.0015	
ED + TD	**0.9155 ± 0.0500**	**0.9470 ± 0.0088**	**0.7263 ± 0.0459**	**0.9865 ± 0.0079**	
Test set – General metrics					
Dataset	BSS	Macro-F1	ROC-AUC	Macro-Precision	Macro-Recall
ED only	0.3644 ± 0.0225	0.7549 ± 0.0106	0.8775 ± 0.0090	0.8034 ± 0.0116	0.7257 ± 0.0099
ED + TD	**0.4252 ± 0.0165**	**0.7827 ± 0.0082**	**0.9098 ± 0.0058**	**0.8158 ± 0.0070**	**0.7592 ± 0.0089**
Test set – Class-specific metrics					
Dataset	Prec. Non-accurate	Prec. Accurate	Recall Non-accurate	Recall Accurate	
ED only	0.7032 ± 0.0205	0.9036 ± 0.0032	0.4933 ± 0.0180	**0.9580 ± 0.0031**	
ED + TD	**0.7160 ± 0.0114**	**0.9156 ± 0.0031**	**0.5635 ± 0.0172**	0.9549 ± 0.0017	

Moreover, evaluating the CATBoost built in feature importance figure 4.2 shows that for the ED model progressive distance, general distance as well as the x and y destination coordinates have the biggest importance. When adding the TD features, the overall importance dilutes somewhat but the main features now seems to be general distance, progressive and side distance, passing lane pressure, smallest passing lane distance to an opponent, smallest angle between an opponent and the passing lane as well as the pass height. Furthermore, binary features for corner, freekick and throw in seems to provide very little importance if any at all seemingly agreeing with target correlations from the EDA.

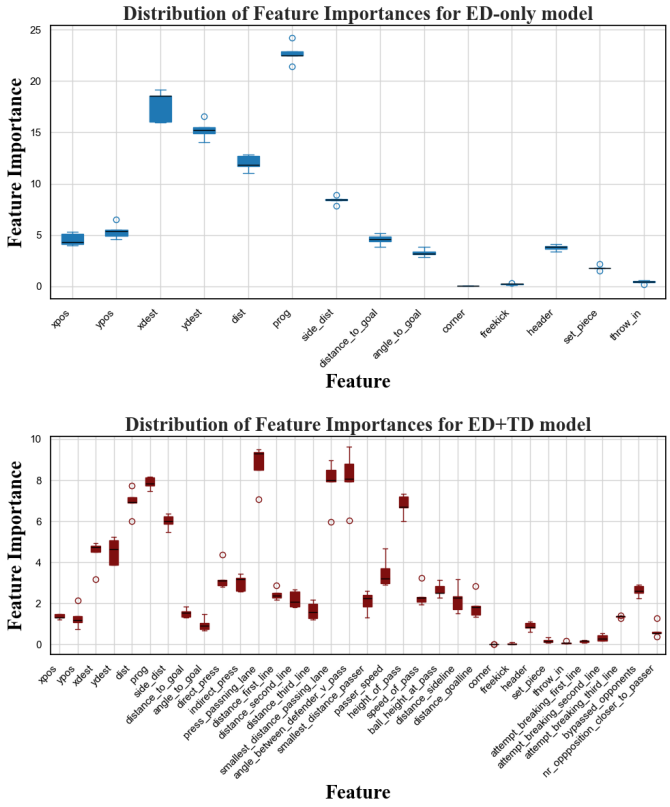

Figure 4.2: CATBoost built in feature importance scores for ED-only model and ED+TD model.

The permutation importance scores for the best performing ROC-AUC models presented figure 4.3 in the case of the ED model seems to mostly agree with the built in feature importance. Although, the permutation importance suggest that the progressive distance feature is even more important than the other features by providing up to a 0.14 value decrease in ROC-AUC. Regarding the model using ED and TD features the difference to the built in feature importance is distinct. The established ED features still show similar rankings but for the added TD features the smallest angle between the passing lane and an opponent dominates with also surprisingly the binary features ranked slightly higher. The permutation importance therefore seem to suggest that the information contained in several of the high performing CATBoost feature importance TD features, are shared with one or a combination of other TD features. Very important to note though is the actual overall scores as they are very small, the most important permutation feature only decreases the ROC-AUC score with about 0.02 which suggest that models trained on the new TD does not solely rely on any single or couple of features.

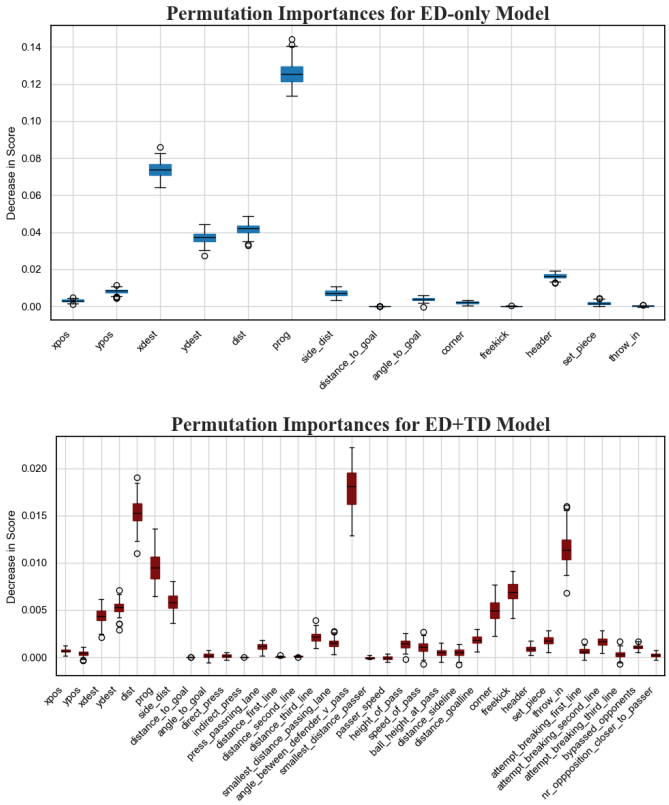

Figure 4.3: Permutation importance scores in regards to ROC-AUC for best ED-only model and best ED+TD model.

Furthermore, the SHAP-values for the best performing ROC-AUC models presented in figure 4.4 seems to agree with the rankings from the CATBoost feature importance by having the same top 5 features. It is also seen that high values for progressive distance and x destination drags the observation predictions down towards a non-accurate prediction drastically. For the model using both ED and TD features the top 5 ranked features also very much correlate with the top features from the CAT-Boost feature importance. More interestingly is that the top ranked features have a much better separation regarding feature value impact on model output compared to the top ranked features for the ED-only model indicating better correlation with the target. The features are not as stretched along the impact on model scale as well which indicates that no single feature can dominate the impact on the model.

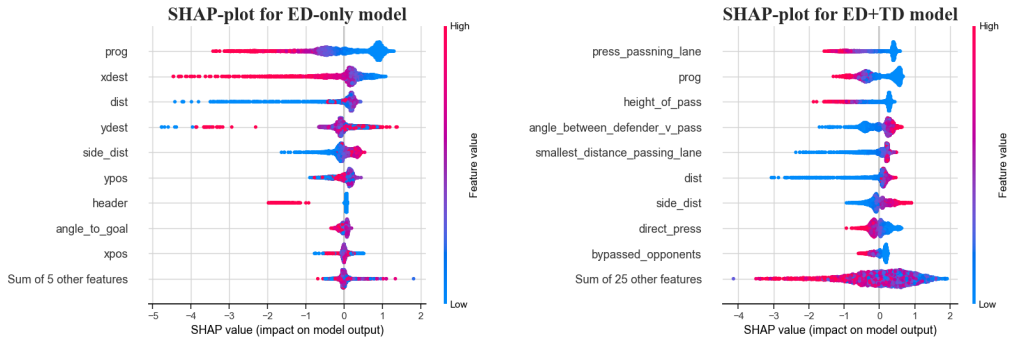

Figure 4.4: SHAP-values for top 9 features + remaining features for best ED-only model and best ED+TD model.

4.2.2 Probability behavior

The calibration curves in figure 4.5 show that the ED-only models seemingly have more stable probability predictions for the higher probability range while the ED and TD model have quite stable probability predictions at the vary high range and also quite stable in the lower region while the middle region is a bit more stochastic. Furthermore, consulting the mean calibration curve suggests that the ED and TD overall provide just slightly better results.

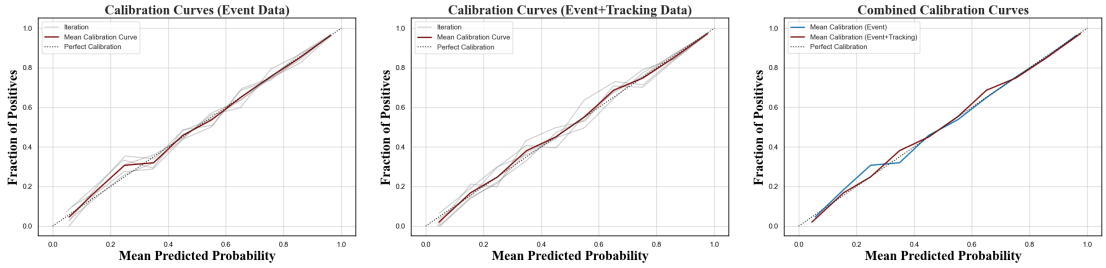

Figure 4.5: Calibration curves from cross validation as well as their mean calibration curve for ED-only models (left), ED+TD models (middle) and also their mean calibration curves together (right).

Table 4.2 first shows that the average probability is quite similar between the different models although looking at the paired differences and the probability distributions in figure 4.6 there is clearly different behavior. Seemingly the models using ED and TD features predicts a lot more observations at extremely high probabilities as well as more probabilities at generally lower values. Further, the probability interval counts also suggest that in general the ED and TD models are either extremely confident or more restrictive. Further, the predicted probability differences for the same individual test samples are both in terms of mean and median about 1% in percentage points higher for the accurate passes while 3-5% in percentage points less for non-accurate passes indicating a bigger disagreement for the non accurate passes. This is further confirmed by the minimum and maximum predicted test sample probability difference which both stem from the non accurate pass events.

