UTRECHT UNIVERSITY

Department of Information and Computing Science

Applied Data Science master thesis

Pitch Control in Football

A Machine Learning Approach for Area Allocation

July 26, 2024

Abstract

The analysis of pitch control in football can be a very insightful practice for coaches. A lot of literature incorporates the concept of pitch control, but it is often embedded within larger models. This makes it difficult to get insights on pitch control itself. Literature that does use a separate model for pitch control use methods that are simplistic or based on assumptions. This study, therefore, aims to create a more accurate, yet insightful model for pitch control, using machine learning. The pitch control models are assessed on how well they predict who will receive a pass and how interpretable the visualisation of the pitch control is. This work shows that machine learning models achieve higher accuracies on the pass recipient predictions. The visualisations are, however, not well interpretable. This shows that predicting the recipient of passes is a difficult way to asses pitch control. In future research, when passes are again used, a broad dataset with greatly varying types of passes should be used.

Contents

1. Introduction

Match analysis in football has long been focused on discrete, on-ball events like number of shots, completed dribbles and possession percentages. However, the performance of a team also highly depends on off-ball actions and positioning. With the introduction of acquiring positional tracking data, more detailed insights in team and individual performances can be obtained. As a result, many models have been created which aim to quantify many different aspects of the sport.

One of the focus areas in football analytics, is quantifying how well players or teams position themselves on the pitch. For a game like football, this is a rather abstract idea. Therefore, existing literature have used many different, yet similar definitions and names for this. One way to do this is by allocating space on the pitch to players or teams, with the idea that the more space in control, the better. Taki et al. (1996) introduced this concept by allocating an area to each player. This area, which they named a 'dominant region', is based on the player being able to reach any point in this area before any other player according to movement models.

Taki and Hasegawa (2000) return to the dominant regions with a new model that is essentially a Voronoi model (Berg et al., 2008) with an adapted distance function. This new distance function utilises individual acceleration models and movement vectors of the players. If both of these were 0, this would be a regular Vonoroi model.

Martens et al. (2021) point out that the advantage of using dominant regions is that they can easily be visualised and therefore are easy to interpret. They also mention that interpretability is crucial in football as it allows non-technical staff like coaches to turn data-driven results into actionable insights. For this reason, many research on higher level subjects such as the evaluation of passes or creation of offensive potential, have used dominance regions as a basis (Chawla et al., 2017; Brefeld et al., 2019).

The previous approaches allocated each location on the pitch to a player. Another approach is assessing the degree of control each player or team has on each point on the pitch. Fernández and Bornn (2018) and Spearman (2018) use models that predict these values for what they call 'pitch control'. They make use of a probability density function of a bivariate Gaussian distribution. The results can be visualised in a very comprehensible way. However, their method uses some assumptions and is very rigid.

Martens et al. (2021) tried to overcome these problems by introducing machine learning methods. They, however, mainly focus their work on improving the complete model

of Fernández and Bornn (2018), of which the pitch control model is only a basis layer. Their results are positive but they mention that when using their linear vector support machine (SVM) based model for pitch control, it does not show any improvements compared to the model of Fernández and Bornn (2018).

Xie et al. (2020) and K. et al. (2024) show that other research that uses spatio temporal data often uses machine learning models like decision trees, SVM's, random forests (RFs) or artificial neural networks (ANNs or NNs). They also found that decision trees usually perform wel and different types of NNs often perform best. Ai et al. (2021) also get the best results with NNs in their comparative study, confirming the high performance of NNs in models using spatiotemporal data in basketball. Lin et al. (2024) and Agyeman et al. (2019) also work with NNs on spatiotemporal data in basketball and football respectively. They both work with data that includes images, which is why they use a convolution neural network (CNN). They also work with sequences of data and argue that a recurrent network features, like short long short-term memory (LSTM), work well in these situations.

As the literature shows, previous works often makes use of pitch control or a similar concept as the basis of their models. More recent research, on higher level concepts, often combine pitch control with other concepts within one machine learning model. This combination makes it difficult to evaluate the implementation of pitch control, to visualise it and to understand the performance of the players and the team as a whole.

Literature that uses an explicitly separable part of their model for pitch control is often older, does not use any machine learning, the actual performance is not well known and it is only focused on who can get to a location first. These models disregard any risks and missed opportunities of positioning poorly on a team level. Machine learning models might be able to learn these more abstract situations and predict the decisions of players to go for the ball or not.

Therefore, the goal of this study is to find out if the implementation of machine learning models are of added value for the performance of models that focus explicitly on pitch control. The research question that follows from this is:

How can the implementation of machine learning improve the performance of pitch control models in football analytics while maintaining interpretability?

To answer this question, several models will be developed and evaluated using a dataset containing spatiotemporal data of players on a pitch. In the remaining part of this study, first this data will be discussed. After this, the methods and models that are used will be described. Then the results will be presented. Finally, the study will conclude with a summary of findings and a discussion.

2. Data

In this chapter, first a description of all the used data will be given. After this, the processing steps will be explained. There are many processing steps and they differ from data ordering to detailed processing steps. Therefore, first some global processing steps will be discussed, and then the more detailed steps.

2.1 Description of the data

The data used for this research is provided by Forward Football, a company specialised in sport technology and data-analysis in football. Forward Football acquired this data with a local positioning measurement (LPM) system made by GenGee.

Data on 49 matches are provided. Two of these matches are not consistent in number of players and pitch size as they were test data. The data on the 47 matches consists of three parts per match. The first is the preprocessed data on the passes, which has been derived by Forward Football from the raw spatiotemporal data acquired with the LPM system. The pass data is stored in a csv file per match. The csv files are slightly inconsistent and the data ranges from 170 to 219 columns. Collectively they have 280 different columns. These large amounts of data are sometimes difficult to comprehend and largely redundant for this study. Furthermore, the data lacks potential valuable information like the speed of players. The combination of these reason led to the choice to take only part of the pass data as a basis of the preprocessing steps. The other two parts of the data are used to fill in any gaps that are left behind.

The second part of the used data is the informational data on the matches. This consists of two excel files per match, one for each team playing and contains information like the start-time of the match, the final score and the players in the team.

The third part of the data is the raw spatiotemporal data of the players and the ball during the matches. For each match the data is in a sheet in another excel file. The other sheets in this files are not used. The sheet consists of one column for the time stamp and then 3 columns for the ball and each player that was at some point on the pitch during the match. Of the three columns per player, one contains the ID and the other two the x- and y-coordinates, and thus no z-coordinate. The origin of this coordinate system is at a corner of the pitch. The timestamps are 0,2 seconds apart, meaning the frequency of the data collection was 5 hertz.

The matches are short practice matches. The total time of all the matches is 3355

minutes, which is the equivalent of the duration of 37,28 actual matches. However, for most matches, the trackers were turned on several minutes before the match and were not turned of during half time, meaning there is fewer minutes of actual play time.

Furthermore, turning the trackers on and off, for example during a substitution of a player, was done manually and was not always done correctly. This resulted in players on the pitch that were not being tracked, and players on the bench that were being tracked. In a match one would expect 11 players a side and thus 22 players being tracked (not taking into account any time penalties from yellow cards and players being sent off with red cards).

The percentage of time with 22 players on the pitch that were being tracked is at most 86,5% for a match, while some matches even have 0%. This percentage is only an estimation, as it can happen that a tracker of a player on the pitch is turned of while a player off the pitch is turned on, also resulting in 22 players. The resulting total time of play with 22 players on the pitch is 811 minutes, which is the equivalent of the duration of 9,01 matches. In Figure 2.1 the distribution of time per number of players can be seen. Most of the time there were only 21 players being tracked.

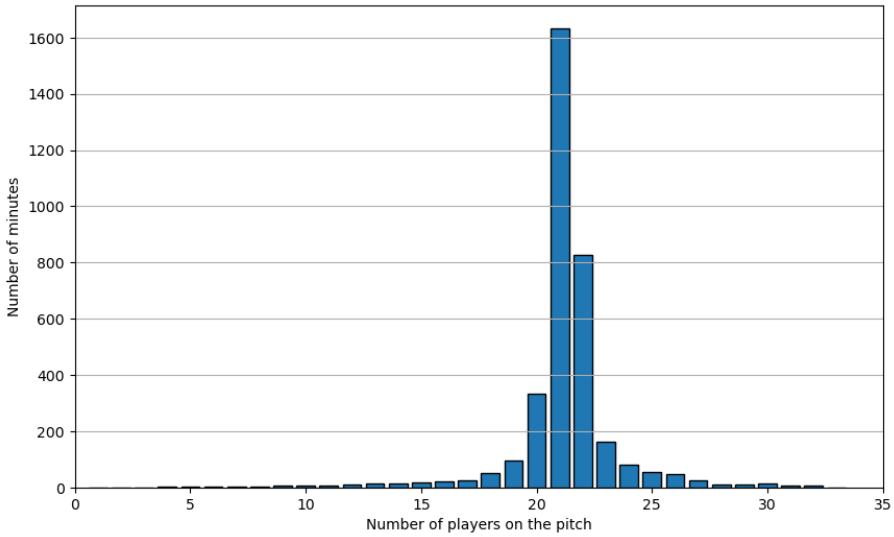


Figure 2.1: The number of minutes per number of tracked players

The hardware used for collecting the data is also not flawless. While the frequency of the data collection is 5 hertz, the collection does not always succeed. Sometimes the fault lies at the collector side, resulting in missing rows of positional data and sometimes the fault is on the tracker side, resulting in missing data on one player. The amount of missing positional data where it should have been collected lies around 3,5%. The hardware was furthermore not consistent in the precision of data collection. In only 12 of the matches did the hardware record with a precision of 0.01 metre, in the other matches the precision was only 1 metre.