Table 4.2: Probability Evaluation Summary (Event vs Event+Tracking)

Predicted Probability Distributions				
Model	Mean ± Std	Median	Min	Max
Event	0.8313 ± 0.2292	0.9354	0.0053	0.9989
Event+Tracking	0.8336 ± 0.2477	0.9616	0.0024	0.9997
Probability Differences (ED+TD – ED)				
Group	Mean ± Std	Median Diff	Min Diff	Max Diff
Overall	0.0023 ± 0.1162	0.0118	-0.8221	0.7929
Accurate Passes	0.0127 ± 0.0958	0.0137	-0.6437	0.7062
Non-accurate Passes	-0.0493 ± 0.1781	-0.0307	-0.8221	0.7929
Probability Interval Counts (A - Accurate pass, N - Non-accurate pass)				
Model	[1.0–0.75] A/N	[0.75–0.5] A/N	[0.5–0.25] A/N	[0.25–0.0] A/N
Event	14490 / 1013	1140 / 654	519 / 837	166 / 786
Event+Tracking	14353 / 766	1227 / 670	568 / 915	167 / 939

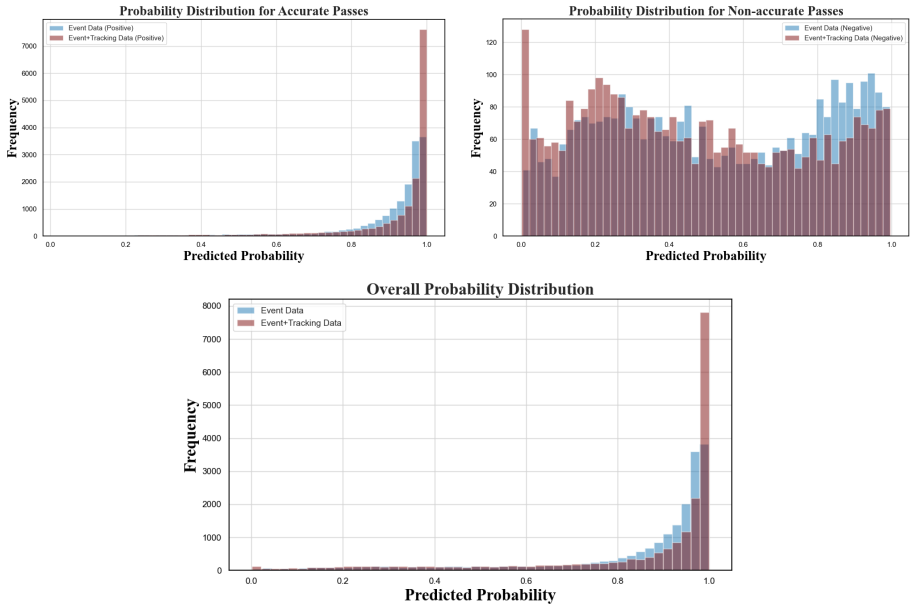

Figure 4.6: Probability distributions from the test observations for the different model types. Top left subplot shows the probability distribution for the accurate passes while the top right subplot shows the probability distribution for the non-accurate passes. Lastly the bottom subplot shows the total probability distribution.

4.2.3 High dimension visualization with predicted probabilities

Figure 4.7 shows that the high dimensional structure using both ED and TD features is much more coherent as established in section 3.2.2.1. But with the predicted probabilities, it also also seen that there exists a very concise and clear probability gradient which in turn seemingly creates a distinct decision boundary produced by the models using both ED and TD features. In contrast, the test samples when only using ED data are clustered more tight and unstructured with seemingly a lot more complex decision boundaries needed. These plots seem to indicate that using ED and TD features gives better and a much more coherent information landscape regarding types of passes which is then in turn navigated with more confidence and

ease by the models using this data. This potentially indicates better understanding of the passing landscape by the models using both ED and TD data and therefore potentially more trustworthy probabilities.

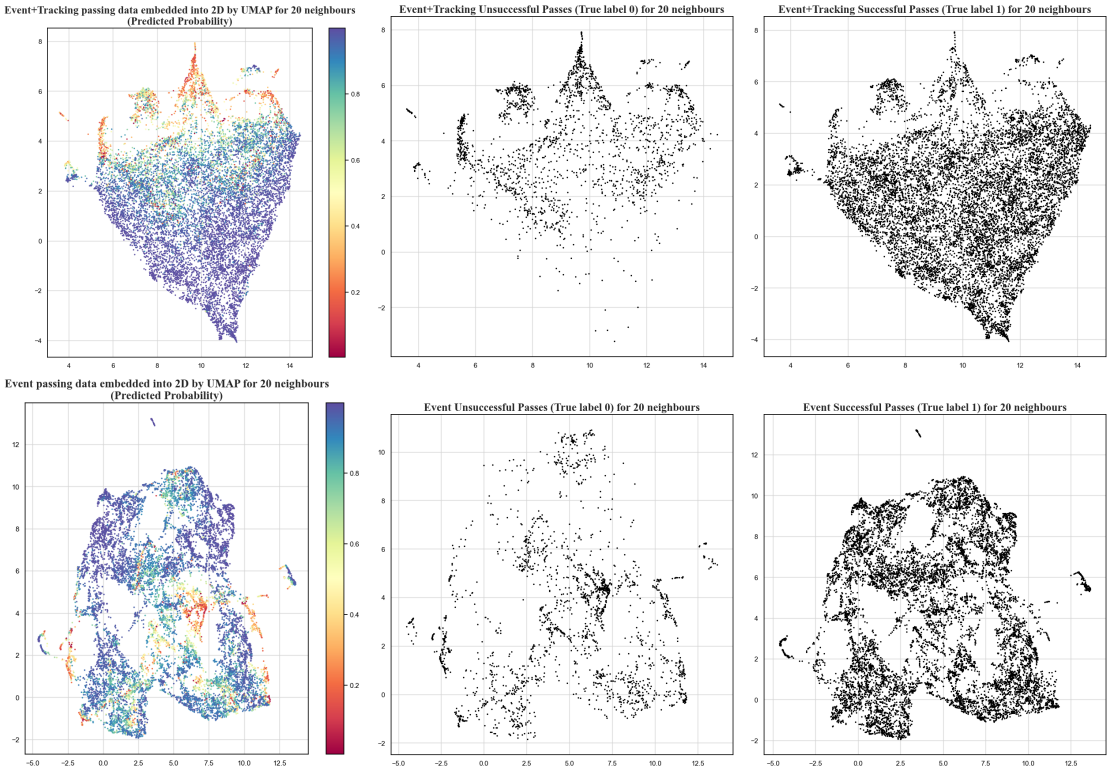

Figure 4.7: All subplots shows UMAP embedding of the same test samples in two dimensions while the top row shows the embeddings using the ED and TD features while the bottom row uses only ED features. Further, subplots to the left shows the samples labeled using their predicted probability, the middle subplots shows the locations of the non-accurate passes and lastly the right subplots shows the accurate pass locations.

4.2.4 Pass type performance

Table 4.3 and figure 4.8 shows that regarding the selected difficult passes the ED-only models seem to predict higher probabilities on average. Although, consulting the p-values from the permutation test in regards to the significance level of 0.01 suggest that this inference can only be done for "prog", "direct_press", "indirect_press" and "press_passing_lane" features.

Table 4.3: Predicted mean probabilities for specific types of passes (Event vs. Event + Tracking)

Filter Condition	Model	Mean ± STD	Median	p-value
dist > 60	**Event**	**0.2828 ± 0.1932**	0.2253	0.0366
	Event+Tracking	0.2596 ± 0.1428	0.2163	
prog > 50	**Event**	**0.2537 ± 0.1445**	0.2341	0.0053
	Event+Tracking	0.2322 ± 0.1039	0.2155	
direct_press > 0.4	**Event**	**0.7848 ± 0.2359**	0.8788	0.0003
	Event+Tracking	0.6915 ± 0.2827	0.8016	
indirect_press > 100	**Event**	**0.6733 ± 0.2925**	0.7890	0.0001
	Event+Tracking	0.5351 ± 0.2963	0.5889	
press_passing_lane > 0.9	**Event**	**0.4859 ± 0.2846**	0.4214	0.0001
	Event+Tracking	0.4519 ± 0.2638	0.3844	

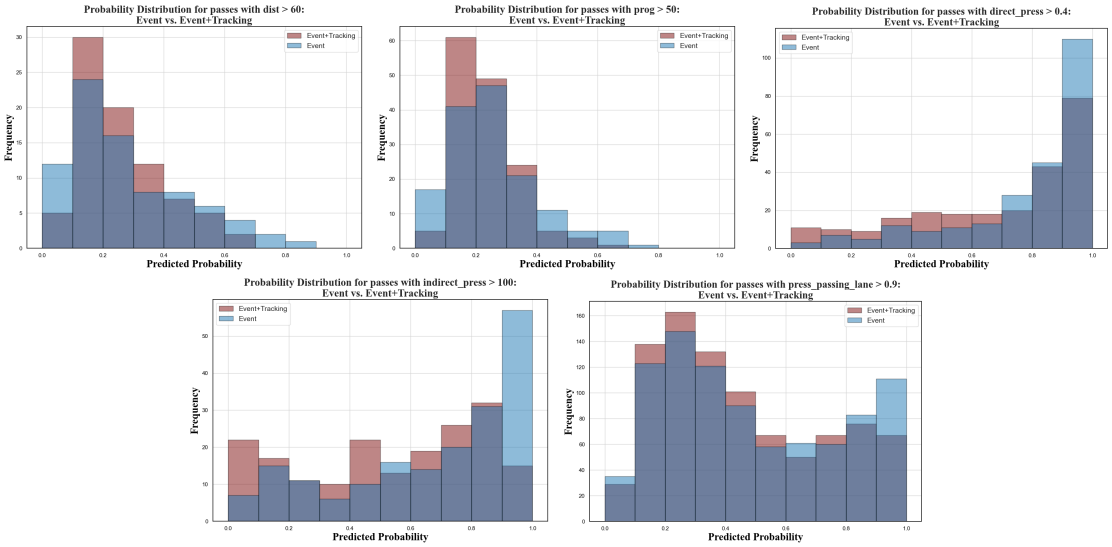

Figure 4.8: All subplots show the test probability distributions for specific type of difficult passes based on feature values. Top row in order from left to right: dist > 60 - prog > 50 - direct_press > 0.4. Bottom row in order from left to right: indirect_press > 100 - press_passing_lane > 0.9.

4.2.5 Misclassifications evaluation

As mentioned in section 3.2.5.4, a rank sum test was used to determine if there was a significant difference between the models' feature distributions within the misclassifications. Only the features which had p-values below the significance level of 0.01 for both the false positive and false negative distributions are presented in this thesis. By consulting figure A.5 as well as tables 4.4 and 4.5 it is shown that there

are five features that have significant distribution differences among the misclassifications. For each and single one them the ED+TD models give more reasonable values, namely, for example within the indirect pressing feature the ED+TD models gives higher mean and median indirect pressure values for the false negatives and lower mean and median for the false positives. This means that the ED+TD models makes more reasonable *wrong* predictions for the accurate and non-accurate predictions in terms of the indirect pressure feature. This behavior is the same for all the remaining four features as well.

Table 4.4: Rank sum test p-values comparing Event vs Event+Tracking Models (Feature Distributions in FP and FN Cases)

Feature	FP p-value	FN p-value
`indirect_press`	0.0020	<0.0001
`press_passing_lane`	<0.0001	0.00101
`smallest_distance_passing_lane`	<0.0001	<0.0001
`angle_between_defender_v_pass`	<0.0001	<0.0001
`height_of_pass`	<0.0001	<0.0001

Table 4.5: Mean and Median Feature Values in False Positives (FP) and False Negatives (FN) for Event and Event+Tracking Models

Feature	False Positives (FP)				False Negatives (FN)			
	Event		Event+Tracking		Event		Event+Tracking	
	Mean ± STD	Median	Mean ± STD	Median	Mean ± STD	Median	Mean ± STD	Median
`indirect_press`	37.21 ± 33.48	31.81	32.52 ± 31.16	26.01	27.15 ± 28.22	20.28	35.04 ± 31.83	31.89
`press_passing_lane`	0.39 ± 0.33	0.36	0.33 ± 0.33	0.25	0.56 ± 0.39	0.67	0.66 ± 0.31	0.73
`smallest_distance_passing_lane`	1.81 ± 1.72	1.41	2.14 ± 1.80	1.77	1.66 ± 2.03	1.15	0.97 ± 0.93	0.68
`angle_between_defender_v_pass`	10.90 ± 13.62	5.74	12.99 ± 14.61	7.63	8.78 ± 14.74	3.86	5.05 ± 8.84	2.51
`height_of_pass`	1.34 ± 2.46	0.17	0.97 ± 2.15	0.07	2.88 ± 3.50	1.80	3.48 ± 3.51	2.74

4.2.6 Manual evaluation

The manual evaluation was done using the set aside pass data from the game ***Real Madrid - Borussia Dortmund at the date 22.10.2024.*** Using the predicted xP values from the best performing ED-only model and the best performing ED+TD model for the passes in the game in combination with a recording of the game these predicted xP values were evaluated manually.

The main results were that overall the TD+ED model and ED-only model in most cases both provide reasonable xP values. The main differences occur for rare type of accurate passes as well as inaccurate passes where in general the TD+ED usually predict better probabilities. Furthermore, there seemed to be clear patterns as to when the different models generally became more careful and restrictive with their probability predictions. For the ED-only model having less context about the passes seems to result in that the destination of the pass becomes a big factor which align with the feature importance. As soon as a pass is placed quite high up on the pitch the ED-only model becomes restrictive with it's predicted probabilities.

Furthermore, this is not reasonable behavior as passes placed high up on the pitch generally might be harder to execute but it depends completely on the actual pass itself. There existed several pass events where the ED-only model gave an extremely easy pass a too low predicted probability seemingly because it happened very far up on the pitch. An example can be seen in figure 4.9 where an extremely easy pass is made under practically no pressure and no presence around the passing lane yet it's predicted ED-only xP is quite low under the circumstances.

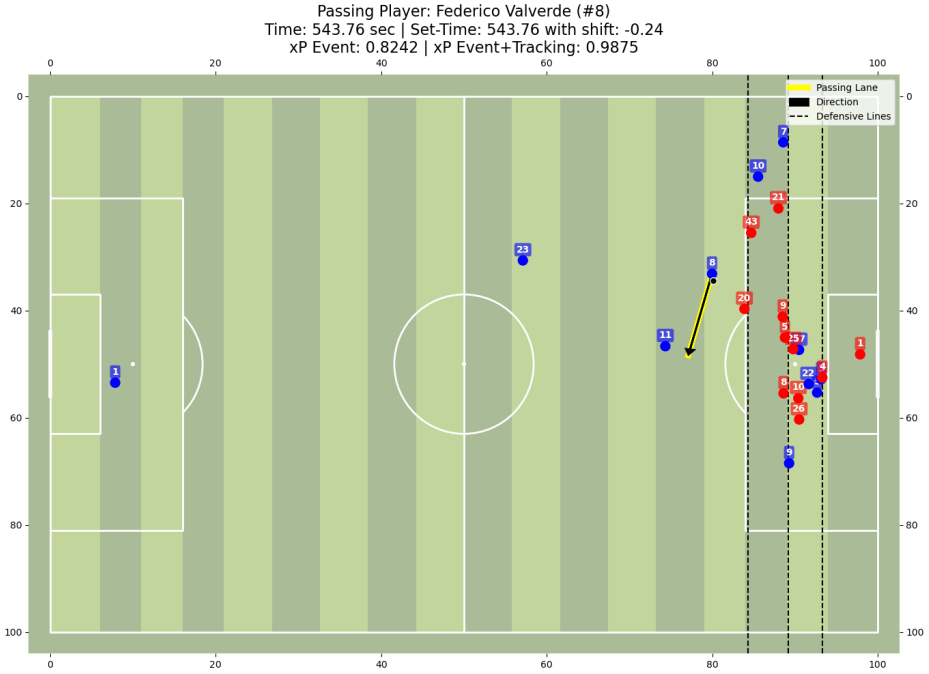

Figure 4.9: Pass event with predicted xP values for ED-only model and ED+TD model showing ED-only restricting probabilities of a simple pass.

Moreover, the ED+TD model seemed to restrict it's probabilities more based on the opposition presence around the area of the passer. This is a more reasonable behavior and a much better indication of a potentially more difficult pass in comparison with the pass destination. Although, it seemed that at some rare events the ED+TD model slightly might have overestimated the impact of the opposition presence but as mentioned this is a more reasonable excuse and feature to be cautious about.

Lastly, for the passing events where the two models disagreed on the classification of the pass, namely accurate or non-accurate, the ED+TD model was correct in the absolute majority of cases. Furthermore, in situations where the ED-only model was correct in regards to the actual outcome and the ED+TD model was not, there were several of these situations were the ED+TD model actually provided more realistic probabilities. An example is given in figure 4.10 where the two models disagree and the actual outcome was an accurate pass but as viewed in the figure the passer (red color) has one opponent (blue color) in very close proximation pressing him and standing in front of the pass. Manual evaluation confirmed that this pass event was indeed a very difficult pass which the ED+TD model captured better even though

the outcome actually happened to be an accurate pass.

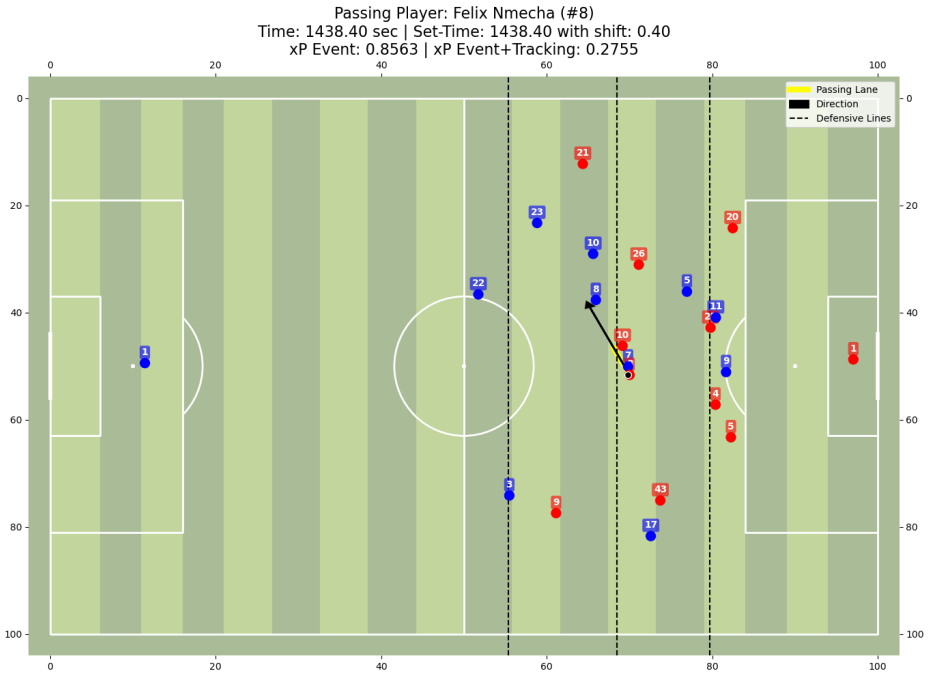

Figure 4.10: Pass event with predicted xP values for ED-only model and ED+TD model showing a disagreement between models but ED+TD model gives better probability prediction.

4.3 xPlay implementation

This section presents the results of the xPlay KPI implementation and applications. Further, the KPI was calculated on the set aside passing data from the game ***Real Madrid - Borussia Dortmund at the date 22.10.2024.*** The applications used the availability distance threshold of 20 meters for calculations as well as cut-off threshold of 50 total team passes during the player's time on the pitch for a player to receive an xPlay value.

4.3.1 Full game xPlay comparison

Tables 4.6 and 4.7 show the sorted xPlay values from the full game. The xPlay values in general show reasonable results as central midfielders and centre-backs rank the highest in both teams and goalkeepers and wingers rank last. This is realistic and reasonable for most modern possession based teams as these positions will see the ball the most and therefore it makes sense they are the most playable positions. The most useful results from the tables comes from comparing the players with similar positions within their own teams and against the opposition team. Dortmund started the game with two deeper central midfielders in Felix Nmecha and Marcel Sabitzer and the xPlay values indicates that Felix Nmecha had a 15% in percentage points higher probability of being within 20 meters and receiving a successful

pass. Felix Nmecha also outperforms the starting deep central midfielder of Real Madrid in Federico Valverde in regards to xPlay. Furthermore, comparing the strikers and wingers of both teams, Kylian Mbappe of Real Madrid shows clearly lower playability in comparison with the Dortmund striker in Sehrou Guirassy. Moreover, Dortmund's Jamie Bynoe-Gittens and Donyell Malen also shows higher playability than their Real Madrid counterparts in Vinicius Junior and Rodrygo. What this might entail is that the Dortmund players in general have more freedom in moving around the pitch and being closer to pass situations while the Real Madrid team might utilize a more rigid structure with clear and strict positions.

Table 4.6: Hypothetical Receiving Stats and xPlay for Borussia Dortmund (sorted by xPlay; only players experiencing 50 total team passes shown)

Receiver	**Total Passes**	**Within 20m**	**%Involved**	**Mean xP**	**xPlay**
Felix Nmecha	324	271	0.8364	0.9041	0.7562
Pascal Groß	111	94	0.8468	0.8862	0.7505
Waldemar Anton	134	96	0.7164	0.9215	0.6602
Niklas Süle	387	257	0.6641	0.9355	0.6212
Marcel Sabitzer	387	271	0.7003	0.8765	0.6138
Julian Brandt	387	282	0.7287	0.8053	0.5868
Nico Schlotterbeck	387	233	0.6021	0.9484	0.5710
Donyell Malen	276	196	0.7101	0.7957	0.5650
Sehrou Guirassy	387	296	0.7649	0.7003	0.5357
Ramy Bensebaini	387	234	0.6047	0.8703	0.5262
Maximilian Beier	63	43	0.6825	0.6945	0.4740
Jamie Bynoe-Gittens	253	119	0.4704	0.8506	0.4001
Julian Ryerson	319	130	0.4075	0.8959	0.3651
Emre Can	68	22	0.3235	0.8917	0.2885
Gregor Kobel	387	80	0.2067	0.9708	0.2007
Team average					**0.5277**

Table 4.7: Hypothetical Receiving Stats and xPlay for Real Madrid CF (sorted by xPlay; only players experiencing 50 total team passes shown)

Receiver	Total Passes	Within 20m	%Involved	Mean xP	xPlay
Eduardo Camavinga	89	74	0.8315	0.9049	0.7524
Federico Valverde	422	341	0.8081	0.8853	0.7154
Antonio Rüdiger	422	305	0.7227	0.9187	0.6640
Eder Militão	422	294	0.6967	0.9227	0.6428
Luka Modric	333	234	0.7027	0.8935	0.6279
Jude Bellingham	409	310	0.7579	0.7446	0.5643
Ferland Mendy	422	237	0.5616	0.9503	0.5337
Kylian Mbappé	409	291	0.7115	0.6556	0.4665
Lucas Vázquez	422	229	0.5427	0.7982	0.4332
Rodrygo	396	230	0.5808	0.7130	0.4141
Vinicius Junior	422	189	0.4479	0.7543	0.3378
Thibaut Courtois	422	93	0.2204	0.9898	0.2181
Team average					**0.5309**

4.3.2 Temporal xPlay comparison

Consulting the temporal change of player xPlay during the game displayed in figure A.6 suggests that the overall playability for Dortmund's players have a stochastic behavior. More importantly, the general trend is that the majority of player's playability decreases with time. The exceptions are Felix Nmecha and Marcel Sabitzer who over the entire game keeps a stable playability which in the case of Nmecha is impressive as he produces quite high xPlay values as well. Moreover, some of the substitutes in contrast increase their playability over time but this is quite expected as they come in with new energy at later stages of the game.