The The age groups of the teams range between under 12 years old and adults. Play styles differ between age groups as strength increases over the years and so does the sense for tactics. This leads to inconsistencies in the data on a deep level. This is something that might impact analysis results, but is too difficult to overcome, especially with the limited size of the dataset.

A detailed list of information on the matches can be seen in Appendix A.1.

2.2 Preparation of the data

The preparations of the data had many steps. The steps taken can be explained in multiple levels of detail. The detailed steps are important to understand, but before going into these steps, a more global overview will be given with the aim of understanding why every step is taken.

2.2.1 Global data preparation

As mentioned before, in the raw data, there is columns on all players that at one point were on the pitch. As not every player was always on the pitch and thus not always being tracked, there is a lot of empty cells. This is a lot of redundant memory and a lot of columns, which can result in a lack of comprehensibility. This, combined with the a lack of understanding of the data on the passes, resulted in the choice to completely restructure the data.

The first step that is taken is creating several dataframes, each with their own purpose. This was done with the pandas package in python and saved as csv files for repeated use. With these clear dataframes the input for the different models is eventually created. In Figure 2.2 the complete process can be seen. The grey boxes represent the provided data and the two grey boxes at the top are the raw data. The first row of white boxes represent the newly created dataframes. The last step is to use these data frames to prepare the input for the different models, each model needs its own input structure and information.

2.2.1.1 Restructuring of the data

As seen in Figure 2.2, four dataframes are made. The dataframes are structured and interconnected like tables in a database. Two of the dataframes are made using the general match data and two of the dataframes are made using the spatiotemporal data.

The two dataframes made with the general match data are the dataframe with information on the players and the dataframe with information on the matches. The dataframe of the players connects team ID's with player ID and match ID combinations. The combination is needed as some players play for different teams during different matches and even play against the team they have played for. The matches dataframe connects the

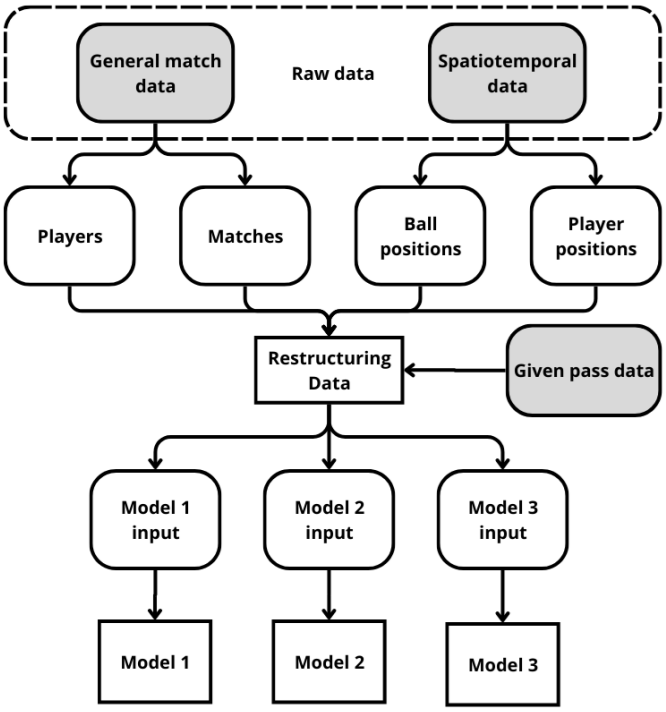


Figure 2.2: The global process of restructuring and preparing the data for the models

match ID's with general data of the match. This means the team ID's of the teams playing and the pitch measurements.

The two dataframes made with the spatiotemporal data are a dataframe containing the positions per timestamp of the ball for each match and a dataframe containing the positions of the players for each timestamp of each match.

2.2.2 Detailed data preparation

In this section detailed preprocessing steps are explained that were used during the process of creating the new dataframes that contain the required data for the models.

2.2.2.1 Interpolation of coordinates

As mentioned, there is a lot of missing positional data. The missing data can result in nan-values in data that is derived from the positional data. For example, when calculating the speed of a player, you need positional data on consecutive timestamps. To solve this problem, interpolation is done on the location of missing data in the positional data sheet. The positional data is two dimensions per timestamp and each dimension can be interpolated separately. Therefore, linear interpolation suffices (G. Zhang, 2023). The builtin interpolate function of the pandas package in python is used for this.

When multiple timestamps after each other are missing, a choice has to be made if

interpolation is done. When data is missing for multiple timestamps and eventually is recorded again this can be because of multiple reasons. In the case of a substitution interpolation is not wanted. When it is because of recording problems, the period is usually short and interpolated data is wanted. For this reason, it was chosen to introduce a threshold for the length of a period for which the data is missing. If the period is longer than the threshold, the data is not interpolated, if it is shorter, the data will be interpolated. If the threshold is too long, the created data gets less value because it will be imprecise. Therefore, a threshold of 10 seconds is chosen, balancing the value and availability of data.

2.2.2.2 Removing Erroneous Data

Even when there is tracked data on certain timestamps, this does not always mean the data is actually following a player or should even be there. There are instances of players losing their tracker and instances of forgetting to turn off the tracker. This data can be partially found and should be filtered out.

When losing the tracker, the data will not show any movement. To find this data, a search for periods with movements smaller than 0.3 meters in both directions was done (0.3 instead of 0 to make up for noise and measurement errors). If the period was longer than a threshold of 30 seconds, the data was removed. The threshold is still quite high in order to reduce the risk of removing valid data.

To remove instances of forgetting to remove a tracker, a search is done for periods that the tracker is off the pitch. In this case, the length of that period is 20 seconds. This slightly more lenient length is because if someone is not on the pitch for such a period, it may not necessarily mean they are substituted.

Table 2.1: Results of removing erroneous data (each datapoint is an x- and y- coordinate for one player on one timestamp).

	Nr. of datapoints	% of total
Complete dataset	22.160.713	100
Removed for standing still	4.158	0,02
Removed for being off pitch	1.231.800	5,56

In Table 2.1 an overview of the removal can be seen. A total of 5,58% of the data points has been removed. In Figure 2.3 the new distribution of number of minutes per number of players can be seen. When comparing this with Figure 2.1, it can be seen that from 19 players on the pitch and up, the number of minutes has been reduced. The number of minutes for numbers of players below 19 have gone up, especially around 13 players. This indicates that when there is erroneous data for a timestamp, the chance is big that multiple data points of that timestamp are erroneous.

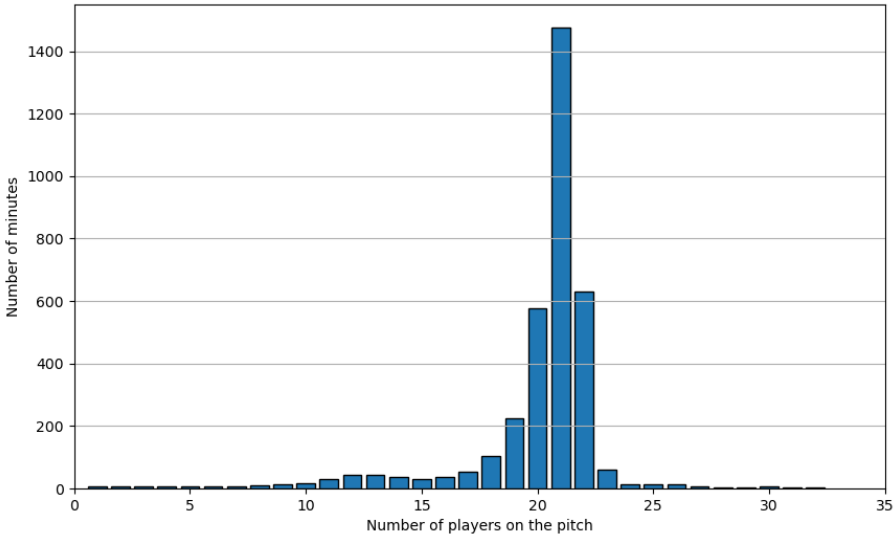


Figure 2.3: The number of minutes per number of tracked players

2.2.2.3 Smoothing of coordinates

Tracking systems are prone to noise and inaccuracies due to environmental conditions, signal interference, tracker quality and tracker calibration (Perš et al., 2002). The tracking data also shows sudden jumps or outliers, that are physically not possible. Furthermore, most of the matches use metres as precision of the measurement, which is rather imprecise as can be seen in Figure B.2.

Smoothing makes sure the data represents the actual movements of the players more accurately, while reducing outliers. This improved accuracy is crucial for parts of the analysis, such as calculations of speeds and distances, and thus increases the reliability of the predictive models.

In previous research with comparable data, different kinds of smoothing techniques have been used. Hen et al. (2004) shows that locally weighted scatterplot smoothing (LOWESS) and the Local Polynomials (LP) method outperform the most common smoothing technique Moving Average (MA). However, these methods require dense datasets for good results and MA still performs well. The 5 Hertz data available might not be dense enough. More recent literature often makes use of a Kalman filter, which is particularly good at eliminating outliers (Najeeb & Ghani, 2020). This method, however, needs information about on errors and noise in advance, information that is not available for this data. Perš et al. (2002) mentions that the obvious choice for similar data is using a Gaussian smoothing kernel and their results show good performance.