For Real Madrid the playability trend during the game is less stochastic and most players either increase or have their playability constant during the game. The results for both teams can partly be explained by the result of the game as Real Madrid won the game 5-2, furthermore Real Madrid scored all their 5 goals in the second half. Therefore it seems reasonable that during this type of game it becomes harder mentally for the Dortmund players to stay in the game which actually once again makes the temporal playability performance from Felix Nmecha impressive. Furthermore, it is also possible that the difference between the teams is also due to the a possible different tactical philosophy between the teams as mentioned in the previous subsection. Namely, if Dortmund apply a possession philosophy with more freedom and where the players move a lot more when the team is in possession, this can also add to the Dortmund players becoming more tired which results in that the overall playability goes down over time. Real Madrid who might employ a more rigid positional structure require less movement in general, with perhaps the exception of central midfielders, and therefore it is easier for them to contain their playability over time especially when they are winning and playing a team that

seems to generally decrease their playability.

4.3.3 Spatial xPlay comparison

Lastly, figure A.7 shows how players' xPlay values differ between different zones on the pitch. Interesting again is comparing the starting deep central midfielders for Borussia Dortmund. Felix Nmecha has quite good playability over the entire pitch, especially from the wide zones, which is incredibly impressive and also aligns with his overall high xPlay for the entire game. In contrast, Marcel Sabitzer is most playable from very central zones and also makes himself more playable on the right hand side of the pitch in comparison with the left hand side.

Moreover, it was declared in an earlier section that Dortmund striker Sehrou Guirassy had a higher playability than his Real Madrid counterpart Kylian Mbappe. In figure A.7 it is also shown that the Dortmund striker not only has better overall playability but also has better playability for more zones than Mbappe which again seems to indicate that some Dortmund players most likely play with more freedom and are allowed to move around more.

Lastly, one other interesting aspect that the spatial xPlay comparison provides is that the Real Madrid wingers Vinicius Junior and Rodrygo differ quite a bit in where they are playable. Vinicius' playability is very focused on 2-3 zones while Rodrygo's playability is a lot more varied especially in the final third of the pitch seemingly indicating more static positioning of Vinicius Junior.

5
Discussion

This chapter analyzes and discusses the presented results in relation to the specific research questions outlined for the thesis. Further, the chapter presents some discussion about key methodology of the thesis in regards to certain assumptions as well as possible improvements.

5.1 Research Question 1

The first research question of this thesis was - *Can information from spatiotemporal tracking data be integrated into established event data to improve a KPI for the probability to complete a pass?* The short answer to this is simply yes and this is based on that every result presented for all evaluation methods in section 4.2 suggest that an xP model that uses both the established ED as well as the new TD outperforms a model only using the established ED.

The general test metrics shows better scores for the ED+TD models in essentially every metric. Furthermore, comparing with the results from Anzer.G and Bauer.P for their trials of different xP models the same results in regards to the improvement using TD features is observed. In contrast their model using all ED and TD features gives better performance than the same type of model in this thesis which might be explained by that they had a lot more data as well as a more complex and seemingly more accurate synchronization process.[9] Further, their data came from the same league which also is better from a modeling perspective as mentioned in the limitations section.

One noteworthy aspect is that their recall and F1 scores are generally very high which is surprising. Although, it is unclear if these scores are not 'macro'-weighted like in this thesis. Why this is surprising is because of the nature of predicting and classifying non-accurate passes. Everyone who has watched football, especially in the last years, know that a big section of non-accurate passes mostly comes down to mistakes and chance. This means that a lot of non-accurate passes have the exact appearance of usual simple accurate passes but by chance and human-error it becomes non-accurate. What this means is that even a person with expert football knowledge could not predict and classify a lot non-accurate passes correctly therefore from a modeling perspective it makes sense an xP model, if it has learned the data well, should also not have very high recall in terms of non-accurate passes, see coming section 5.6 for more in depth discussion about this. In contrast, as Anzer.G

and Bauer.P although mentions that precision, recall and F1 does not really make sense in terms of judging xP models and therefore scores like ROC-AUC and Brier skill score are more appropriate. In regards to these metrics the models using both ED and TD data of this thesis performs a bit closer to their best model which also used both ED and TD data.

Moreover, the more interesting evaluations for answering research question one has been the predicted probability behavior. By consulting calibration curves, predicted probabilities and decision boundaries in higher dimensions, probability behavior for difficult passes, feature distributions within misclassifications and manual evaluation, the ED+TD model shows better performance by having a more realistic behavior when judging and predicting pass probabilities. As the performance improvement has been confirmed throughout all of these evaluation strategies, it should be no doubt that the information from the TD with certainty improves the xP KPI.

5.2 Research Question 2

The second research question of this thesis was - *How does a model utilizing tracking data information differ in the specific outputted passing probabilities in comparison to a model only utilizing event data?* As just previously mentioned the ED+TD models clearly shows more reasonable and realistic probability behavior both in terms of specific difficult types passes, type of passes within the misclassifications as well as when the xP values were manually evaluated.

What seems to be the logical trend which explain this behavioral difference throughout the probability evaluations is that the models have clear differences in what they base their probabilities on, and this aspect is probably the most interesting aspect when answering research question two. This difference is shown clearly by the feature importance scores especially regarding the built in CATBoost model feature importance and the SHAP-values. Both of them shows very clearly that the ED-only model base it predictions almost exclusively on distances and destination coordinates of the pass. The permutation importance further confirms this statement as the progressive distance of the pass feature alone can decrease the ROC-AUC score with up to 14 percentage points. In contrast for the ED+TD model the feature importance is more diluted and shared across several features. Reasonably several features are still considered more important but the ranking disparity is not as huge as for the ED-only model. Once again by consulting the permutation importance this statement is confirmed as for the ED+TD model no feature alone can decrease the ROC-AUC score more than around 2 percentage points. The feature importances therefore show that the ED+TD xP models efficiently uses a lot more features and context in a balanced way while the ED-only models rely on 3-4 features.

Lastly, the high dimensional visualization with the predicted probabilities perhaps gives the best results as to what this additional context gives and can summarize the answer to how the models differ in their probabilities. With the UMAP visualization in combination with the predicted probabilities it is shown that ED+TD xP models

are able to view the passing landscape in a much more detailed and contextual way. In turn this enforces a much better and more clear defined structure in terms of areas with difficult and non difficult passes. Naturally this helps the model as it becomes a lot easier to navigate the passing landscape which results in a probability gradient that is a lot more coherent and defined across the landscape. In the end this entails a better understanding for pass events within football which improves classification but more importantly likely gives better and more realistic predicted probabilities.

5.3 Research Question 3

The third research question posed in this thesis was - *Can new knowledge in the form of a new type of KPI measuring player playability be achieved by using information from tracking data?* This thesis has provided a positive answer to this question by the developed xPlay KPI using TD and the new ED+TD xP model in developing and averaging hypothetical pass opportunities. xPlay or *Expected Playability* measure player playability by providing a probability measure for how likely it is for a player to be within close proximity to a pass and also how likely it is that the potential pass to the player would be accurate.

Furthermore, the metric has been shown to be able to provide interesting information through three different applications. The full game application show how xPlay can be used to capture playability over an entire game, providing information about which players have the most playability within their own team. Moreover, it can be used to show how players playing in similar positions differ in playability both within the same team as well as compared to the opposition team of a game. The second and third application allows to view xPlay in a more detailed way by exploring first the temporal aspect of how it changes during a game and secondly the spatial aspect by showing where on the pitch players are playable. These two additional applications provide more detailed game specific information that can be used to once again compare players within their own teams regarding if they drop or increase their playability over time, or perhaps if they are more or less static by being playable in more or less zones on the pitch.

Important aspect to discuss is that the metric is based on synchronized ED and TD data for this thesis and as established the synchronization algorithm does not result in all passing events being synchronized. This fact results in that for the example implementation of the KPI in this thesis, the average receiver xP value uses a large subset of the real pass events and not all of them. As the non-synced passes can be considered to appear roughly randomly, the subset can be considered a random sample which results in that the average receiver xP becomes a Monte-Carlo approximation for this thesis implementation. This does not change that the complete real methodology and theory behind xPlay utilizes in theory all pass events. Furthermore, for this thesis in regards to the time-frame limitation, the Monte-Carlo approximation was the most reasonable approach and still provides very valuable, relevant and interesting results.

Furthermore, the xPlay KPI has not necessarily been evaluated in terms of accuracy although as playability is not distinctly measurable, it's a very hard task evaluate the accuracy of a playability KPI. A possible way would be to use some sort of expert panel who for several games determine if the metric is reasonable in terms of player playability rankings. Although, based on the rankings from the test game it showed realistic results in terms of which type of players were ranked high and low. It would be relevant and also interesting to apply the KPI on a bigger scale for example by using it for all players within a league for a complete season and then analyze average xPlay per game to see which players ranks the highest for an entire league. Furthermore, you could implement a similar player comparison as to how xP and xG KPIs are used by using players' average xPlay values during a season to compare with other players playing in similar positions for the entire league. Further, from this one could conclude which percentile a player is placed in regarding xPlay and this can then be used to provide additional player information for scouting players for recruitment.

All in all, this thesis has been able to answer research question three by providing an exciting suggestion for a new KPI measuring player playability in a quite simple and elegant way. Further, although the KPI should to be evaluated further on a bigger scale, this thesis has shown that the KPI already can provide potentially interesting information for comparing player playability both in terms of overall playability for a game, playability from distinct pitch zones during a game and playability changes during a game.

5.4 Synchronization and Data processing results

The synchronization of the ED and TD data served as the vital first step from which the later feature engineering and statistical learning could be developed from. The process of developing a rigid method for synchronizing this type of data is in itself a project and to explore the absolute best and most accurate way of doing it could be turned into it's own separate thesis. This means that there exists a lot of aspects of the synchronization method and algorithm used within this thesis that could be improved. For example, the algorithm deployed feature weights which were manually tuned and also an aggregation of the feature rankings which could have been done in another perhaps better way. Furthermore, it is very difficult to actually evaluate the accuracy of the synchronization method as it has to be done manually unless there is a a type of ground truth to measure against. Adding to this is that the manual validation is time consuming and therefore it's impossible to thoroughly know exactly how accurate the synchronization is from game to game and there is a real risk that for some reason, some games are not synchronized as well as other games.

For this project the aspect of *good enough* has been key for the synchronization stage of the project. Namely, it just needs to be good enough, not more, not less. As this project is mainly focused on using the TD for statistical learning and comparison,

the time to develop a *perfect* synchronization algorithm and method did not exist. It is because of this that the synchronization just needs to be good enough in terms of providing enough results with enough accuracy so that a fair statistical modeling can be executed.

The first of those aspects is trivial to evaluate by simply looking at the number of synchronized passes that ends up in the final dataset from the 27 games. Furthermore, 19659 passes is a good dataset sample size especially when the dimensionality is not going to be very big. What matters is the size of the dataset produced by the synchronization and not necessarily the proportion of events that are synchronized even though this value is presented in the results. Getting almost 20000 pass events synchronized and forming a dataset for the modeling has to be argued as a success as it is a good dataset size.

Moreover, the other aspect regarding *how accurate* these syncs are is far more difficult to estimate but equally important to the size of the data set. This aspect as mentioned had to be done manually. The method used was simply to view a random subset of around 100 passes per half for each of the 28 games and then calculate what proportion of those passes was synced accurately. The theory is that this monte-carlo approximation should be representative enough throughout the entire game but it must be admitted that it is a rough estimate. As mentioned, an accuracy of at least 90% was estimated which leaves around 10% as potential outliers and noisy passes. But in conversations with supervisors both at the collaboration company PlaymakerAI and from academia, it was established that this was good enough as by using the right statistical learning model for the xP modeling it's easy to circumvent possible effects from noisy data. The success of this aspect is actually probably best measured by looking at the modeling results and as they have shown very clear and reasonable results in terms of much better performance with the added TD information the synchronization accuracy must also regarding this aspect once again be perceived as a success.

5.5 Tracking Data features and Feature Importance

A big section of this project and thesis was spent on deciding and then later implementing methods for extracting specific information from the TD through certain features. The more complex methods were the ones providing information about pressure and defensive lines. By consulting the feature importance of the ED+TD model it seems that the pressure features aided in providing important information and context. Especially the passing lane pressure which gave top importance scores both for the CATBoost built in feature importance but also for the ranking within the SHAP-values. Furthermore, the mutual information and Spearman correlation with the target explored in figure 3.8 also seemed to suggest that the passing lane pressure provided a lot of information about the target.

In contrast, the features derived from the defensive lines did not seem to prove as useful. The specific features describing the distances to the lines seemed to add some value but the binary features for if a defensive line was attempted to be intersected seemingly gave very little information which contradicted the mutual information and Spearman correlation scores in figure 3.8. One possible reasons is that there are other features that describe quite similar information for example the feature that gives a value for the amount of people the pass attempts to pass by. This feature in several cases will correlate with the amount of defensive lines attempted to intersect. If a pass attempts to pass by around 6-8 opponents, most likely at least 2 defensive lines are attempted to be intersected with the pass.

An important aspect to also mention is that for several of the features some assumptions were made which means that there exist possibilities for improvement. To name some key assumptions the biggest one is how to model the added sideline and goalline pressure in the direct pressure feature model. The threshold distance of 10 meters as well as the maximum multiplier of 1.15 were purely arbitrary values developed through manual tuning. Although the implementation of these values gives a quite reasonable theoretical effect on the passer pressure, the values themselves and perhaps even the method of how they are used have no real scientific or statistical background and could definitely be improved.

All in all, the feature engineering of the TD has to be classified as a success both in terms of the overall improvement in predicted probabilities of the xP model by using these features, but also in terms of the feature importance. Seemingly the feature importance evaluation suggests that several TD features actually could be classified as being among the most important features for understanding and predicting pass events.

5.6 The general problems and difficulties with statistical learning for predicting passes

Throughout this thesis it has been declared many times that the non-accurate passes pose several difficulties in different ways. The nature of the non-accurate pass is actually very important to understand when evaluating and building xP models. Non-accurate passes are in many cases simply mistakes. This means that a big section of non-accurate passes look exactly like normal passes but by chance or human error it becomes non-accurate. In fact the amount of passes that on appearance have very low probability of being accurate probably does not make up a majority of the non-accurate passes. Adding to this, football today is better than ever which also means that the average player is better than ever. This in a combination with that modern football has become more and more possession based means that professional footballers today are so good at passing that they frequently complete very difficult passes with seemingly a low probability of being accurate. The point being made is that football is extremely far from being binary in terms of passes and the passing landscape can be quite big and complex. More importantly, the

absolute majority of passes *should* be and are accurate which usually results in that, although not a majority, a big section of non-accurate passes also *should* be accurate. Through a data science perspective, this means that the passing dataset which is unbalanced with a fraction of around 16% non-accurate passes, actually contains even fewer *truly* expected non-accurate passes and that a sizable section of that 16% are outliers or rather noise as they look similar to many simple accurate passes but by human mistake becomes non-accurate.

Therefore it's crucial to have the nature of the data and the task established when developing and also evaluating an xP model. Namely it becomes relevant to for example focus on statistical learning models that deal with outliers and noise very well as the model should not overfit to the non-accurate passes that actually are mistakes. Furthermore, when evaluating the classification performance it's also therefore potentially reasonable to not have a very high recall for the non-accurate passes in comparison with accurate passes as seen in the results. Even a person with football knowledge would struggle to with certainty identify a lot of non-accurate passes therefore it would be unreasonable to expect that from a statistical learning model. Moreover, seemingly the class specific metrics also suggest some overfitting for the non-accurate recall and precision. This can also be argued to be a result of the structure of the non-accurate passes as the models ultimately will be able to predict some non-accurate passes that actually are mistakes if they have been used for training the model. But when the models encounter new non-accurate passes that are mistakes and importantly if the models have truly learned the appearance of accurate passes, it's reasonable that these passes are misclassified which can cause the noteworthy disparity between training and test metrics for the non-accurate passes.

6
Conclusion and Future Work

This chapter presents relevant suggestions for areas of future work in regards to the established work presented within the thesis. Lastly the chapter also summarizes the thesis work in a conclusion section that encapsulates the main points made in the thesis.

6.1 Future work

The limitations, assumptions as well as the results for the the project methodology prompts suggestions for where future work may lie in relation to the established work in this thesis and these suggestions will be outlined in coming subsections.

6.1.1 Thesis Assumptions and Hyperparameters

This thesis has during the execution of the methodology utilized several assumptions, generalizations as well as manually tuned hyperparameters. As mentioned during the discussion chapter the synchronization algorithm used manually tuned weights for the different pass detection features. Further, the aggregation method for ranking the frames is quite trivial and there is a possibility that it can be improved. For example, by possibly utilizing ground truth manually annotated pass event targets in TD files, it would be possible to train a simple statistical learning model like logistic regression to score each frame a probability between 0 and 1 being the correct pass event frame. By doing this the manually tuned weights are exchanged for learned coefficients which most likely more accurately can weight and predict the correct pass frames. Further, the need for an aggregation method disappears as the statistical learning model would also handle that itself in a better way.

Moreover, the thesis also utilized several assumptions and generalizations during the feature engineering. When extracting intended pass teammate, a distance of 8 meters was used to the receiving opposition, further a constant time of 2 seconds was used if the pass lane had to be simulated by the ball trajectory equation. Further, as mentioned in the discussion chapter the direct pass feature model utilized the independent presser assumption, manually tuned multipliers of 1.15 for goalline and sideline pressure as well as a threshold for them of 10 meters. All of these are examples of manually assumed hyperparameters and values. Because of the thesis time frame the strategies employed regarding the implementations of these hyperparameters works well as they provide realistic and good enough results. In

contrast, future work can entail finding more exact and correct hyperparameters that takes the accuracy and realism of the results even further.

6.1.2 Improved TD feature engineering

The last subsection touched briefly on how some of the TD feature engineering could be improved by more accurate hyperparameters. To add to this there exist an interesting possibility of improving the feature engineering as a whole. For example the features developed within this thesis has a strong theoretical purpose in terms of effecting pass outcomes, although the exact execution of extracting those features could very well be very different and perhaps a lot better. Future work could try and find much better ways of capturing direct, indirect and passing lane pressure and perhaps even provide a method to combine all of them into a single feature that can describe the pressure of a pass situation from multiple perspectives in a single value. Further, the defensive line features could potentially be improved as they seemingly did not add that much information as mentioned in the discussion. The TD feature engineering could also be improved in general by adding completely new information that the feature engineering in this thesis completely missed. All in all, there exist genuine possibilities to improve the TD feature engineering in several ways which most likely would improve the xP modeling even further by providing either more accurate information or perhaps new information which would allow a statistical learning model to navigate the football passing landscape with more ease.

6.1.3 xPlay implementations and evaluation

The suggested playability KPI presented in this thesis xPlay has as mentioned in the discussion section been able to provide a very promising metric in terms of describing player playability. Further, the discussion section also outlined the clear need for more exact evaluation as well as implementation on a much bigger scale. Therefore these aspects become very relevant aspects that future work can explore. Future work could first of most likely entail as suggested in the discussion section investigate some type of more direct evaluation regarding the accuracy and trustworthiness of the metric perhaps utilizing some type of manual evaluation from experts. More interestingly and also more doable is implementing the KPI on a big scale in terms of perhaps an entire league or specific tournament and through this further evaluate the behavior of the KPI.

There is also a possibility to further improve the KPI itself through it's methodology and theory. The most obvious aspect that future work can improve is the underlying assumption that all the artificial passes created are equally likely which is used within the law of total probability to simply use the average artificial receiver xP values. This is a slightly naive assumption as not all passes are equally likely, not even in an artificial or hypothetical setting. Therefore future work could perhaps by utilizing an advanced machine learning model as presented by Rahimian.P et al in "Pass Receiver and Outcome Prediction in Soccer Using Temporal Graph Networks", assign hypothetical passes in a pass event with a probability of happening.[27] Fur-

ther, each artificial pass and xP value could then get a probability which could perhaps be used within the law of total probability calculation. This adjustment would provide a more realistic and therefore more accurate value of player playability within a game.

6.2 Conclusions

This thesis has developed and researched spatiotemporal TD from football games in relation to two main objectives. The first objective was to integrate the new TD information with established ED information and improve an xP model. This was executed by developing a synchronization algorithm for the ED and TD, extensive feature engineering of the TD and finally statistical modeling utilizing a gradient boosting statistical learning model. The statistical modeling compared a model only trained on ED with a model trained on ED and TD information. Both models were evaluated on regular classification metrics as well as more in depth probability evaluation looking at calibration, high dimensional decision boundaries, probability outputs for specific passes, type of misclassifications as well as manual evaluation. In all evaluation strategies the models utilizing the added TD information prevailed in seemingly understanding football passes better and making better probability predictions.

The second objective of this thesis was to develop new statistical information regarding player playability using the new TD data. This was achieved by utilizing once again synchronized ED and TD information as well as the newly developed and improved xP model from the first objective. By developing a clear theoretical framework for what playability constitutes, translate this into probability measures and then utilize artificial passes and xP-values, it was possible to estimate the relevant probability measures which in turn gave estimated player xPlay values. The results of suggested applications of the xPlay metric shows very interesting and more importantly realistic outputs in terms of playability rankings as well as temporal and spatial behavior.

In relation to the aim, research questions and time-frame of the thesis the results are viewed with positivity. The integration of the new TD information has proved very successful in improving an established statistical metric in xP as well as allowing the development of a potentially very useful new statistical metric in xPlay. However, as established there still exists several areas which can be improved and further developed and evaluated in a better and more accurate way regarding both the executed objectives of the thesis. Future work regarding finding better and more accurate hyperparameters utilized within the thesis methodology, improving the synchronization process as well as improved or extended TD feature engineering would allow for an even more improved and more realistic xP model. Further, a better synchronization process and a more realistic and likelihood weighted averaging of the artificial receiver xP in combination with thorough evaluation and implementation on a bigger scale would significantly improve the both the accuracy, realism as well as the evaluation of the xPlay KPI.