This information, together with the fact that both MA and a Gaussian smoothing kernel have a low computational load, makes the choice to compare MA and Gaussian smoothing and pick the best one. For MA a rolling window in python is used. For the

Gaussian smoothing the 'gaussian_filter1d' of the scipy.ndimage package is used.

While smoothing can have positive effects on data analysis, over-smoothing can be problematic. In football it can diminish the abrupt movements that are a part of the game. Furthermore, the impact of smoothing can vary between the two groups of matches, with tracking precision differing between metres and centimetres. Therefore, different levels of smoothing are evaluated and they are evaluated on the two groups of matches separately.

The differences between the different levels of smoothing are not only caused by the difference between MA smoothing and Gaussian smoothing. They are also caused by the settings of the window size for MA and the σ value for Gaussian smoothing. Both values cause more intense smoothing when increased.

The evaluation of the smoothing methods is done by looking at the number of passes that during analysis would be affected to different degrees. Furthermore, player trajectories are visually evaluated to see if they are well smoothed or maybe over-smoothed.

The in depth analysis of the performances of the different smoothing methods can be found in Appendix B.1. Based on the analysis a choice has been made to apply MA smoothing with a window of 3 on tracking data with a 0,01 metre precision, and Gaussian smoothing with $\sigma = 0,9$ for tracking data with 1 metre precision.

2.2.2.4 Adjusting play directions

Finding patterns in consistently structured data is easier than in unstructured data. In case of the passes it would help if the direction of play is consistent. This way the model can, for example, find out easier if a pass is forward. The direction of play is changed for each pass, such that the goal on $x = 0$ is the goal the initiator of the pass is defending.

There is little consistency in the data regarding direction of play. Some matches there is no side change, while other matches there are up to two side changes. When there is a side change, this is not necessarily at the half of the total duration of the match. Because of this inconsistency, a specific method had to be created. An in depth explanation of this method can be found in Appendix B.2. This method is used to adjust passes to a consistent play direction.

2.2.2.5 Filtering relevant passes

To optimise the performance of the ML models, the training data should be as consistent as possible. In the pass data there are some inconsistencies, therefore some filtering should be done. In Table 2.2 the number of passes after each filtering step and the success rate of these passes can be seen. The success rate is the number of passes that end up with a team mate of the passer divided by the total number of passes.

Table 2.2: The number of passes and the success rate of these passes after each filtering step.

	Nr. of passes	% of total	% succesfull
All passes	32.404	100	70,6
Removed for missing receiver data	6.532	20,2	88,1
Removed for other reasons	725	2,23	93,9
Remaining passes	25.147	77,6	88,1
Remaining passes, 11 a side	5.574	17,2	87,6

As can be seen in Table 2.2, there are 32.404 recorded passes. The success rate of these passes is 70,6%. This success rate is not representative as there are passes of which the receiver is unknown. The positional data of the actual receiver is missing and can therefore not be used in evaluating any models. Then there are passes that are removed because they seem to be either duplicates or during the warming up.

25.705 passes remain after the filtering filtering steps. The consistency of the remaining data is still not perfect as there is a lot of missing positional data and sometimes redundant extra data. For the ML models a trade-off can occur between the amount of data and the consistency of the data. This is why the passes are further filtered down to only the passes with 11 players per side. Both datasets can be used for training and evaluation of the models.

3. Method

The goal of this study is to use machine learning models to define pitch control or dominance areas. To enable evaluating the model well and in a clear manner, a clear definition is needed. The clearest definition from the literature is that for each point within the area assigned to a team or player, that team or player is able to reach the point before the other team or any other player (Taki et al., 1996).

However, while on the pitch, a player cannot be sure he is the first that reaches the ball, nor is it always the best tactical decision to go after the ball. Machine learning techniques might predict these decisions and thus a more suitable definition is the following: 'The region on the pitch where a player or team is most likely to reach and control the ball if it were to arrive there.' This definition implies that both positional advantage and tactical decision-making are important for the control of an area.

To accurately predict movements and actions on the pitch that are influenced by these implications, a data-driven approach is needed. This data should have positional information on both the ball and the players. Furthermore, it should have information of the reaction of the players on the movement of the ball and each other.

The combination of positional data of the players and pass data is a viable option for this. A pass provides a clear instance of the movement of the ball into a different position and players reacting to this movement. There is also information on who receives the pass after this movement of the pass. This indicates that the receiver is a player with control of the location of the reception of the pass at the moment when the pass is given.

This data can, therefore, be used to train machine learning models, which then can identify patterns and predict which player or team is most likely to reach and control the ball at different locations on the pitch. The evaluation of these models will focus on their accuracy in predicting actual pass outcomes and their effectiveness in visualisation of game scenarios. With this the models capability of understanding tactical decision making and positioning on the pitch the model can predict outcomes for any point on the pitch if the ball would move there. Because of this, the models are not only useful for the prediction of the abstract concept of pitch control, but can also offer insights for coaches.

In the remaining part of this chapter the different methods will be discussed. First two baseline methods will be chosen and discussed. After that the newly created machine learning models will be presented.

3.1 Baseline methods

To evaluate any newly created pitch control models, a comparison with an existing model can give insights on the relative performance. To make this baseline more representative, it is chosen to take two models as baseline model. A simple Voronoi diagram method (Berg et al., 2008) and the method used by Fernández and Bornn (2018). This way both the approaches solely based on physical proximity of individual players and the pitch control method based on physical proximity of the whole team is represented through the Voronoi method and the method of Fernández and Bornn (2018).

3.1.1 Voronoi method

A Voronoi diagram is a technique to partition a plane into regions based on a specified set of points. Every point in the region corresponding to a point in the set of points is closer to this point than any other point in the set of points (Berg et al., 2008). In Figure 3.1 an example of a Voronoi diagram can be seen. This diagram can be useful in football with the plane being the pitch and the set of points being the players.

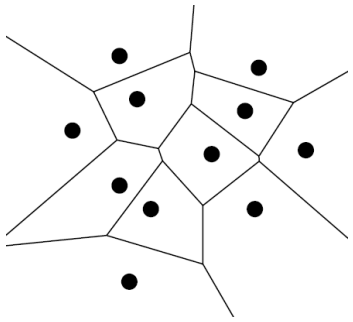
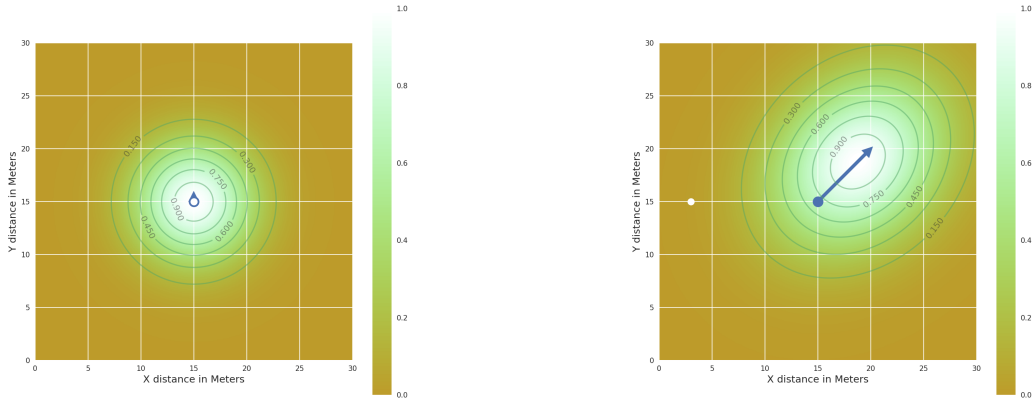


Figure 3.1: A simple example of a Voronoi diagram (Berg et al., 2008).

When it comes to evaluating the passes, the Voronoi diagram is created at the moment of initiation of the pass. As the location where the pass arrives is known, it is known in which region the pass arrives. The player to whom this region belongs is the expected receiver of the pass.

3.1.2 Fernandez method

Fernández and Bornn (2018) came up with the idea that players have influence on the area around them. They argue that this influence area should be shaped by three factors. The first is that the further a player is from the ball, the more time he has to react if the ball were to move towards him. Levels of influence will therefore reach wider if the player is further from the ball. Additionally, they say that the player's velocity has an important role in defining the area of influence. In the direction of it's speed the player will have more influence and this influence increases when the speed increases. Lastly,



(a) Player influence area for a player in possession of the ball with a speed close to zero.

(b) Player influence area for a player 15 meters away from the ball and running at 6.36 m/s.

Figure 3.2: A comparison of two situations with differences on all three influential aspects. (Fernández & Bornn, 2018).

they mention that the further away a point is, the less influence the player will have on this point, as it would take longer to reach it.

Fernández and Bornn (2018) argue that expressing the player influence area with a probability density function of a bivariate Gaussian distribution suits the needs of such an area. The shape of the influence area can then be adjusted by incorporating the three influential situational factors. The resulting differences caused by differences in the influential factors can be seen in Figure 3.2.

In order to delve deeper into the specifics of the this method, readers are encouraged to consult the appendix provided in Fernández and Bornn (2018), *'Wide Open Spaces: A statistical technique for measuring space creation in professional soccer'*. This appendix offers the details on how the player influence area can be calculated.

Fernández and Bornn (2018) not only use this player influence area to get pitch control on player level, they also use pitch control on a team level. They do this by summing the player influence of each player at each point of the pitch. This is then transformed to a value within the [0,1] range. In Figure 3.3 a situation can be seen where the pitch control for the red team is visualised.

Both a model based on individual player influence areas and a model that is based on the team-level pitch control will be used as baseline. The visualisations, however, will all be done on team level. When the visualisation is done for a model on player level, the player's team is used for the color. This is because visualising with colors on player level would involve too many colors and comprehensibility would be affected. Furthermore, the visualisations are to be compared to each other, this is why the Fernandez visualisa-

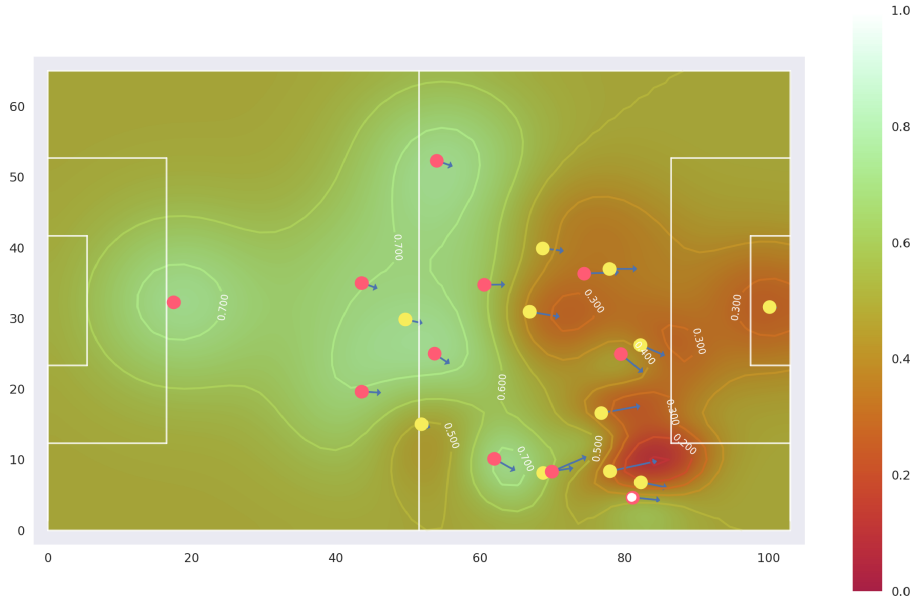


Figure 3.3: Pitch control surface indicating the degree of control for the red team (red dots). The greener the surface the more control the red team has over that area. The arrows show players velocities and the axis dimensions are in meters (Fernández & Bornn, 2018).

tions are also done binary instead of showing degree of control such as in Figure 3.3.

3.2 Machine learning methods

The literature that was discussed in the introduction showed that random RF and NNs usually perform best in studies that use similar data. Therefore, this study is also limited to these two methods. The literature also mentions that CNNs and models with LSTM features perform well.

The CNNs are mainly used to process data on football matches that consists of images. While the data can be adjusted in order to get a similar manner of data analysing, this requires a lot of work and the benefits are unknown. As time is limited for this study, this will not be in the scope of the study.

LSTM models could have benefits for analysing the passes as it takes into account the temporal aspect of the data. The model could for example learn from a players movement that he is anticipating to receive a pass even before it is given. Literature found on this particular model, uses video data and thus a CNN as a basis of their model Q. Zhang et al., 2022; Cai et al., 2020; Sarabu and Santra, 2021).

Implementing an LSTM model would probably increase performance. However, data with high quality is however scarce for this study, while LSTM models require extensive datasets of high quality (Al-Selwi et al., 2024). Therefore, together with the time limitations, LSTM models will not be considered in thi study.

Instead of having images or similar input data, this study will use basic statistics as input such as the distance per player to the receiving point. This is the same as the Voronoi model and therefore the benefit of ML models may not directly seem clear. However, these basic statistics combined might still contain hidden information. Adding the x-coordinate of the position of pass initiation and the position of receiving the pass might contain information on what risks players are taking. As mentioned before, a player would not quickly take any risks when being in front of their own goal.

3.2.1 Model input structure and content

The data will be given to the ML models in the form of a pandas dataframe. Each row represents a pass. The first few columns in this dataframe will have general information on the pass, such as the distance it travels.

The columns after that have information on each player. For each player there may be several columns such as the distance to point of reception, the x-coordinate and the speed in each direction. For each player these columns will be next to each other. The groups of columns of players in the same team are also next to each other (here there is no general order in teammates of the passer or not).

Three different combinations of statistics will be evaluated. The first is only the distance of each player to the point of pass reception, similar to the Voronoi model. The second is the distance of the pass combined with the x-coordinate of the location of the pass initiation, the x-coordinate of the location of the pass reception and the x-coordinate of each player. This is to see what happens if small amounts of information are added. The last combination is a set of statistics that is found to get the best accuracy results. The set is visible in Table 3.1.

Table 3.1: The contents of the third set of statistics.

'player_x'	The x-coordinate of the player
'player_y'	The y_coordinate of the player
'player_dist'	The distance of the player to the ball
'player_speed_ball_dir'	The velocity of the player in the direction of the ball
'player_angle'	The angle between the location of pass initiation to the location of the player and the x-axis
'player_absolute_speed'	The speed of the player
'player_max_speed'	The maximum speed the player has reached during the match
'player_x_dist_pas'	The distance on the x-axis between the player and the position of the pass initiation
'player_y_dist_pas'	The distance on the y-axis between the player and the position of the pass initiation
'player_x_dist_rec'	The distance on the x-axis between the player and the position of the pass reception
'player_y_dist_rec'	The distance on the y-axis between the player and the position of the pass reception

Many possible statistics and combinations have been tried. The sklearn's feature_importances function was used to determine which statistics should be used. An example of a feature importance graph that was used can be seen in Figure 3.4.

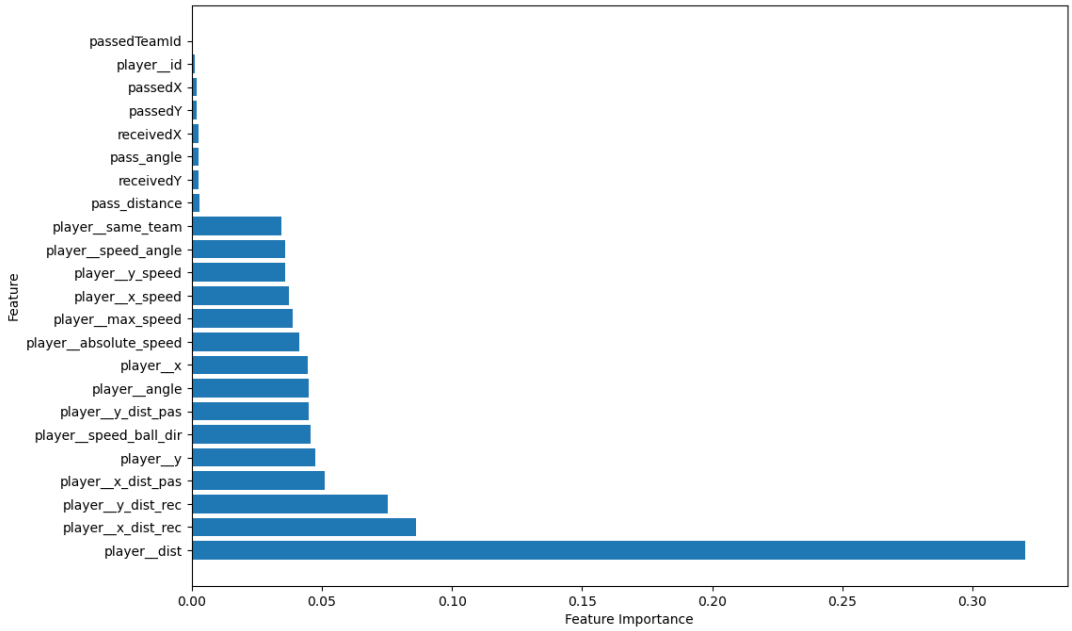


Figure 3.4: An example of a used feature importance graph. Some of the features have not been used as they has low importance for the predictions.

3.2.2 Random forest model

The RF model that is used is the RandomForestClassifier function of the Sklearn package in Python. This model will be trained for many datasets. Many combinations of pass datasets and different sets of used statistics will be tested. Therefore, not all hyperparameters are tuned as this would be too time consuming. Only the number of estimators and the max depth of the trees are tuned. This is done with by doing a grid search with the GridSearchCV pakcage provided by sklearn.

For each combination of pass datasets and set of used statistics the grid search is done separately. In each case, the model performed best without limiting the depth of the trees. The optimal number of estimators for most situations is very high, sometimes even higher than 1000. However, above a 1000 estimators, the difference in performance becomes small, while computational costs are very high. This is why the number of estimators is limited to 1000. The chosen number of estimators is 1000 for each situation, except where the balanced dataset is used. Here the optimal number of estimators ranges between 500 and 800 for the different sets of statistics.

3.2.3 Artificial neural network model

For building the NNs, the Tensorflow package is used in Python. The NNs have relatively simple structures because of the basic data structures. The data structure requires at least one convolutional layer as this enables the model to learn which columns are connected with each other by being on the same player.