Bibliography

[1] CatBoost Developers. 'Feature Importance (fstr) — regular feature importance." Available: https://catboost.ai/docs/en/concepts/fstr#fstr__regular-feature-importance. [Accessed: 06-May-2025].

[2] scikit-learn Developers. "sklearn.cluster.KMeans." *scikit-learn v1.6.1 Documentation.* Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accessed: 05-May-2025].

[3] scikit-learn Developers. "Permutation feature importance." *scikit-learn v1.6.1 Documentation.* Available: https://scikit-learn.org/stable/modules/permutation_importance.html. [Accessed: 19-May-2025].

[4] scikit-learn Developers. "sklearn.pipeline.Pipeline." *scikit-learn v1.6.1 Documentation.* Available: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html. [Accessed: 20-May-2025].

[5] SciPy Community. "scipy.integrate.solve_ivp." *SciPy v1.15.2 Manual.* Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html. [Accessed: 05-May-2025].

[6] SciPy Developers. "scipy.stats.permutation_test." *SciPy v1.13.0 Manual.* Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.permutation_test.html. [Accessed: 01-May-2025].

[7] Andrienko, G., Andrienko, N., Budziak, G., *et al.* (2017). "Visual analysis of pressure in football," *Data Mining and Knowledge Discovery*, 31, 1793–1839. doi: 10.1007/s10618-017-0513-2

[8] Anzer, G., & Bauer, P. (2021). "A goal scoring probability model for shots based on synchronized positional and event data in football," *Frontiers in Sports and Active Living*, 3. doi: 10.3389/fspor.2021.624475

[9] Anzer, G., & Bauer, P. (2022). "Expected passes: Determining the difficulty of a pass in football using spatio-temporal data," *Data Mining and Knowledge Discovery*, 36, 295–317. doi: 10.1007/s10618-021-00810-3

[10] Bekkers, J. (2024). "Pressing intensity: An intuitive measure for pressing in soccer," arXiv:2501.04712 [stat.AP]. Available: https://arxiv.org/abs/2501.04712. [Accessed: 05-May-2025].

[11] Bentejac, C., Csorgo, A., & Martinez-Munoz, G. (2019). "A Comparative Analysis of XGBoost." arXiv:1911.01914 [stat.ML]. Available: https://arxiv.org/abs/1911.01914.

[12] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers," in *Proc. 5th Annual Workshop on Computational Learning Theory (COLT '92)*, Pittsburgh, PA, pp. 144–152. ACM, New York, NY. doi: 10.1145/130385.130401

[13] Caron, M. "Understanding positional play," *Medium*, 13 Jan. 2023. Available: `https://medium.com/@markcaron/understanding-positional-play-84b2b6c92e08`. [Accessed: 29-Apr-2025].

[14] Chen, T., & Guestrin, C. (2016). "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD '16)*, pp. 785–794. ACM. doi: 10.1145/2939672.2939785

[15] Coenen, A., & Pearce, A. (n.d.). "A deeper dive into UMAP theory." Google PAIR. Available: `https://pair-code.github.io/understanding-umap/supplement.html`. [Accessed: 06-May-2025].

[16] Coenen, A., & Pearce, A. (n.d.). "Understanding UMAP." Google PAIR. Available: `https://pair-code.github.io/understanding-umap/`. [Accessed: 06-May-2025].

[17] Del Coso, J., Brito de Souza, D., Moreno-Pérez, V., *et al.* (2020). "Influence of players' maximum running speed on the team's ranking position at the end of the Spanish LaLiga," *International Journal of Environmental Research and Public Health*, 17(23), 8815. doi: 10.3390/ijerph17238815

[18] Dick, U., Link, D., & Brefeld, U. (2022). "Who can receive the pass? A computational model for quantifying availability in soccer," *Data Mining and Knowledge Discovery*, 36, 1–28, May 2022. doi: 10.1007/s10618-022-00827-2

[19] Holt, C. A., & Sullivan, S. P. (2023). "Permutation tests for experimental data," *Experimental Economics*, 26(4), 775–812. doi: 10.1007/s10683-023-09799-6

[20] Holmström, M. (2024). *Playing Through the Lines: Machine Learning for Analysing Build-Up Play in Football*. M.Sc. thesis, Dept. Information Technology, Uppsala University, Uppsala, Sweden. Available: `https://www.diva-portal.org/smash/get/diva2:1885312/FULLTEXT01.pdf`.

[21] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning: with Applications in Python*. Cham: Springer. doi: 10.1007/978-3-031-38747-0

[22] Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). "Estimating mutual information," *Physical Review E*, 69(6), 066138. doi: 10.1103/PhysRevE.69.066138

[23] Lundberg, S., & Lee, S.-I. (2017). "A unified approach to interpreting model predictions," arXiv:1705.07874 [cs.AI]. Available: `https://arxiv.org/abs/1705.07874`.

[24] Mann, H. B., & Whitney, D. R. (1947). "On a test of whether one of two random variables is stochastically larger than the other," *Annals of Mathematical Statistics*, 18(1), 50–60. doi: 10.1214/aoms/1177730491

[25] McInnes, L., Healy, J., & Melville, J. (2020). "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," arXiv:1802.03426 [stat.ML]. Available: `https://arxiv.org/abs/1802.03426`.

[26] Niculescu-Mizil, A., & Caruana, R. (2005). "Predicting good probabilities with supervised learning," in *Proc. 22nd International Conference on Machine Learning (ICML 2005)*, pp. 625–632. doi: 10.1145/1102351.1102430

[27] Rahimian, P., Kim, H., Schmid, M., & Toka, L. (2023). "Pass Receiver and Outcome Prediction in Soccer Using Temporal Graph Networks," in *Proc. 10th Workshop on Machine Learning and Data Mining for Sports Analytics (MLSA 2023)*, Turin, Italy, pp. 52–63. doi: 10.1007/978-3-031-53833-9_5

[28] Pifer, N., Wang, Y., Scremin, G., Pitts, B., & Zhang, J. (2018). "Contemporary global football industry: An introduction," in *The Global Football Industry: Marketing Perspectives*, J. J. Zhang & B. Pitts, Eds. London: Routledge, pp. 3–35. ISBN 978-0-8153-6056-8. doi: 10.4324/9781351117982-1

[29] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2019). "CatBoost: Unbiased boosting with categorical features," arXiv:1706.09516 [cs.LG]. Available: `https://arxiv.org/abs/1706.09516`.

[30] Rainio, O., Teuho, J., & Klén, R. (2024). "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, 14, Article 6086. doi: 10.1038/s41598-024-56706-x

[31] Robberechts, P., Van Roy, M., & Davis, J. (2023). "un-xPass: Measuring soccer player creativity," in *Proc. 29th ACM SIGKDD Conf.*, Long Beach, CA, pp. 4768–4777. doi: 10.1145/3580305.3599924

[32] Rufibach, K. (2010). "Use of Brier score to assess binary predictions," *Journal of Clinical Epidemiology*, 63, 938–942. doi: 10.1016/j.jclinepi.2009.11.009

[33] Schober, P., Boer, C., & Schwarte, L. A. (2018). "Correlation coefficients: Appropriate use and interpretation," *Anesthesia & Analgesia*, 126(5), 1763–1768. doi: 10.1213/ANE.0000000000002864

[34] Sharpe, C. (2024). "xPass 360: Upgrading expected pass models." *Hudl StatsBomb*, 13 Aug 2024. Available: `https://statsbomb.com/articles/soccer/xpass-360-upgrading-expected-pass-xpass-models/`. [Accessed: 05-May-2025].

[35] Spearman, W., Basye, A., Dick, G., Hotovy, R., & Pop, P. (2017). "Physics-based modeling of pass probabilities in soccer," in *Proc. MIT Sloan Sports Analytics Conf.*, Boston, MA.

[36] StatsBomb. (n.d.). "Expected goals (xG) explained." *Hudl StatsBomb*. Available: `https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/`. [Accessed: 21-Apr-2025].

[37] Van Roy, M., Cascioli, L., & Davis, J. (2023). "ETSY: A rule-based approach to event and tracking data synchronization," in *Proc. MLSA Workshop, ECML-PKDD 2023*. Cham: Springer, pp. 11–23. doi: 10.1007/978-3-031-53833-9_2

[38] Zwillinger, D., & Kokoska, S. (2000). *CRC Standard Probability and Statistics Tables and Formulae*. Boca Raton, FL: CRC Press. ISBN 1-58488-059-7.

A

Appendix

Listing A.1: Truncated row from Tracking data JSONL file


```
{
    "frame_id": 53591,
    "time": "35:38.440000",
    "period": "half_1",
    "pitch_side": {
        "left": "Real␣Madrid",
        "right": "Borussia␣Dortmund"
    },
    "ball": {
        "x": 82.31334767713055,
        "y": 12.251893193672732,
        "z": 0.0,
        "x_velocity": -4.948690945082319,
        "y_velocity": -4.493107365484192,
        "z_velocity": 0.0,
        "x_acceleration": 1.7203626335415834,
        "y_acceleration": 1.561983584300819,
        "z_acceleration": 0.0,
        "is_imputed": false,
        "speed": 6.6841271582539505,
        "acceleration": 2.323669578169703
    },
    "players": [
        {
            "person_id": 0,
            "person_name": "Thibaut␣Courtois",
            "team_id": 3468,
            "team_name": "Real␣Madrid",
            "jersey_number": 1,
            "category_name": "goalkeeper",
            "x": 4.404935062535536,
            "y": 32.66175931388771,
            "x_velocity": -9.37312022151443e-05,
            "y_velocity": -4.991654465223566e-05,
            "x_acceleration": 0.00233399274881676,
            "y_acceleration": 0.001242967661704883,
            "speed": 0.00010619416037953738,
            "acceleration": 0.002644331817222135,
            "head_angle": NaN,
            "shoulders_angle": NaN,
            "hips_angle": NaN,
            "is_imputed": true
```

```
        }
        // Remaining players would follow here
    ],
    "referees": [
        {
            "person_id": 42,
            "person_name": "Istvan Kovacs",
            "category_name": "referee",
            "x": 69.65171108538637,
            "y": 29.99022625972133,
            "x_velocity": -3.425706871531177,
            "y_velocity": -0.5990786936034375,
            "x_acceleration": 0.16664037634019985,
            "y_acceleration": 0.28217299174185284,
            "speed": 3.4776950485609754,
            "acceleration": 0.3277050690717969,
            "head_angle": NaN,
            "shoulders_angle": NaN,
            "hips_angle": NaN,
            "is_imputed": true
        }
    ]
}
```

Listing A.2: Example row from Event Data csv file

```
{
    "game_time": 1.0,
    "start_time": 0.0,
    "player_name": "Serhou Yadaly Guirassy",
    "team_name": "Borussia Dortmund",
    "action": "Passes accurate",
    "xpos": 49.8,
    "ypos": 49.90,
    "xdest": 41.4,
    "ydest": 47.20,
    "distance": 9.00,
    "progression": 0.0,
    "side_distance": 1.83,
    "action_order": 3.9,
    "corner": false,
    "distance_to_goal": 52.71,
    "angle_to_goal": 0.076,
    "end_time": 1.0,
    "foot": null,
    "freekick": false,
    "game_state": 0,
    "goal": false,
    "goal_mouth": null,
    "header": false,
    "home": "Real Madrid CF",
    "opponent": false,
    "penalty": false,
    "period_id": 1,
    "receiver_name": "Marcel Sabitzer",
    "set_piece": false,
```

```
    "shot": false,
    "single_event_id": 1,
    "throw_in": false,
    "xt": 0.0,
    "xt_in_play": -0.0013,
    "xt_prevented": -0.0013,
    "chain_id": "317539056868528-1",
    "xt_chain": -0.0019,
    "chain_duration": 46.0,
    "chain_team": "Borussia␣Dortmund",
    "chain_xpos": 49.8,
    "chain_ypos": 49.90,
    "chain_start_xpos": 49.8,
    "chain_start_ypos": 49.90,
    "chain_end_xpos": 41.2,
    "chain_end_ypos": 54.90,
    "chain_shot": false,
    "chain_goal": false,
    "attack_type": "Open␣Play",
    "xp": 0.8877,
    "xg": null,
    "xg_chain": null,
    "game": 317539056868528,
    "value": 0.0057,
    "home_formation": "4-3-3",
    "away_formation": "4-2-3-1"
}
```

Table A.1: Final enriched dataset structure

Feature	Type	Source	Description
action (target)	Binary	ED	Successful vs. unsuccessful pass.
xpos	Numerical	ED	x-coordinate of pass.
ypos	Numerical	ED	y-coordinate of pass.
xdest	Numerical	ED	x-coordinate of actual pass destination.
ydest	Numerical	ED	y-coordinate of actual pass destination.
dist	Numerical	ED	Total distance of pass.
prog	Numerical	ED	Distance of pass along x-axis.
side_dist	Numerical	ED	Distance of pass along y-axis.
corner	Binary	ED	Pass from a corner.
distance_to_goal	Numerical	ED	Distance to opposition goal.
angle_to_goal	Numerical	ED	Angle to opposition goal.
freekick	Binary	ED	Pass from a freekick.
header	Binary	ED	Pass from a header.
set_piece	Binary	ED	Pass from a set piece.
throw_in	Binary	ED	Pass from a throw-in.
direct_press	Numerical	TD	Direct pressure value (see Sec. 3.1.3.2).
indirect_press	Numerical	TD	Indirect angular pressure (see Sec. 3.1.3.3).
press_passing_lane	Numerical	TD	Passing lane pressure (see Sec. 3.1.3.4).
attempt_breaking_first_line	Binary	TD	Intersects first defensive line (see Sec. 3.1.3.5).
attempt_breaking_second_line	Binary	TD	Intersects second defensive line (see Sec. 3.1.3.5).
attempt_breaking_third_line	Binary	TD	Intersects third defensive line (see Sec. 3.1.3.5).
distance_first_line	Numerical	TD	Dist. to first defensive line (see Sec. 3.1.3.5).
distance_second_line	Numerical	TD	Dist. to second defensive line (see Sec. 3.1.3.5).
distance_third_line	Numerical	TD	Dist. to third defensive line (see Sec. 3.1.3.5).
bypassed_opponents	Ordinal	TD	# opponents between passer and receiver (see Sec. 3.1.3.6).
smallest_distance_passing_lane	Numerical	TD	Min. opponent-to-lane distance (see Sec. 3.1.3.6).
angle_between_defender_v_pass	Numerical	TD	Min. angle defender-to-lane (see Sec. 3.1.3.6).
smallest_distance_passer	Numerical	TD	Min. opponent distance to pass location.
passer_speed	Numerical	TD	Passer speed at moment of pass.
height_of_pass	Numerical	TD	Ball height during pass.
speed_of_pass	Numerical	TD	Ball speed during pass.
ball_height_at_pass	Numerical	TD	Ball height at pass moment.
distance_sideline	Numerical	TD	Dist. to nearest sideline.
distance_goalline	Numerical	TD	Dist. to nearest goalline.
nr_opposition_closer_to_passer	Ordinal	TD	# opponents closer to pass loc. than own goal.

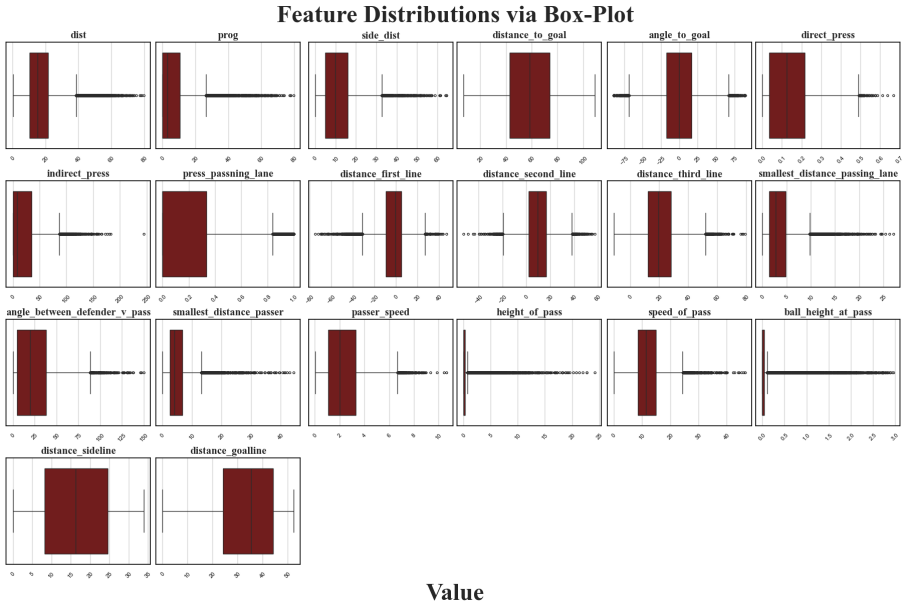

Figure A.1: Distributions of numerical features in final dataset visualized using boxplots.

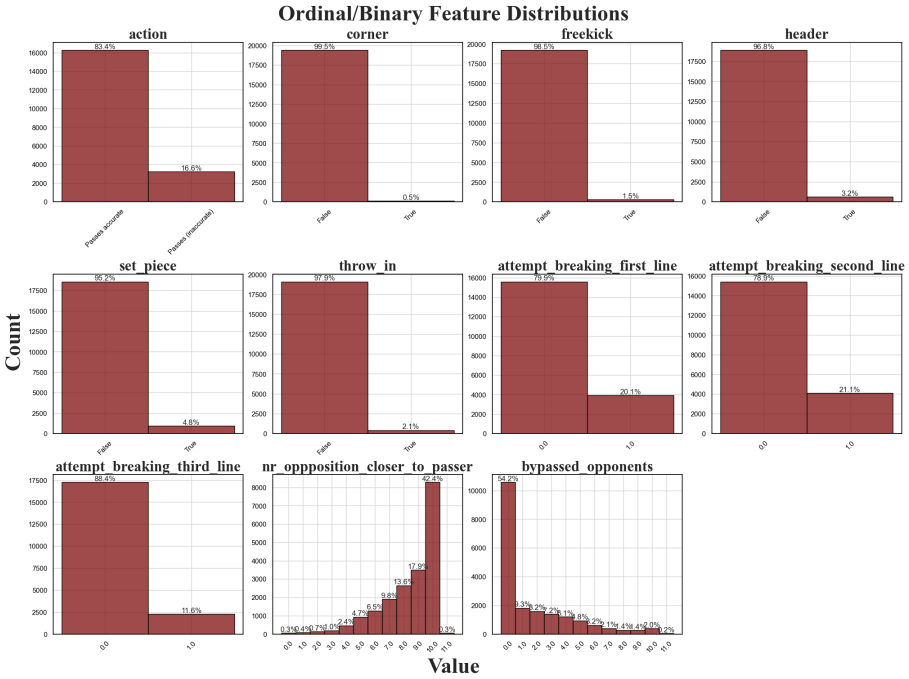

Figure A.2: Distributions of categorical and binary features in final dataset visualized using histograms.

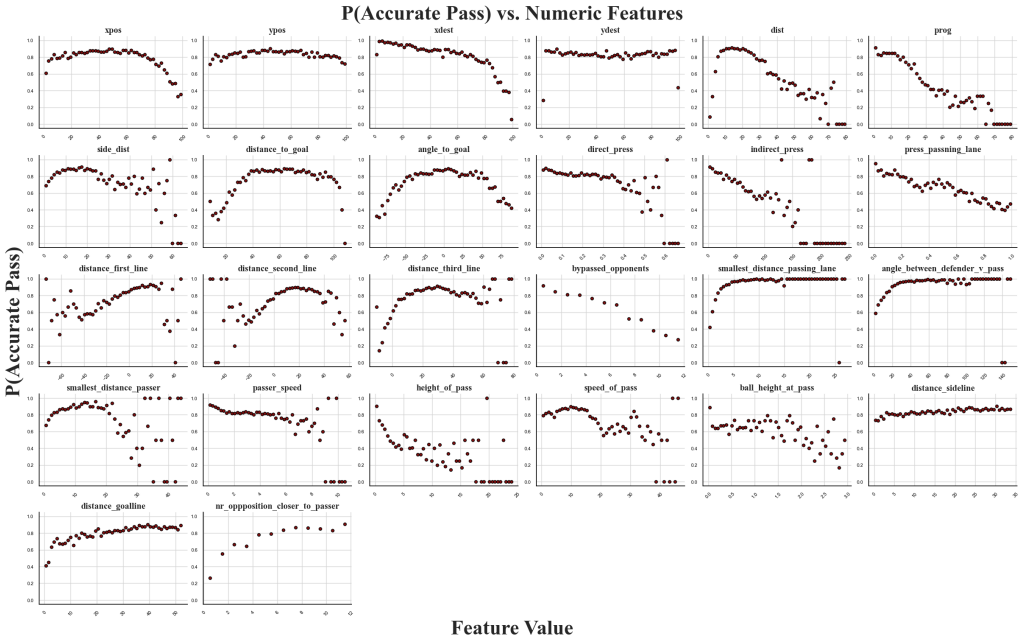

Figure A.3: Change in pass probability based on numerical feature values.

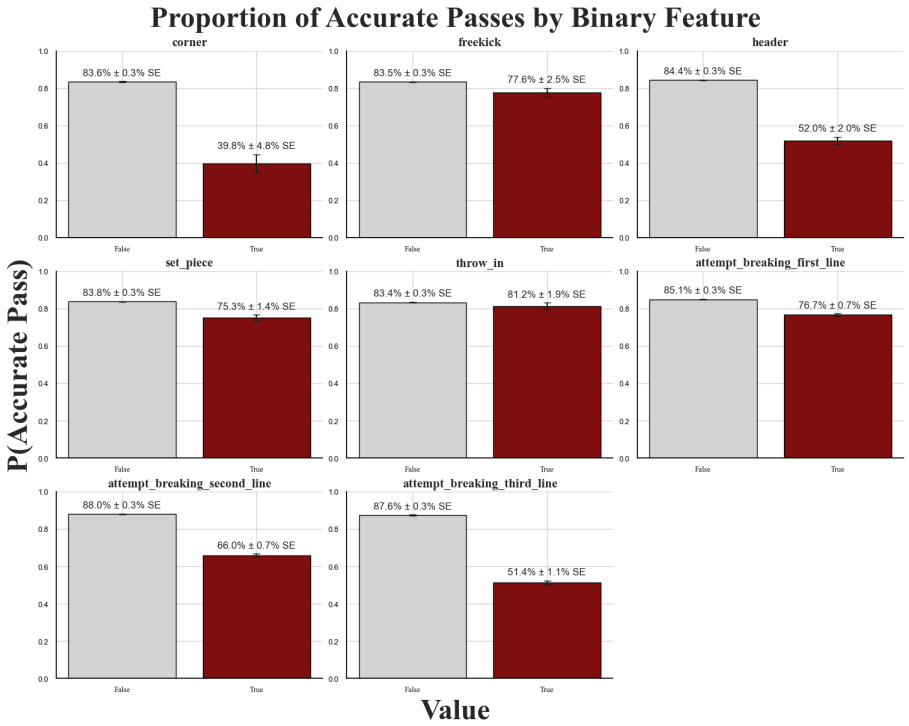

Figure A.4: Change in pass probability based on binary feature values.