Again, because many different datasets and sets of statistics will be tested, many different settings of the model will be used. The number of layers correlates with the complexity of the training data. The model that is trained with only the distances has one convolutional, and one dense layer. The models that used the distances and x-coordinates, have two convolutional layers and one dense layer. The models that use thw large set of statistics have three convolutional layers and one dense layer. The amount of layers and the amount of nodes for each layer were found by trial and error.

In order to prevent overfitting, and thus increase generalisation, every layer will have a dropout added to it. For the convolutional layers this is 25% and for the dense layers this is 50%. The difference is because the convolutional layers only find patterns between the columns, while the dense layers find patterns over the whole row of information. The second is more prone to overfitting and thus a higher dropout rate is required. All layers use a Rectified Linear Unit (ReLU) activation.

The amount of epochs was found by trial and error and ranges from 15 epochs to 90 epochs. Furthermore, to limit time spent testing setting, it was chosen to always use a batch size of 32.

4. Results

In this chapter the results of the different methods and models will be presented. First the results on the baseline methods are discussed, followed by the newly created models. The results will be on different subsets of the passes. This is to be able to evaluate the influence of consistency/quality of the data and the influence of the amount of data on the models.

The first subsets is the whole dataset. The second subset is the dataset filtered for 11 players per side, which should increase data quality and consistency but decreases the amount of data. The third dataset is a dataset balanced for an equal number of passes that do and do not end up with a team mate of the passer. With this any hidden biases of the models may be found. The last subset is a combination of filtering for 11 players per side and having a balanced dataset.

The balancing of the dataset is done with undersampling the succeful passes. Other sampling techniques have been disregarded due to time limitations.

4.1 Baseline methods

As can be seen in Table 4.1 the Voronoi method outperforms the Fernandez method on all subsets of the dataset when predictions are on player level. In Table 4.2, which is the accuracies of the predictions on team level, the Voronoi model again outperforms the Fernandez method. Only in one set does the Fernandez method on player level outperform the Voronoi model, but the difference there is not much.

The data seems to get less predictable once the success rate of the passes is balanced. This is not surprising as successful passes are on average closer to the receiver than unsuccessful passes. Unsuccessful passes thus have a higher probability of being just over the border of the area allocated to the receiving player.

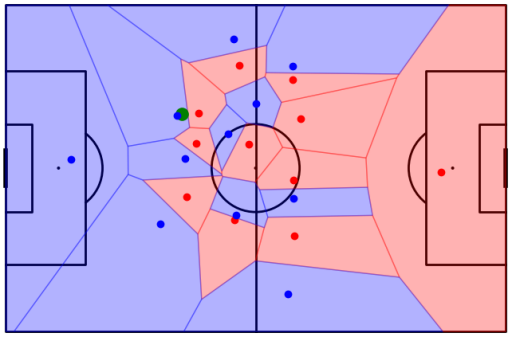
Table 4.1: Accuracies of the baseline methods for predicting which player will receive the ball. The accuracies are on different subsets of the passes.

	Pass dataset			
Method	**All passes**	**11 a side**	**Balanced**	**Balanced and 11 a side**
Voronoi	**72,95%**	**74,54%**	**66,56%**	**68,26%**
Fernandez, player level	69,05%	72,57%	64,66%	66,09%

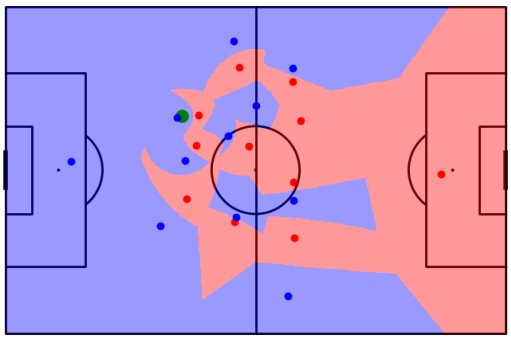
Table 4.2: Accuracies of the baseline methods for predicting which team will receive the ball. The accuracies are on different subsets of the passes.

	Pass dataset			
Method	**All passes**	**11 a side**	**Balanced**	**Balanced and 11 a side**
Voronoi	**83,02%**	**83,84%**	**78,66%**	79,04%
Fernandez, player level	81,80%	82,69%	77,98%	78,97%
Fernandez, team level	79,94%	81,61%	78,12%	**79,12%**

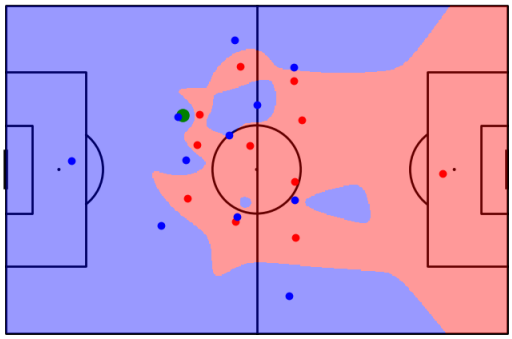
The visualisations of the baseline methods can be seen in Figure 4.1. Besides some small differences, the Voronoi metod in Figure 4.1a and the Fernandez method on player in Figure 4.1b level are very similar. Figure 4.1c shows a bit larger differences with the other two, some of the blue players are for example in red areas. The differences are, however, still very small.



(a) Visualisation of the predictions of the Voronoi method.



(b) Visualisation of the predictions of the Fernandez method on player level (with tile sizes of 0.2m by 0.2m).



(c) Visualisation of the predictions of the Fernandez method on team level (with tile sizes of 0.2m by 0.2m).

Figure 4.1: Visualisations of the baseline methods for a single timestamp and for the whole pitch.

4.2 Machine learning methods

The accuracies of the ML models on individual player basis can be seen in Table 4.3. Both the RF and the NN start to perform better when the amount of information they are

trained with is increased. Both the number of passes and the amount of different statistics per player increase the performance.

The performance of the NN has some correlation with the dataset size. Once the size of the dataset becomes smaller the performance starts to decrease. And while when the dataset is large, the performance increases with the data complexity, the performance actually decreases with the complexity when the dataset is small. Indicating that the model is not trained on enough data to find the more complex patterns. The RF model is affected in the same way but slightly less.

Both models outperform the baseline models when there is enough data. When data is more scarce, the baseline models start to perform better as they did not need training.

Table 4.3: The resulting accuracies of the ML models on player level on the different subsets of the passes. As these values are affected by randomness, every value is averaged over 5 times.

	Pass dataset			
Method	**All passes**	**11 a side**	**Balanced**	**Balanced and 11 a side**
NN, only distances	74,50%	**74,50%**	61,41%	57,91%
RF, only distances	74,51%	71,22%	64,50%	61,17%
NN, distances and x-coordinates	73,44%	69,06%	61,97%	55,52%
RF, distances and x-coordinates	75,69%	74,46%	64,32%	**61,19%**
NN, extended set of statistics	76,02%	69,42%	57,64%	54,87%
RF, extended set of statistics	**77,36%**	73,56%	**66,43%**	59,7%

The visualisations of the ML models can be seen in Figure 4.2. The visualisations are very different than those of the baseline models and are do not well represent pitch control. The visualisations are of predictions by models trained with the whole pass dataset. The visualisations with models trained with subsets of the passes show similar patterns. During calibration of the models some other patterns have been seen, but these patterns were not better than the ones visible.

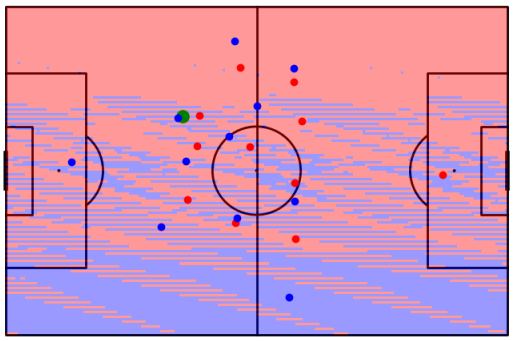
In Table 4.4 the accuracies of the ML models on team basis can be seen. For these models, the pass datasets were required to be balanced. The models would otheriwse simple predict that some teammate gets the ball and then the succesrate of the passes would decide the accuracy. Even when not indicating who the teammates are, the ML models are able to find these patterns.

Compared to the baseline models these models seem to underperform. However, just like the models on player level, these models are affected by the datasets being smaller. With an increased dataset, these models might perform just as well as the baseline models or even better.

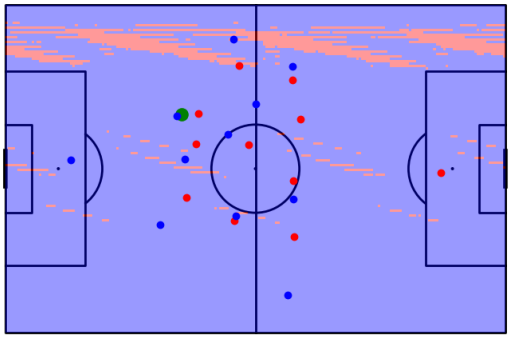
Table 4.4: The resulting accuracies of the ML models on team levelthe different subsets of the passes. As these values are affected by randomness, every value is averaged over 5 times.

	Pass datasets	
Method	**Balanced**	**Balanced and 11 a side**
NN, only distances	**78,98%**	75,82%
RF, only distances	74,52%	73,88%
NN, distances and x-coordinates	72,34%	65,37%
RF, distances and x-coordinates	69,42%	73,13%
NN, extended set of statistics	67,70%	63,43%
RF, extended set of statistics	68,73%	**76,87%**

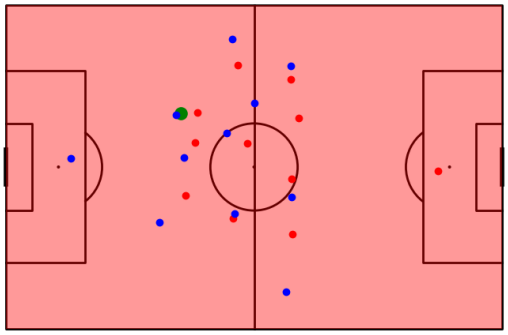
The visualisations of the models that predicted on team level are very similar to those on player level. Therefore, the figures in 4.2 are representative for these models and thus they do not perform well on visualisations either.



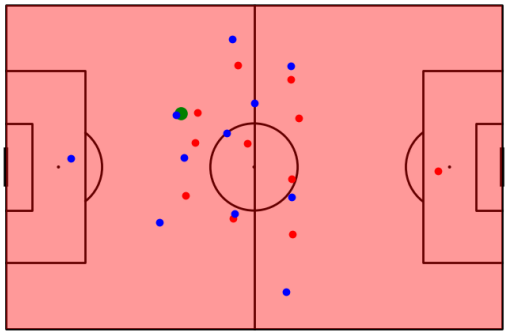
(a) NN model with distances as data.



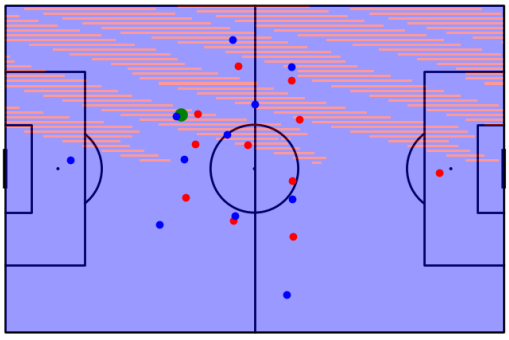
(b) RF model with distances as data.



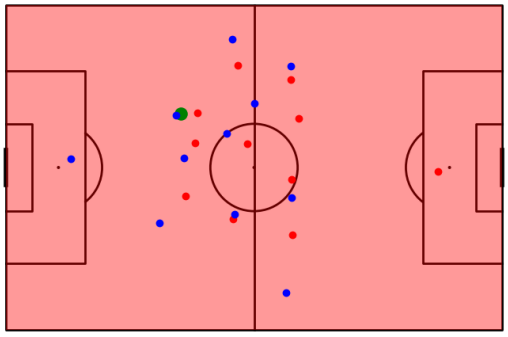
(c) NN model with distances and x-coordinates as data.



(d) RF model with distances and x-coordinates as data.



(e) NN model with the extended set of statistics.



(f) RF model with the extended set of statistics.

Figure 4.2: Visualisations of the ML models for a single timestamp and for the whole pitch. For training the models that made these predictions, the complete pass dataset was used.

5. Conclusion

When there is enough data to train the ML models properly, they are better at predicting the receivers of the passes than the baseline methods.

This study tried to answer the following question: *'How can the implementation of machine learning improve the performance of pitch control models in football analytics while maintaining interpretability?'* This question is tried to be answered with training ML models on pass data to predict who will receive the pass. This is done with having in mind that when somone receives a pass on a certain location, aparently this player was in control of that location. After this, the models are used to predict who would receive the ball for each point on the pitch. These predictions are used to asses the interpretability of the visualisations of the model.

The findings show that the ML models achieve higher accuracies in predicting who will receive the pass than the baseline models. This is, however, only when there is enough data to properly train the models. Furthermore, despite the higher accuracy, the visualisations created by the ML models were not easy to interpret. This suggests predicting the recipients of passes is a difficult way of assessing pitch control.

5.1 Discussion

This study was based on ML models. ML models are highly dependent on the data they are trained with. The data used in this study is limited in both size and quality. Another limitation is time allocated for this study. The high pace that resulted from this, forced quick decisions that may not always have been the right one, and coming back from decisions is difficult with limited time.

The ML models performed well in predicting the recipients of passes. However, evaluating the visualisations reveals that the current pass dataset does not suit the needs of the ML models.

To improve the performances on visualisations, it is recommended to expand the pass dataset to include a large variety of passes. Currently, uncommon passes, like passes toward a location where nobody is, are underrepresented. Ideally, a large dataset would be collected by having players on the pitch with the ball being thrown to random locations, and then having players react as if it were a real match. However, this approach would not be worth the trouble as it would require thousands, if not tens of thousands, of repetitions.

The overall quality of the data is another point that could be improved. Most matches have tracking data with a precision of only 1 meter, and there are a lot of instances where trackers did not work properly. The problems are not limited to these two examples, there is alot more that needs improvement.

Another recommendation is delving deeper into the patterns identified by the ML models that influence the visualisations. Overfitting at the match level could have happened, For example, a right back may frequently pass to the center back or right winger. Another problem could be that normalisation processes have impacted the visualisations. Besides these two examples, the cause of the problematic visualisations could be because of many other reasons.

Furthermore, football is a dynamic game, and passes are almost never identical. Passes to the same location can be given in different heights, speeds. Also, different events might have happens before the pass. For instance, a sudden acceleration by a player might indicate that he wants to receive the ball. An LSTM model, as previously discussed, could find patterns in these events.

A last recommendation is that while it is tempting to use the Fernandez model, as it seems more advanced, if possible, the simpler Voronoi model should be used.

In conclusion, improving the dataset, addressing data quality issues, analysing model patterns, and selecting appropriate models are crucial steps toward creating an improved model that performs well on predictions and has easily interpretable visualisations. This improved model could improve the overall usability and interpretability of the concept of pitch control.

Appendices

A Extra information on the matches

A.1 Table of all matches

In the table some information on the matches can be seen.

index	Match date	Pitch width (m)	Pitch length (m)	Measurement precision	Nr. of passes
0	2023/03/27	64.0	100.0	centimetre	548
1	2023/05/23	69.0	103.0	centimetre	732
2	2023/03/28	68.0	104.0	centimetre	748
3	2022/02/17	63.0	100.0	metre	772
4	2022/02/10	63.0	100.0	metre	420
5	2022/02/10	63.0	100.0	metre	697
6	2022/02/17	63.0	100.0	metre	730
7	2023/01/31	69.0	102.0	metre	514
8	2023/09/26	69.0	102.0	centimetre	511
9	2023/06/24	69.0	102.0	centimetre	705
10	2023/04/18	64.0	100.0	centimetre	675
11	2022/04/13	58.0	98.0	metre	1195
12	2022/03/30	58.0	98.0	metre	890
13	2022/05/12	63.0	99.0	metre	546
14	2022/05/05	63.0	99.0	metre	725
15	2022/05/03	63.0	99.0	metre	851
16	2022/05/05	63.0	99.0	metre	610
17	2022/05/12	63.0	99.0	metre	552
18	2022/05/12	63.0	99.0	metre	536
19	2024/01/17	68.0	103.0	centimetre	904
20	2024/01/17	68.0	103.0	centimetre	234
21	2023/05/25	64.0	100.0	centimetre	639
22	2023/03/21	64.0	104.0	centimetre	537
23	2021/06/24	65.0	103.0	metre	701
24	2022/04/26	67.0	104.0	metre	815
25	2022/05/17	67.0	104.0	metre	813
26	2022/04/05	65.0	102.0	metre	962
27	2022/02/22	63.0	100.0	metre	747
28	2022/02/24	63.0	100.0	metre	487
29	2022/04/12	64.0	104.0	metre	551
30	2022/04/06	64.0	104.0	metre	562
31	2022/01/07	60.0	100.0	metre	967
32	2022/03/02	61.0	105.0	metre	861
33	2024/04/09	69.0	102.0	centimetre	628
34	2024/03/19	69.0	102.0	centimetre	325
35	2022/05/18	56.0	100.0	metre	1153
36	2021/04/24	68.0	105.0	metre	712
37	2022/05/09	63.0	100.0	metre	740
38	2022/05/16	63.0	100.0	metre	481
39	2021/06/04	68.0	105.0	metre	798
40	2021/05/25	64.0	100.0	metre	792
41	2021/03/27	64.0	100.0	metre	650
42	2021/04/10	64.0	100.0	metre	643
43	2021/04/17	64.0	100.0	metre	755
44	2021/03/27	64.0	100.0	metre	661
45	2021/04/10	64.0	100.0	metre	635
46	2021/04/17	64.0	100.0	metre	694

B Detailed processing steps

B.1 Smoothing

In Table B.1 the number of affected passes can be seen for the different smoothing methods on the matches with data with centimetre precision. It can be seen that the MA methods perform better. Furthermore, as expected, when the window size and the the σ are increased, fewer passes are affected.

Table B.1: The number of passes affected at three levels of intensity per smoothing type for matches with tracking data measured to centimetre precision.

	Nr. of passes affected		
Smoothing type	**>10m/s**	**>15m/s**	**>20m/s**
No smoothing	346	74	64
MA window = 3	87	68	54
MA window = 5	68	33	20
Gaussian $\sigma = 0.5$	198	72	71
Gaussian $\sigma = 0.7$	132	79	68
Gaussian $\sigma = 0.9$	115	73	50

When comparing the different trajectories in Figure B.1 some conclusions can be made. No real over-smoothing can be seen in the player trajectories after Gaussian smoothing. In the trajectory of the player after MA smoothing with a window size of 3, no real over-smoothing can be found either. In the player trajectory after MA smoothing with a window of size 3, some signs of over-smoothing are visible at the sharp corners of the trajectory. Therefore, a choice is made for MA smoothing with a window size of 3, which balances the number of affected passes and the realism in player trajectories.

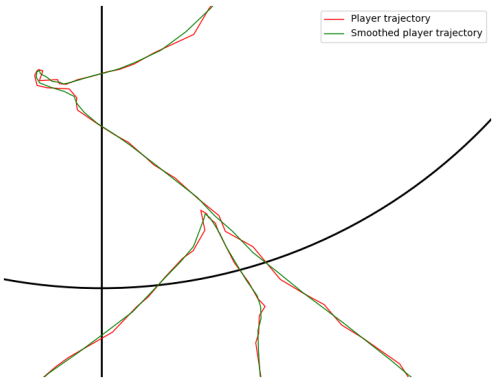
In Table B.2 the information on the affected passes in the matches with data with metre precision can be seen. In this case, the performance of the MA- and Gaussian method do not differ that much.

Table B.2: The number of passes affected at three levels of intensity per smoothing type for matches with tracking data measured to metre precision.

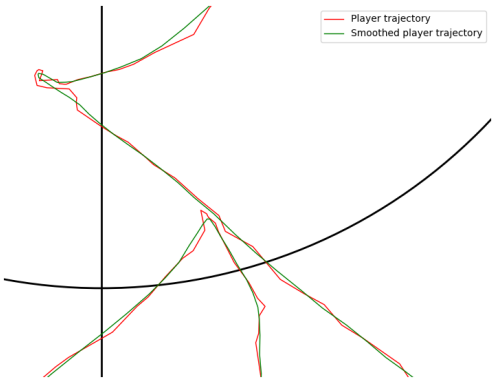
	Nr. of passes affected		
Smoothing type	**>10m/s**	**>15m/s**	**>20m/s**
No smoothing	1916	40	1
MA window = 3	106	47	30
MA window = 5	89	43	17
Gaussian $\sigma = 0.7$	110	28	13
Gaussian $\sigma = 0.9$	83	32	20
Gaussian $\sigma = 1.1$	82	32	18

The trajectories in Figure B.2 all show unrealistic trajectories, indicating the difficulty of smoothing data that is only measured up to metre precision. The trajectories of the MA

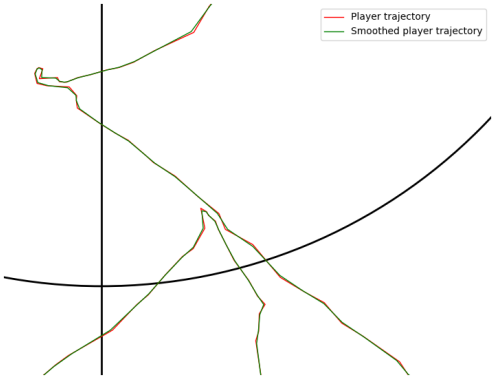
method look slightly more unrealistic as corners are not rounded. Therefore, a choice for Gaussian smoothing seems more accurate. According to Table B.2, the Gaussian smoothing with $\sigma = 0.7$ performs slightly worse than the others, while the others perform almost the same. As $\sigma = 0.9$ and $\sigma = 1.1$ perform almost the same in Table B.2, a choice is made for $\sigma = 0.9$ in order to avoid any over-smoothing.



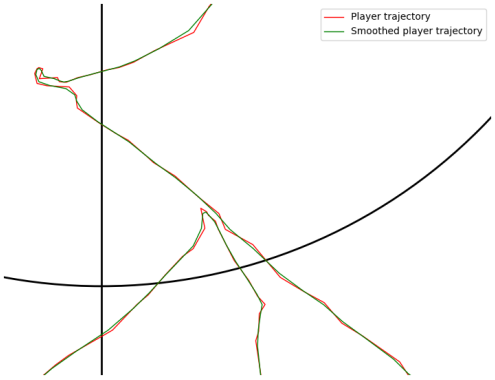
(a) MA, window=3



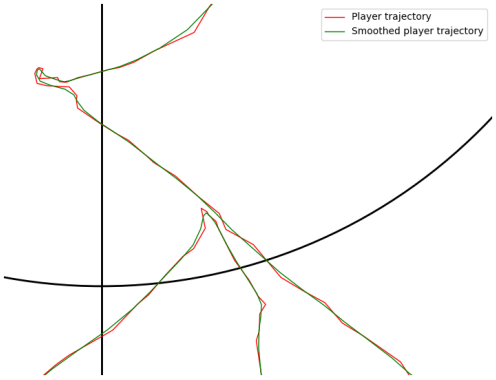
(b) MA, window=5



(c) Gaussian, $\sigma = 0.5$

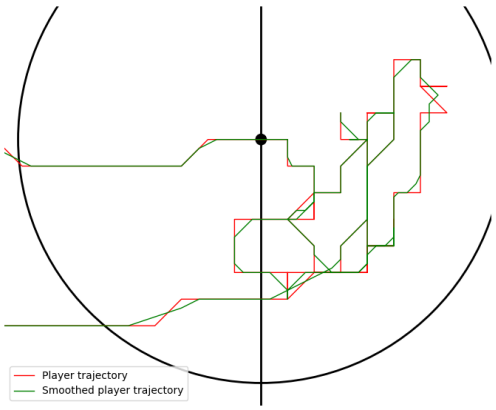


(d) Gaussian, $\sigma = 0.7$

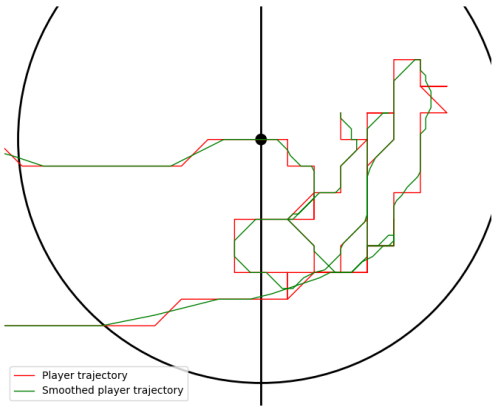


(e) Gaussian, $\sigma = 0.9$

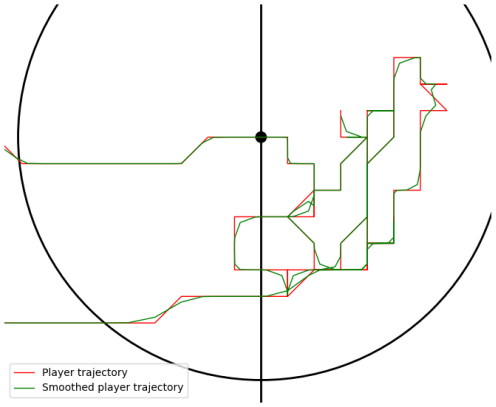
Figure B.1: Trajectory of a player after the different smoothing techniques for data with a precision of one centimetre.



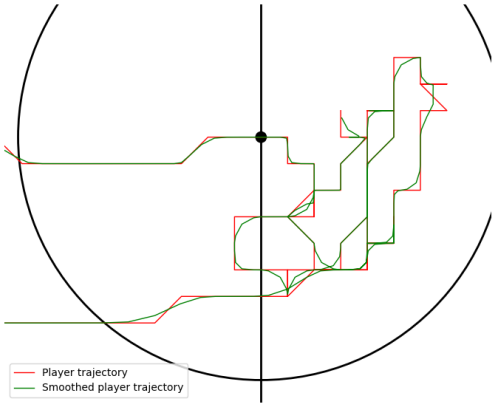
(a) MA, window=3



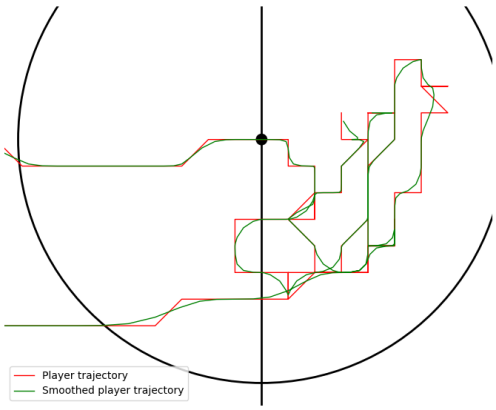
(b) MA, window=5



(c) Gaussian, $\sigma = 0.7$



(d) Gaussian, $\sigma = 0.9$



(e) Gaussian, $\sigma = 1.1$

Figure B.2: Trajectory of a player after the different smoothing techniques for data with a precision of one metre.

B.2 Play direction corrections

For the start of this method, four ratios are needed for each timestamp. These ratios are defined as follows: Let t be a given timestamp of a match and T be the set of all timestamps of that match. Define $T_{t,b} = \{t' \in T \mid t' < t\}$ and $T_{t,a} = \{t' \in T \mid t' > t\}$, where the small b indicates before this timestamp and the small a after this timestamp. Let $x_1(t')$ and $x_2(t')$ be the average x values for team 1 and team 2 at timestamp t', respectively. Then the ratios for each timestamp t are given by the

Then ratio $R_{1,b}(t)$, $R_{1,a}(t)$, $R_{2,b}(t)$ and $R_{2,a}(t)$ for each timestamp t are given by equations (B.1a), (B.1b), (B.1c) and (B.1d). The small b again indicates before this timestamp and the small a after this timestamp. The small 1 and 2 indicate the team.

$$R_{1,b}(t) = \frac{|\{t' \in T_{t,b} \mid x_1(t') < x_2(t')\}|}{|T_{t,b}|} \tag{B.1a}$$

$$R_{1,a}(t) = \frac{|\{t' \in T_{t,a} \mid x_1(t') < x_2(t')\}|}{|T_{t,a}|} \tag{B.1b}$$

$$R_{2,b}(t) = \frac{|\{t' \in T_{t,b} \mid x_2(t') < x_1(t')\}|}{|T_{t,b}|} \tag{B.1c}$$

$$R_{2,a}(t) = \frac{|\{t' \in T_{t,a} \mid x_2(t') < x_1(t')\}|}{|T_{t,a}|} \tag{B.1d}$$

The progress of these different ratios over the timestamps of one match can be seen in Figure B.3. This figure is of the data of the match with index 20 in appendix A.1

With these four ratios two new ratios are created, $R_1(t)$ and $R_2(t)$. This is one ratio per timestamp for each team. The ratios are not calculated in the exact same way. How they are calculated can be seen in equations (B.2a) and (B.2b).

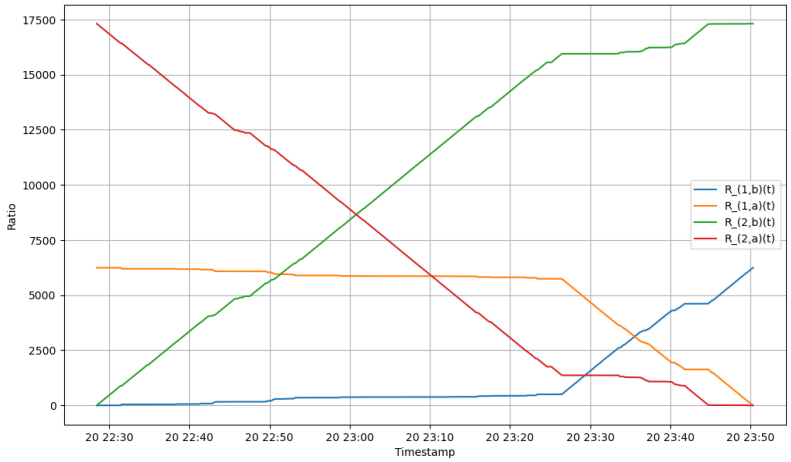


Figure B.3: The first tier of ratios over the course of one match

$$R_1(t) = \frac{R_{1,b}(t)}{R_{1,a}(t)} \tag{B.2a}$$

$$R_2(t) = \frac{R_{2,a}(t)}{R_{2,b}(t)} \tag{B.2b}$$

In Figure B.4 the progress of these two ratios over the same match as in Figure B.3 can be seen. Where these lines cross each other is the moment the teams switch sides, and thus switch playing direction.

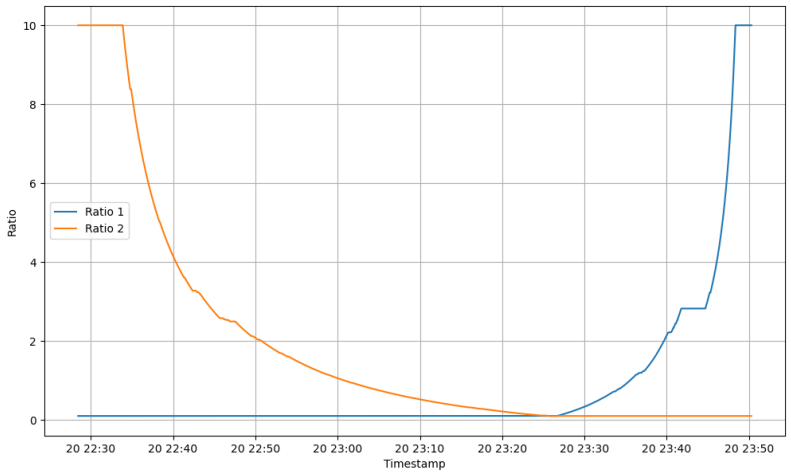


Figure B.4: The second tier of ratios over the course of one match

Bibliography

Agyeman, R., Muhammad, R., & Choi, G. S. (2019). Soccer video summarization using deep learning. *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 270–273. https://doi.org/10.1109/MIPR.2019.00055

Ai, S., Na, J., Silva, V. D., & Caine, M. (2021). A novel methodology for automating spatio-temporal data classification in basketball using active learning. *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, 39–45. https://doi.org/10.1109/PRML52754.2021.9520715

Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A., Sumiea, E. H., Alqushaibi, A., & Ragab, M. G. (2024). Rnn-lstm: From applications to modeling techniques and beyond—systematic review. *Journal of King Saud University - Computer and Information Sciences*, *36*(5), 102068. https://doi.org/https://doi.org/10.1016/j.jksuci.2024.102068

Berg, M., Cheong, O., Kreveld, M., & Overmars, M. (2008, January). Voronoi diagrams. https://doi.org/10.1007/978-3-540-77974-2_7

Brefeld, U., Lasek, J., & Mair, S. (2019). Probabilistic movement models and zones of control. *Mach Learn*, *108*, 127–147. https://doi.org/10.1007/s10994-018-5725-1

Cai, J., Hu, J., Tang, X., Hung, T.-Y., & Tan, Y.-P. (2020). Deep historical long short-term memory network for action recognition. *Neurocomputing*, *407*, 428–438. https://doi.org/https://doi.org/10.1016/j.neucom.2020.03.111

Chawla, S., Estephan, J., Gudmundsson, J., & Horton, M. (2017). Classification of passes in football matches using spatiotemporal data. *ACM Transactions on Spatial Algorithms and Systems*, *3*(2), 1–30. https://doi.org/10.1145/3105576

Fernández, J., & Bornn, L. (2018). Wide open spaces: A statistical technique for measuring space creation in professional soccer.

Hen, I., Sakov, A., Kafkafi, N., Golani, I., & Benjamini, Y. (2004). The dynamics of spatial behavior: How can robust smoothing techniques help? *Journal of Neuroscience Methods*, *133*(1), 161–172. https://doi.org/https://doi.org/10.1016/j.jneumeth.2003.10.013

K., I., Saya, A., BhagyashriShetty, H., Manaswini, C., & Ashjay, C. A. (2024). Survey on machine learning models to analyze urinary tract infection data. *International Research Journal on Advanced Engineering and Management (IRJAEM)*. https://api.semanticscholar.org/CorpusID:269273435

Lin, H., Bao, M., & Kang, C. (2024). A spatiotemporal deep learning-based multisource data analytics framework for basketball game. *IEEE Access*, *12*, 73066–73078. https://doi.org/10.1109/ACCESS.2024.3403688

Martens, F., Dick, U., & Brefeld, U. (2021). Space and control in soccer. *Front. Sports Act. Living*. https://doi.org/10.3389/fspor.2021.676179

Najeeb, H. D., & Ghani, R. F. (2020). Tracking ball in soccer game video using extended kalman filter. *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 78–82. https://doi.org/10.1109/CSASE48920.2020.9142058

Perš, J., Bon, M., Kovačič, S., Šibila, M., & Dežman, B. (2002). Observation and analysis of large-scale human motion. *Human Movement Science*, *21*(2), 295–311. https://doi.org/https://doi.org/10.1016/S0167-9457(02)00096-9

Sarabu, A., & Santra, A. K. (2021). Human action recognition in videos using convolution long short-term memory network with spatio-temporal networks. *Emerging Science Journal*, *5*(1), 25–33.

Spearman, W. (2018). Beyond expected goals.

Taki, T., & Hasegawa, J. (2000). Visualization of dominant region in team games and its application to teamwork analysis. *Proceedings Computer Graphics International 2000*, 227–235. https://doi.org/10.1109/CGI.2000.852338

Taki, T., Hasegawa, J., & Fukumura, T. (1996). Development of motion analysis system for quantitative evaluation of teamwork in soccer games. *Proceedings of 3rd IEEE International Conference on Image Processing*, *3*, 815–818 vol.3. https://doi.org/10.1109/ICIP.1996.560865

Xie, P., Li, T., Liu, J., Du, S., Yang, X., & Zhang, J. (2020). Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion*, *59*, 1–12. https://doi.org/https://doi.org/10.1016/j.inffus.2020.01.002

Zhang, G. (2023). *The analysis of team tactical behaviour in football using gnss positional data* [Other thesis]. Liverpool John Moores University [Available under License Creative Commons Attribution Non-commercial]. https://doi.org/10.24377/LJMU.t.00019091

Zhang, Q., Zhang, X., Hu, H., Li, C., Lin, Y., & Ma, R. (2022). Sports match prediction model for training and exercise using attention-based lstm network. *Digital Communications and Networks*, *8*(4), 508–515. https://doi.org/https://doi.org/10.1016/j.dcan.2021.08.008