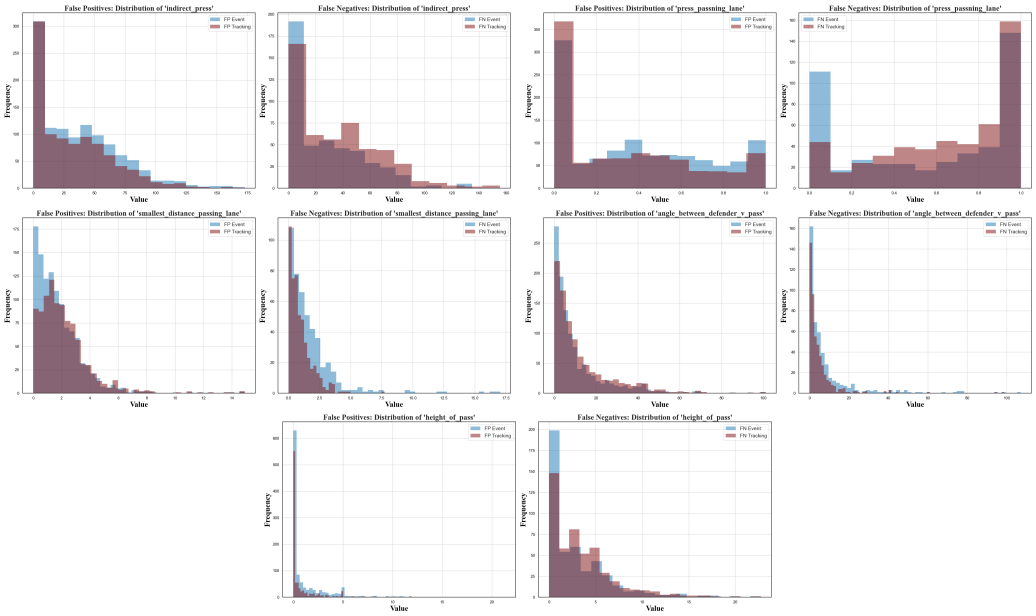

Figure A.5: All subplots show feature distributions among the false positive and false negative classifications of the ED-only models and the ED+TD models. Features in the top row in order from left to right: "indirect_press" - "press_passing_lane". Features in the middle row in order from left to right: "smallest_distance_passing_lane" - "angle_between_defender_v_pass". Feature in bottom row: "height_of_pass".

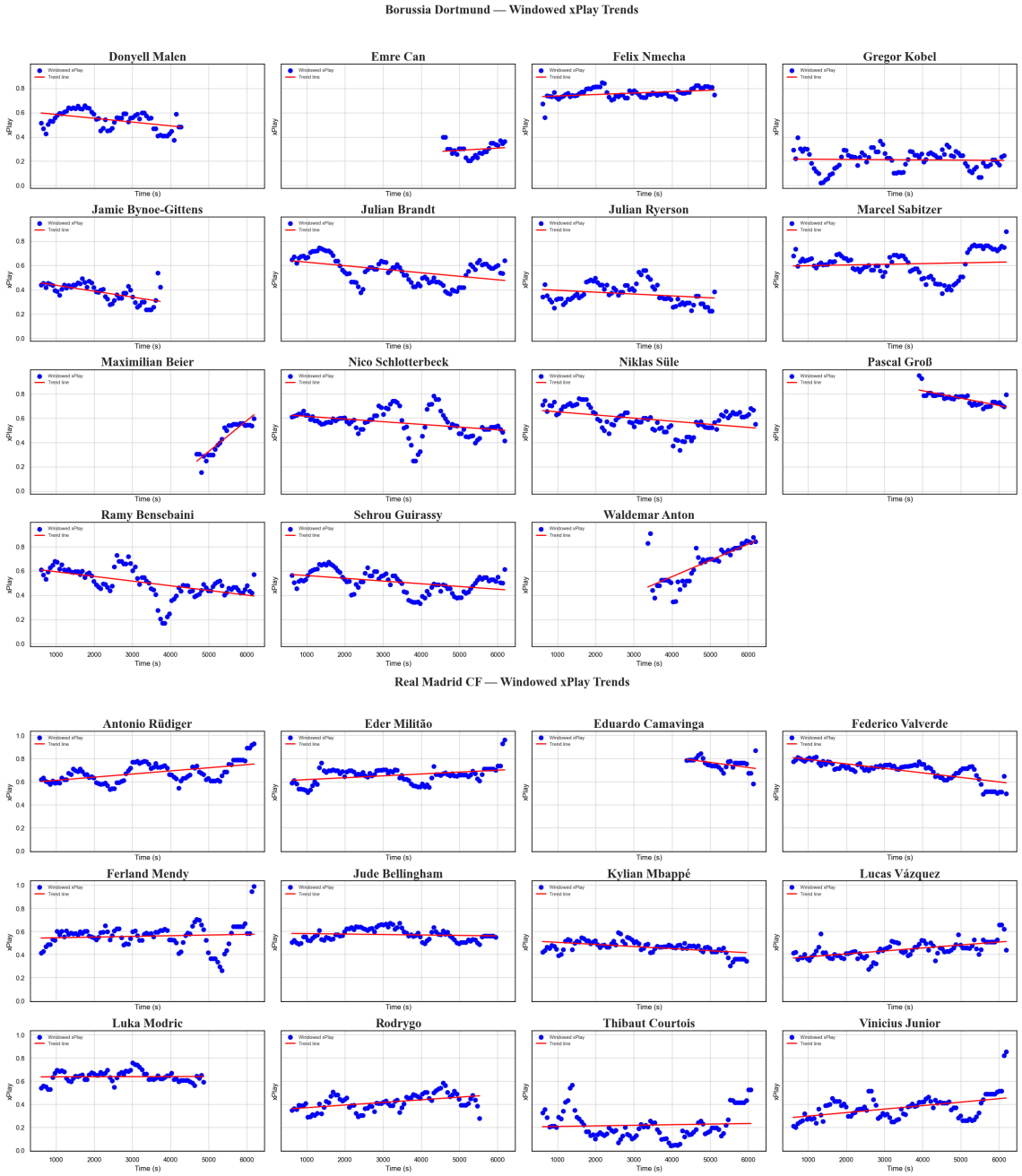

Figure A.6: Player xPlay values within a 10 minute moving time window throughout the entire match. First subplot is for the Borussia Dortmund players while the second subplot is for the Real Madrid players.

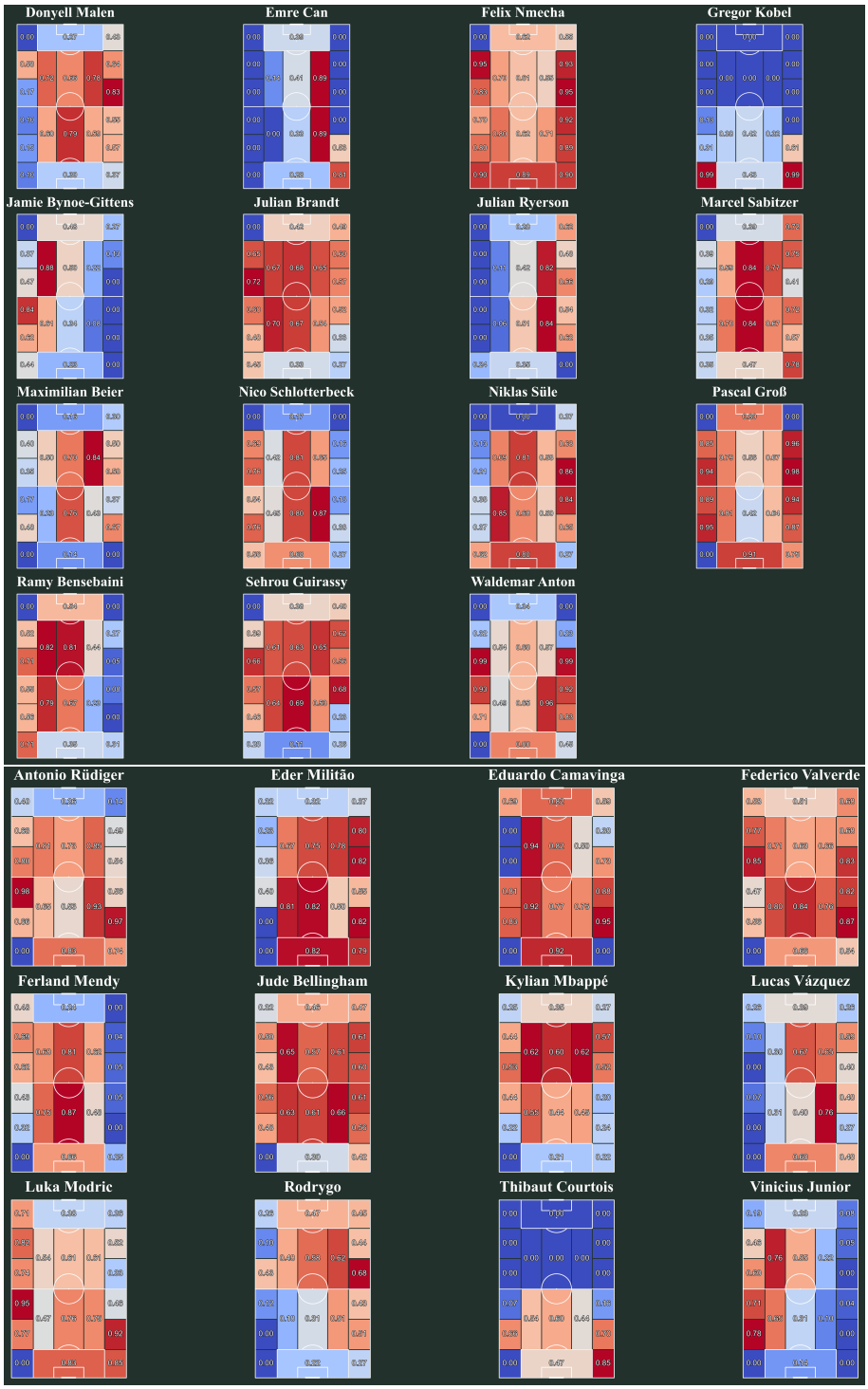

Figure A.7: Player xPlay values depending on the "Juega de Pocision" zones. First subplot is for the Borussia Dortmund players while the second subplot is for the Real Madrid players.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY