5-010

FOOTBALL ANALYTICS BASED ON PLAYER TRACKING DATA USING INTERPOLATION TECHNIQUES FOR THE PREDICTION OF MISSING COORDINATES

Christos Kontos and Dimitris Karlis
Department of Statistics, Athens University of Economics and Business, Athens, Greece
kontos.christos@yahoo.com, karlis@aueb.gr

Corresponding Author: Dimitris Karlis **(ORCID: 0000-0003-3711-1575)**

Abstract

In recent days we have seen an increasing interest in using tracking data for sports and especially for football. Such data can reveal the location of the players and the ball many times per second allowing for examining tactics, efficiency of players, formations, and many other characteristics of the game. Unfortunately, such systems are still expensive, and their data are not widely available. As an alternative, limited tracking data can be obtained from broadcasting videos. They are of less quality and of course they are censored in the sense that they do not provide information for all players but only those in the frame taken. Within this framework, the primary aim of this paper, is the exploration of the most suitable method for retrieving the missing information of players' and ball's positions and rectify as much as possible the effect of censoring which leads to discontinuous player tracks and unreliable player identification. In this paper we explore and compare different interpolation methodologies. Moreover, we try to distinguish possible differences between the actual data, as they were tracked from the camera and the interpolated data that have been estimated from our best selected method, by extracting insights to support tactical analyses as well as players' performance evaluation.

Keywords: *Football analytics, Player tracking data, Missing values, Interpolation techniques, Regression and time series algorithms*

1. INTRODUCTION

Due to the rapid development of tracking technologies, processing software and more powerful data storages, analytics in sports industry and especially in football, has become increasingly popular in the last decade. Nowadays, teams and sports organizations are increasingly interested on data to inform players, coaches and other stakeholders, facilitate decision making both during and prior to sporting events. Major fields of applications of sports analytics include, scouting, revenue increase, performance analysis (Mohr et al., 2003), team tactics (Rein and Memmert, 2016), match outcomes (Karlis and Ntzoufras, 2003), player development (Rampinini et al., 2007), injuries prevention and rehabilitation (Dvorak et al., 2000) among others.

With the advancement of technology, a new data source that is commonly used nowadays has been created, the so-called tracking data (Rein and Memmert, 2016). In our days a huge amount of tracking data is being collected for nearly every popular professional team sport. More and more companies are now offer their services, using either camera-based systems, or wearable technology. Over the last few years, initiatives and techniques have been implemented around the area of player tracking data. One of the main ingenuities are the different implementations that are based on analyses

of coordinate data that are generated by converting a traditional broadcast video feed to player locations, utilizing different computer vision and artificial intelligence techniques (Lu et al., 2013). This type of data could be easily interpreted as broadcast tracking data. In contrast with the traditional multicamera tracking data, the broadcast tracking data are currently distinguished from their advancement of availability they provide (websites of Skillcorner and Sportlogiq for example).

As the collection of event and tracking data has been rapidly evolved, different statistical methodologies have been applied in various areas of interest. For example, Link et al., (2016) presented an approach to quantify the attacking performance in football, characterizing as dangerousity the probability of a goal being scored for every point in time at which a player is in possession of the ball. In addition, Bialkowski et al. (2016) an unsupervised method has been examined in order to learn a formation template which allows to align the tracking data at the frame level. An innovative hierarchical clustering method has also been developed by Diquigiovanni and Scarpa, (2018) in order to divide a sample of undirected weighted networks into groups. (Horton et al., 2014) presented a model that constructs numerical predictor variables from spatiotemporal match data using feature functions based on methods from computational geometry. Another research has been conducted, which estimated how both the risk and reward of a pass across tracking data can be measured (Power et al., 2017). Finally, Shaw and Glickman, (2019) developed an innovative model for the computation of the formation of any team at any moment of a match, by taking the position of team members relative to their teammates. The examples above are indicative of the increasing interest about tracking data and of course is far from being a complete list of the existing applications.

Broadcast tracking data contain less information as we collect data only for the players in the video frame. One major challenge, for applying existing methods for multicamera tracking data to broadcast tracking data is the issue of censoring and the related effect of missingness (Mortensen, 2020). In addition, the main information that is currently extracted from this type of data, is based at the location of events (passes, shots on target, goals, etc.) and players. Therefore, any action that happens outside the recorded events is not easily inferable and cannot be taken into consideration for future decisions. Another great challenge, is the ability to derive powerful insights from frame-by-frame tracking data and provide all the necessary information needed for the event of interest. Many sports organizations nowadays, are interested in relationships that can occur from specific outcomes or when they are using a specified type of formation when they are attacking or defending. By using this type of data, often leads to millions of possible locations of players and thus results in undesirable results due to the complexity of data.

Existing methods for tracking data when capturing players' movements, are widely used by professional football clubs to provide insights into activity demands during training sessions and competitive matches. Global positioning systems (GPS), local positioning systems (LPS) trackers and accelerometers, are some of the specific methods. In recent years, computer vision techniques can be applied directly to broadcast video in order to identify the precise location and movement of players and ball. Automated broadcast tracking data, have made a huge leap forward, as they enable locations of objects to be accurately collected from a standard television broadcast, without the need for any capital investment in stadiums or human operator costs (SkillCorner, 2020). However, despite their undisputed value, broadcast tracking data, lag behind on a very basic issue, and that is they are only available for a player and the ball as long as they are observed inside the main broadcast camera shoot, which pans left and right across the pitch (see, Figure 1).

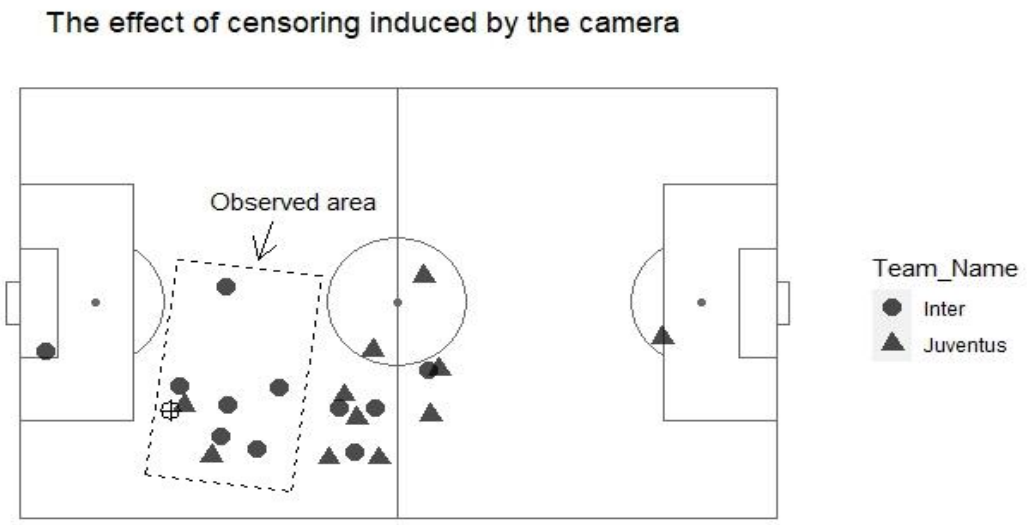


Figure 1: The effect of censoring induced by the camera. The rectangular area refers to the video frame, all players outside the rectangular cannot be positioned in the field and hence their locations are unknown.

Hence, our purpose in the current work, is to estimate the missing coordinates of the players and the ball, that are produced from the censoring effect of the broadcast camera. In that way, we tried to predict the exact position of each object during the live broadcast of a game. To make this clear we want to "estimate" the positions of the players not seen in the frame based on the existing information from the frame and past frames. We proceeded also with the exploration and discovery of the best and most accurate method for filling the missing coordinates of players and rectify as much as possible the effect of censoring, that is mainly produced from broadcast tracking data and often leads to discontinuous player tracks and unreliable player identification. We explore and compare different approaches, by using a number of various interpolation algorithms, non-linear regression models and a time series forecasting approach.

We also try to examine possible differences generated between the interpolated data and the actual data as they are derived from the broadcast camera, by extracting important insights that are based on the tactical analyses of teams as well as on the players' performances on different types of scenarios. Finally, as an extra step of further research, we try to derive important insights by effectively estimate teams' formations and players' consistencies based on their initial formations, as well as possible correlations when a team is attacking or defending, using an appropriate pitch control model that quantifies the probability of a player could control the ball assuming it is at that location.

The remaining of the paper is organized as follows: In Section 2, we present the data sources used on this work. The methodological approach is described in Section 3. Section 4 applies the methodology to the data. We present a comparison of the different approaches but also certain applications for tracking data based on the interpolated data so as to illustrate the potential of the methodology. Finally, concluding remarks, limitations and further research taken into consideration is also discussed in Section 5.

2. DATA USED

The dataset used for this work, retrieved from an open-source joint initiative between SkillCorner and Friends of Tracking. The specific repository consists of 9 football games of broadcast tracking data, collected through computer vision and machine learning out of the broadcast video and are referring to the 2019/2020 league matches between the champions and runners up in English Premier League, French L1, Spanish LaLiga, Italian Serie A and German Bundesliga. In the current paper, a match from the Italian Serie A, between Inter and Juventus was selected (around 600.000 observations). Each match consists of two files. The first file, contains all the necessary information about the lineup formations, players, referee and ball characteristics, pitch size, coaches etc. The second file, includes all the tracking data for the players, the referee and the ball. For the spatial coordinates, the unit of the field modulization is the meter and the center of the coordinates is at the center of the pitch (0,0). The initial predetermined dimensions of the field have a size of 105m x 68m and the x axis is the long side of the field while the y axis is the short side of the field. The frame of the video of the data comes from at 10 frames per second and the timestamp in the match has a precision of 1/10 seconds. The broadcast shows an average of 14 players out of 22 at each frame and during replays or close up views, the data is not included. Therefore, the actual data present missing information for all players throughout the duration of the game at different intervals each time. Finally, 95% of the player identity that is provided is accurate while for the rest 5% the algorithm could not identify for sure the player.

For the data cleansing and feature engineering part, several procedures were implemented before running our analyses. Firstly, we had to exclude all the unwanted list of elements existing in the tracking data file, as they did not correspond to the actual playing time and thus did not provide us with any important information. The elements of the tracking data file were related with information regarding the possession, the frames of the video, the coordinates of the players, as well as the period and the time of the game. Therefore, we tried to match every single player with the elements of the tracking data file, by using the unique identifier of each player, as exists in the first file which included all the necessary information about the players characteristics. As a next step, we treat each element of the list as a unique data frame, in order to include all the necessary information existed at each unique element, resulting to five (5) unique data frames of a total 70.299 observations each. By un-listing also, all the elements of the specific list (≈ 600.000 observations), we took care that each frame had to be repeated equal times as the number of elements that is included at each unique list of the initial list. The (x, y) coordinates of each object (players and ball) have been also rescaled, in order to be aligned with the common size of a field.

Finally, after a thorough check on the data, we decided to increase the number of players that were visible on each frame (14 players out of 22 were visible at each frame on average). At some instances of the game, if a player was not easily observable, i.e., it was not possible from the broadcasting to identify him, the column referring to the unique identifier of a player ascribes as values the “home team” or “away team” attributes accordingly. For the specific implementation, we relied on a variable acting like a unique identifier when a player is observed on consecutive frames. It should be mentioned that for a number of slim picking instances during the game, some players had also the same unique identifier with a different player. On that account, we tried to remove the players that were observed in the same frame, by keeping the player who appeared first in the frame and had the correct unique identifier with what had been stated to him from the beginning. This issue provoked from the fact that during the game the positions of two (2) or more players, were so close to each other that the technique of detecting a player eventually ended up detecting more than one.

Before the implementation of any of the proposed methods that are discussed in Section 3, we had to properly define first the proportions of missing observations for each one of the objects (players and ball), that were induced due to the censoring effect of the camera. Afterwards, we resample the data from unevenly time data to equi-spaced time data, as between time increments the data were missing without any specific ordinance or pattern (replays, close-up views, goals, etc.). This gives an idea of the kind of data destruction that we can have (Figure 2). In Figure 2 one can see the full data for a player at the left and those with missing values for the same player at the right, while we have randomly omitted some of the time stamps pretending that the data were censored. This resampling approach allows to check different proportion of missingness to investigate the effect. In principle what we want to be able to do is to derive the full path the left using the available data at the right. Note that for each player we have a different proportion of missingness and also different patterns of consecutive missing points. A simple example is the goalkeeper who can be unobserved for a long time if his teams is in offense.

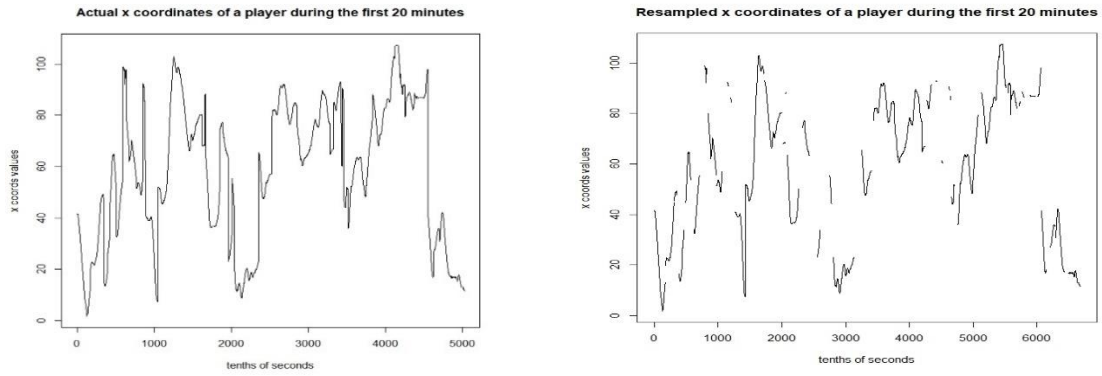


Figure 2: An example of data: the left-hand side refers to the full data for a player for a large time interval and the right one, data sampled from the full data to have missingness.

3. METHODOLOGY

3.1 INTERPOLATION OF MISSING DATA

In order to successfully address the problem, we compared several methods and we tried to retrieve the method whose predicted values were as close as possible to the actual positional data that we had in our disposal. For the implementation of this attempt, we used some interpolation algorithms known for their strong predictive ability in matters of interpolation of missing values using imputation techniques. Following that, we employed three (3) non-linear models by using imputation techniques via regression. Due to the structure of our data, we decided also to observe the predictive interpolation power of a time series forecasting technique, on the specific type of data. Finally, it should be stressed that, for all the regression type of models we have used some covariates, namely

- player's position,
- timestamp,
- player's distance from the ball at each frame and
- distance of the player travelled from the last few frames, which is a proxy of his speed.

Note that several other variables have been also tried but we have decided to use only those listed above based on our findings. Such variables not used in the analysis are speed and acceleration of the player, ball possession, ball position and referee position.

Regarding the interpolation algorithms, we based our approaches on methods related for indexed totally ordered observations (Grothendieck and Zeileis, 2005) and on cases associated with univariate time series imputation (Moritz and Beielstein, 2017). We have used different approaches, including linear interpolation and splines methodology.

3.1.1. LINEAR INTERPOLATION

Regarding the linear interpolation approach, we replaced each missing value with linear interpolated values via approximation. Specifically, linear interpolation is a method of curve fitting using linear polynomials to construct new data points within the range of a discrete set of known data points. In our case, if the two known points are given by the coordinates (x_0, y_0) and (x_1, y_1), the linear interpolation is the straight line between these points. For a value x in the interval (x_0, x_1), the value y along with the straight line is given from the equation of slopes $\frac{y-y_0}{x-x_0} = \frac{y_1-y_0}{x_1-x_0}$ and consequently by solving the equation for y, which is the unknown value at x gives the following which denotes the formula for linear interpolation for the interval (x_0, x_1): $y = y_0 + (x - x_0)\frac{y_1-y_0}{x_1-x_0} = \frac{y_0(x_1-x)+y_1(x-x_0)}{x_1-x_0}$.

3.1.2. SPLINES INTERPOLATION

For the spline interpolation approach that we followed, we replaced each missing value with a spline interpolation approach. Spline interpolation is a method of using piecewise polynomials. That is, instead of fitting a single, high-degree polynomial to all of the values at once, spline interpolation fits low-degree polynomials to small subsets of the values such that they fit smoothly together. More specifically, with cubic spline interpolation we tried to construct a spline $f: [x_1, x_{n+1}] \to R$, which consists of n polynomials of degree three, referred to as f_1 to f_n. Opposed to regression, the interpolated function traverses all $n + 1$ pre-defined points of a dataset D. The resulting function has the following structure:

$$f(x) = \begin{cases} a_1x^3 + b_1x^2 + c_1x + d_1, & if\ \ x \in [x_1, x_2] \\ a_2x^3 + b_2x^2 + c_2x + d_2, & if\ x \in (x_2, x_3] \\ \dots & \\ a_nx^3 + b_nx^2 + c_nx + d_n, & if\ x \in (x_n, x_{n+1}] \end{cases}$$

With properly chosen coefficients, $a_i, b_i, c_i\ and\ d_i$ for the polynomials, the resulting function traverses the points smoothly. For determining the coefficients, several equations are formulated which all together compose a uniquely solvable system of equations, such as the natural spline, the not-a-knot spline, the periodic spline and the quadratic spline. In our case, we employed a natural spline boundary condition, in order to minimize possible extrapolations. It should be noted that the natural spline is defined as setting the second derivative of the first and the last polynomial equal to zero in the interpolation function's boundary points.

3.1.3. STINE INTERPOLATION

A method sharing the good elements of the different method is also proposed (Stineman, 1980). In Stineman interpolation, missing values are replaced by piecewise rational interpolation and by default

the time index associated with each unique object is used for interpolation. According to Stineman, the interpolation procedure has the following properties: If values of the coordinates of the specified points change monotonically, and the slopes of the line segments joining the points change monotonically, then the interpolating curve and its slope will change monotonically. If the slopes of the line segments joining the specified points change monotonically, then the slopes of the interpolating curve will change monotonically. Finally, suppose that the first and second conditions are satisfied by a set of points, but a small change in the ordinate or slope at one of the points will result the previous mentioned conditions being no longer satisfied. Then making this small change in the ordinate or slope at a point will cause no more than a small change in the interpolating curve.

Let x_i, y_i denotes the rectangular coordinates of the i^{th} point on curve and y_i' the slope of the curve at i^{th} point. Given x such that $x_i \leq x \leq x_{i+1}$, the procedure for calculating y, which denotes the corresponding interpolated value, is defined using a slope between two points by $s_i = \frac{y_{i+1}-y_i}{x_{i+1}-x_i}$. Next $\Delta y_i = y_i + y_i'(x - x_i) - y_0$, where Δy_i denotes the vertical distance from the point (x, y_0) to a line through (x_i, y_i) with slope y_i'. Similarly, $\Delta y_{i+1} = y_{i+1} + y_{i+1}'(x - x_{i+1}) - y_0$, denotes the vertical distance from the point (x, y_0) to a line through (x_{i+1}, y_{i+1}) with slope y_{i+1}'. Therefore, according to Stineman, y can be calculated as follows:

If $\Delta y_i, \Delta y_{i+1} > 0$, then Δy and Δy_{i+1} have the same sign and the equation is the following:

$$y = y_0 + \frac{\Delta y_i \Delta y_{i+1}}{\Delta y_i + \Delta y_{i+1}} \quad (1)$$

If $\Delta y_i, \Delta y_{i+1} < 0$, an inflection point exists between x_i and x_{i+1} and the equation is derived accordingly from Stineman (1980):

$$y = y_0 + \frac{\Delta y_i \Delta y_{i+1}(x - x_i + x - x_{i+1})}{(\Delta y_i + \Delta y_{i+1})(x_{i+1} - x_i)} \quad (2)$$

In this work, we tried also to observe the predictive power of some robust models beyond the interpolation algorithms' ability. The most suitable algorithms for such problems when the predicted output is a continuous numerical value as in our case, are mainly supervised learning algorithms which are based on regression type techniques. We wanted also to observe the effectiveness of models based on non-linear regression, in order to see in that way, which ones responds better in such cases. Specifically, three machine learning regression algorithms have been employed, where each one of the algorithms estimated separately both (x, y) coordinates of each player again. It should be mentioned that both (x, y) coordinates, did not show some correlation between them. This means that each variable was affected by different factors and variables and for that reason we chose to proceed with the estimation of the missing positions using only univariate models.

3.1.4. RANDOM FOREST

The first approach under scope was a Random Forest regression algorithm. As first proposed, Random forests build multiple decision trees and merge their predictions together to get a more accurate and stable prediction rather than relying on individual decision trees (Ho, 1995). The samples

of the training observations, are drawn with replacement, known as bootstrapping, which means that some samples will be used multiple times in a single tree and a prediction is recorded for each sample. After all of the predictions have been assessed, the ensemble prediction is calculated by averaging the predictions of the above trees producing the final estimations.

For the implementation of the specific model, we kept only the initial actual values of each entity and omitted any extra values, which in fact did not offer something to our analysis. This step was necessary, in order to precisely estimate the prediction accuracies of our model. The most correlated variables for both (y, x) coordinates, were the "Frame" and "Timestamp" variables. In order to properly define the actual samples for both training and testing sets, we deemed appropriately to use as a training set each time, all the initial observable (k) values of the response variable (either y, x each time) and as a testing set all the missing values of the response variable, that were induced due to the censoring effect from the camera. Therefore, each time the splitting proportions between the training and testing sets, varied according to the missing proportions of each player. Afterwards, a decision tree was generated associated to these (k) data points, by defining also, the correct number of trees. For a new data point, we made each one of the defined trees, predict the value of y and x as the case may be, for the data point in question and assigned the new data point to the average across all the predicted values of the response variable each time.

3.1.5. EXTREME GRADIENT BOOSTING

The second non-linear regression algorithm, was a stochastic gradient boosting algorithm, or otherwise called as Extreme Gradient Boosting algorithm. Each object treated as a unique entity again and the positional data (x, y coordinates) were separately calculated, by omitting all the extra values that were filled in the beginning and keep only the initial actual values. The most related variables for both response variables ((x, y) coordinates), were again the frame and timestamp of each player. For the training part, we used all the initial (k) observable values of the response variable and as a testing set all the missing information. A fraction of the training observations sampled, in order in each iteration a new tree generated from each subsample and any errors from the previous trees to be corrected. The training procedure that we followed, proceeded iteratively adding new trees for the prediction of the residuals of the errors of prior trees and then combined with previous trees to make the final prediction for our response variable. Finally, extra tuning parameters calculated at each iteration, in order to define the best selected model and at the end. RMSE was used to select the optimal model using the smallest value.

3.1.6. K-NEAREST NEIGHBORS REGRESSION

The third non-linear regression algorithm that we used, was the Nearest-Neighbor Regression. The idea is to identify from the training set observations with regressors similar to the one we want to predict. Similarity is based on some distance, Euclidean in the standardized variables in our case. Then a weighted average of the k-nearest observations is obtained as the prediction. In the training phase of the algorithm, only the features are stored, by defining appropriately each time the optimal k value, which determines the number of neighbors we look at when we assign a value to any new observation. Regarding the modelling phase, the calculation of the most related variables, the preprocessing splitting procedures and the omission of the "extra" values have been commonly

implemented as the previous two non-linear algorithms. For each object the size of the neighborhood was calculated by using a heuristically optimal number *k* based on the RMSE and a cross-validation technique in order to select the value of *k* that minimizes the mean squared error. The Euclidean distance computed in order to find the corresponding *k* number of neighbors each time and the distance was increased by ordering the labeled examples. An inverse distance weighted average was also estimated with a number of k-nearest multivariate neighbors. Finally, for the evaluation purposes, the last optimal *k* RMSE value was selected in the last iteration of the whole mentioned procedure, using the smallest possible value.

3.1.7. TIME SERIES METHODS

On the grounds that, our data are in fact time series data, we wanted also to observe whether a time series model, could also respond to the present problem and how well compared to the other executed methods. As a first step, we create twenty-nine (29) different time series objects, in order to observe possible various patterns of each object (22 starting lineup players, 6 substitute players and the ball). Each object as it is reasonable, was consisted from a different number of observations, due to the fact that each player was observed at different time instances from the camera. We deemed appropriate also, to observe the different characteristics of each one of the different objects, by examining their trend, seasonality and random components for each object's coordinates. A specific trend and seasonality, did not exist for the majority of objects, comprising mainly of random noise, as the observations were either decreasing or increasing depending on the position, speed and acceleration of the object. In order to stabilize variance, we used the technique of the kernel smoothing, by appropriately defining previously the optimal corresponding bandwidth. We also extracted the logarithmic values for both (x, y) coordinates in order to observe possible differences and we differenced as well for the integration part the logarithmic values to turn the data into its stationary form.

As the coordinates of the objects were inextricably linked also with future values, we reduced the sample size to achieve stationarity of each object. For the autocorrelation and partial autocorrelation part, the majority of players presented a strong positive autocorrelation, decreasing slowly over time and thus indicating that each object, could not achieve absolute stationarity. This can be explained due to the structural format of the data. After all the preliminary checks, we defined an autoregressive integrated moving average (ARIMA) model for every object on the pitch, by including a necessary differencing step to eliminate the non-stationarity. Regarding the technical implementation of each ARIMA model, we employed for the training part of the model, the previous ten (10) observations of the y or x coordinate as the case may be, to be fitted in the model and predict accordingly the following ten (10) observations of every single object. The constant term defined as the average period-to-period change in *Y*. The number of orders changed every time a new object was selected, by including also a linear drift term each time a model was running. The frequency parameter, was equal to the number of tenths of seconds observed at each training subsample. If a missing value was observed, the model imputed the specific value with the corresponding prediction and continued the whole process, by dividing the training part into small subsamples of ten (10) corresponding observations, up until a player appeared on the pitch again. Adding covariates is also an option but this creates estimation problems and improved very little the results.

Finally note that the three interpolation methods are applied to the position data only. The three supervised methods are using covariate information while the time series approach is based on solely the past observation of the relevant variable. Also, we emphasize that we have used other methods also but for saving space we do not present their results since they were inferior.

3.2 METRICS USED

In order to compare the different methods, we employed several metrics and criteria. Each assessment criterion was computed separately for each object. The first metric used, was the root mean square error (RMSE). In our case, let y_i denote the i^{th} observation of the actual coordinate of a player, $\hat{y}_i$ the imputed value and n the number of missing values of the specific coordinate. Then, the RMSE is given by:

$$RMSE(\hat{y}, y) = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2} \tag{3}$$

Another measure is the mean absolute percentage error (MAPE), which is a measure of prediction accuracy. MAPE is commonly used as a loss function for regression problems, due to its very intuitive interpretation in terms of relative error. MAPE is given by:

$$MAPE(\hat{y}, y) = 100\frac{\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|}{n} \tag{4}$$

We have also employed the typical Pearson correlation between the imputed and the original values (PMCC). We also use Cohen's h, which is a measure of distance between two proportions of probabilities, where for $h = 0.2$ implies a small difference, $h = 0.5$ a medium difference and $h = 0.8$ implies a large difference (Cohen, 1960). Cohen's h metric is defined as:

$$h(\hat{y}, y) = \frac{M_y - M_{\hat{y}}}{SD_{y\hat{y}}}, \qquad SD_{y\hat{y}} = \sqrt{\frac{SD_y^2 + SD_{\hat{y}}^2}{2}} \tag{5}$$

where M_y and $M_{\hat{y}}$ are the sample means of y and $\hat{y}$ respectively and SD_y and $SD_{\hat{y}}$ the respective standard deviations.

Finally, as an extra evaluation criterion for accessing the accuracy between the actual and the interpolated observations, we compared also the average distances per minute among all players. The actual and imputed average distances were calculated by the Euclidean distance as:

$$d_{avg(actual)} = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2}, \quad \hat{d}_{avg(imputed)} = \sqrt{(\hat{y}_i - \hat{y}_j)^2 + (\hat{x}_i - \hat{x}_j)^2} \tag{6}$$

In order also to properly define the accuracy effect among the actual and interpolated distances, we calculated the RMSE and MAPE between the distances. In our case let x_i, y_i denote the i^{th}observation of the actual x, y coordinates of a player, and x_j, y_j the j^{th} observation of the actual x, y coordinates of another player. In addition, let $\hat{x}_i, \hat{y}_i$ denote the interpolated values of the missing coordinates happened in the same time instance of a player and $\hat{x}_j, \hat{y}_j$ the interpolated coordinates of another player as well. The corresponding error metrics after the calculation of the average distances per minute among the players, were the following:

$$RMSE\left(\hat{d}_{imputed}, d_{actual}\right) = \sqrt{\frac{\sum_{t=1}^{n}\left(\hat{d}_{imputed} - d_{actual}\right)^2}{n}} \quad (7)$$

$$MAPE\left(\hat{d}_{imputed}, d_{actual}\right) = 100\frac{\sum_{t=1}^{n}\left|\frac{\hat{d}_{imputed} - d_{actual}}{d_{actual}}\right|}{n} \quad (8)$$

where $\hat{d}_{imputed}$ denotes the average distance per minute between each object after the necessary interpolation procedures, d_{actual} the actual average distances per minute among each object and n denotes the complete data points.

4. RESULTS

4.1 COMPARING THE DIFFERENT METHODS

After discussing the proposed methodology, we can present at this point all of our findings and make the necessary comparisons between the evaluation metric results of each separate algorithm. Our experiment was as follows: For each player we used the observed data, i.e. all the time points that their position was available. Then we randomly selected a number of points to be considered as missing and we applied the different approaches to recover this together with the positions that they were missing in the data set. The reported metrics are based on the comparison of the actual available methods that we tried to predict and the ones derived from each methodology. For each player the proportion of data that we hide for the experiment was proportional to the observed missing proportion we had.

Looking the results from Table 1, regarding the interpolation algorithms, all of them presented satisfactory results, with the best performing algorithms for all metrics, to be the Stine interpolation, followed by the Linear interpolation. For all the evaluation metrics, the densities concerning the errors between the actual and the imputed values are smaller than the other methods. Both samples concerning the actual and imputed values, seem to present great similarities and very small differences (value < 0.2). It is worth mentioning, that all algorithms perform better, in lower rates of missingness but in higher rates the results were also very satisfactory.

Table 1: Evaluation metrics derived from the different imputation algorithms

	RMSE		MAPE		Pearson Correlation		Cohen	
	y	x	y	x	y	x	y	x
Interpolation								
Linear	0.207	0.165	0.297	0.310	0.999	0.999	0.172	0.167
Spline	0.247	0.279	0.271	0.318	0.999	0.999	0.180	0.153
Stine	0.198	0.158	0.259	0.287	0.999	0.999	0.172	0.166
Supervised Methods								
k-NN Regression	0.500	0.547	1.069	1.184	0.995	0.999	0.172	0.165
Random Forest Regression	1.230	1.883	3.179	4.028	0.992	0.997	0.169	0.157
Extreme Gradient Boosting Regression	2.513	3.216	9.257	8.521	0.978	0.991	0.162	0.165
Time Series								
ARIMA	2.062	2.421	7.089	7.885	0.953	0.991	0.213	0.170

According to the three (3) mentioned supervised methods, namely Random Forest and Extreme Gradient Boosting algorithms, presented mediocre results, with big variations concerning the differences between the actual and the predicted values. Random Forest behaved better on cases with lower missing rates, while the Extreme Gradient Boosting exactly the opposite. On the other hand, k-NN regressor presented very good results and quite close as the corresponding results derived from the Stine and Linear interpolation algorithms. This seems logical as the k-NN algorithm assumes the similarity and as an extension the prediction of the most common neighbors of a missing observation. The similarities between distances of the actual and the predicted values were very high, indicating that the specific model has a fairly high predictive efficiency as well. Regarding the evaluation results of the time series forecasting implementation, fluctuated at the same levels as the evaluation results of the Random Forest and the Extreme Gradient Boosting algorithms. It handled more efficiently, players with lower missingness rates and with a higher speed and acceleration. Finally, the similarities between the distances of the actual and predicted values were lower in comparison with Stine, Linear and k-NN models.

Recapitulating, the best algorithms were the interpolation algorithms using the Stine interpolation and the k-Nearest-Neighbor regressor, as both models indicate the lowest RMSE and MAPE values and with the smallest dispersion in their predictions between the actual and the predicted values. Afterwards, the ARIMA and the Random Forest algorithms seem to fluctuate at the same levels, while the Extreme Gradient Boosting algorithm, seems to present a higher variability concerning both evaluation metrics and as a result it cannot be considered as a suitable approach (Table 1). Regression type models fail to account for the temporal effect and thus they provide worse

results. K-nn regression is the only one that can consider such effects by selecting points close to the time points under examination.

Figure 3 presents the average distances per minute among the players comparing the actual values and the interpolated ones. We plot only selected methods to save space. The Stine interpolation and k-NN regression derived the best results.

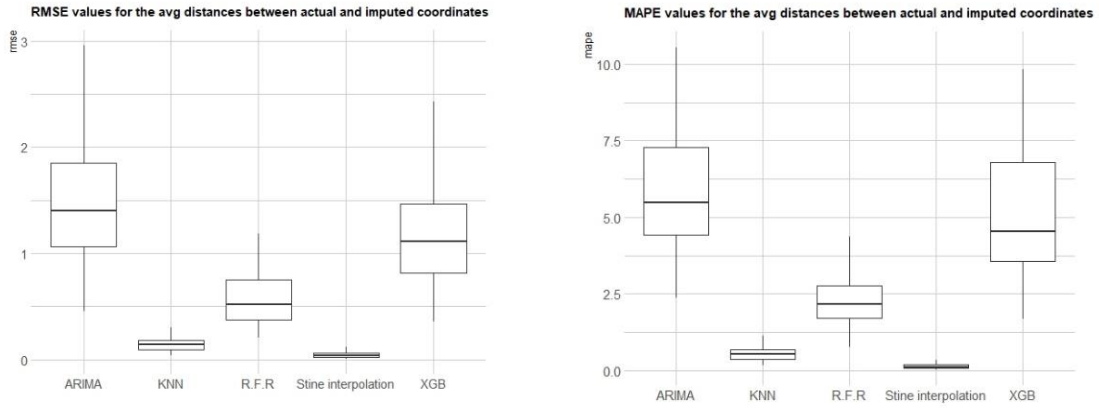
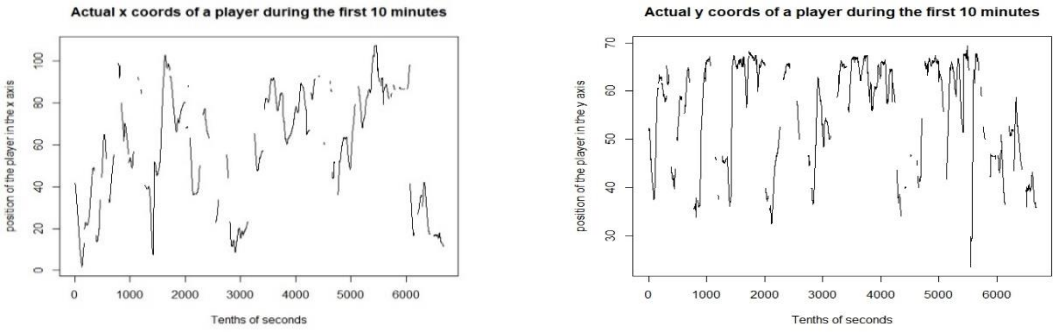


Figure 3: RMSE and MAPE results for the average distances per minute among players derived from each method

In conclusion, the most efficient algorithm is considered to be the interpolation algorithm using the Stine approach, as the predicted values of both x, y coordinates are very close to the actual ones. Therefore, it is proposed to be used when the censoring effect induced from the camera is present either for higher either for lower missing rates of objects. However, it must be stressed that all algorithms present satisfactory results as the values of the RMSE metric yielded an average of 0.5 - 2.4 (except from the XGB algorithm), indicating a margin error of approximately 0.5 ~ 2.4 meters prediction of a player's actual position. An example of the positional values of a single player for a time interval of 10 minutes after the interpolation procedures using the Stine method, can be observed below (Figure 4). The upper two (2) line plots represent the actual position values in both of his x, y axes on the pitch, while the tow (2) line plots below constitute the corresponding values after the interpolation.



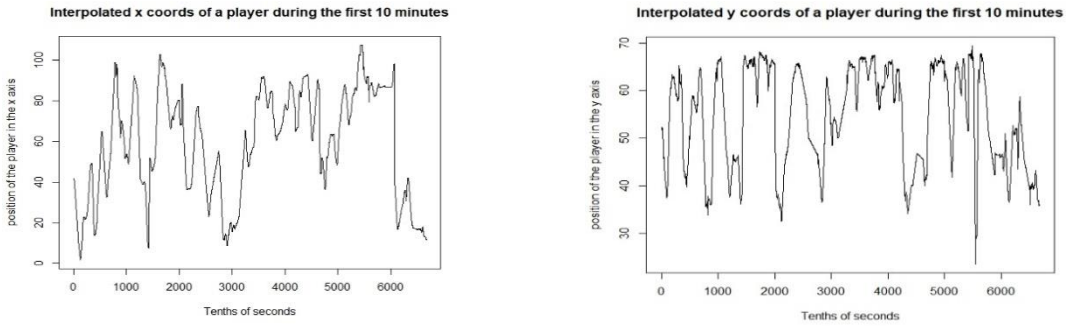


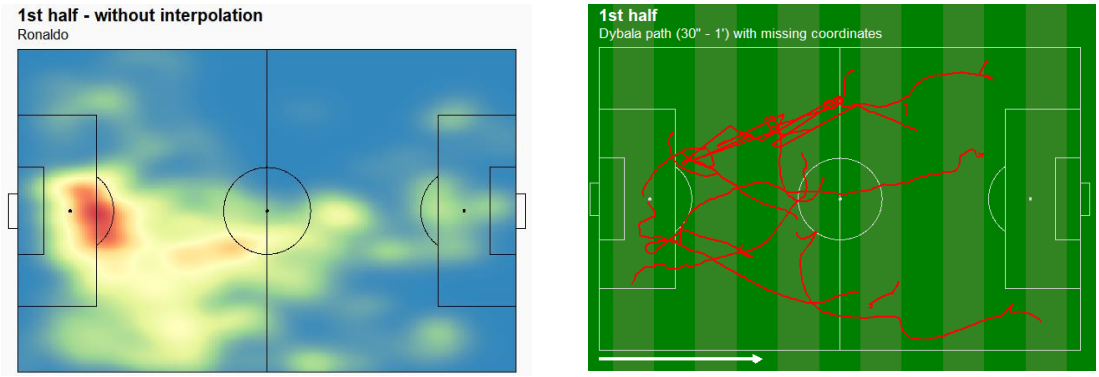
Figure 4: Positional values of a player for the first 10 minutes of the game after the interpolation procedures

4.2 APPLICATION OF INTERPOLATED DATA

In order to have also a more concrete view about the usefulness of this approach, we proceeded to some comparisons between the usage of the full information generated using the interpolated coordinates in relation with the actual data as they have been recorded from the broadcast camera. We provide some examples based on typical metrics based on tracking data.

4.2.1. HEATMAPS

A representative example can be considered for cases like players heatmaps showing the appearance of the players in different parts of the pitch and occasions like the players' soccer paths during a certain period of time (Figure 5). In the left-hand side one can see the heatmap for Ronaldo (Juventus) based on the actual available data (top) and the same for when the missing data have been imputed (interpolated). One can see a very close agreement between the two plots. At the right part of the figure one can see the observed paths of Dybala for the same match (top) and the paths when missing data are interpolated. Both cases show a large agreement.



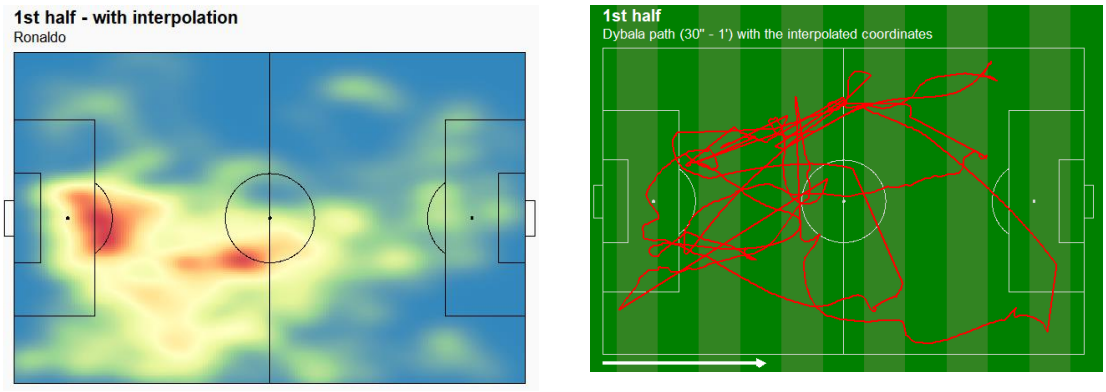
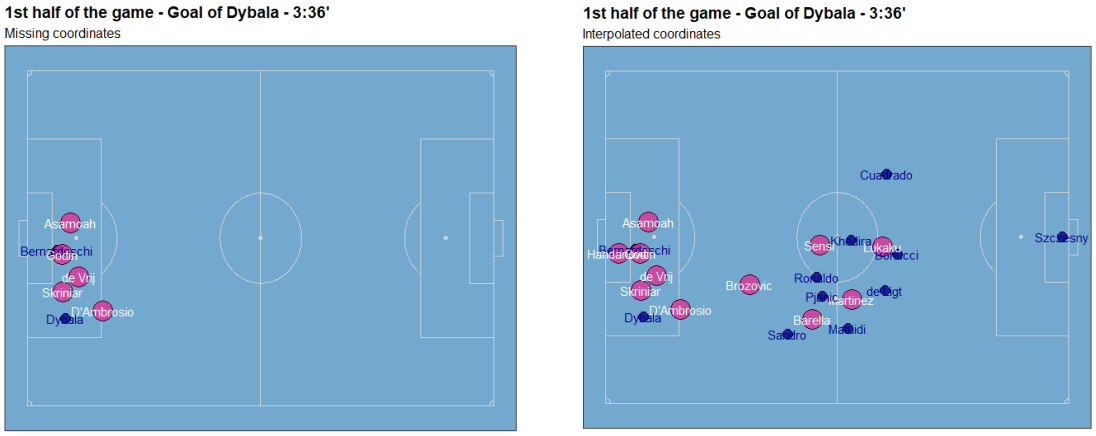


Figure 5: Football heatmaps showing the effectiveness of a player in different parts of the pitch and player's soccer path during a certain time instance

4.2.2. VORONOI DIAGRAMS

In mathematics, a Voronoi diagram is a partitioning of a plane into regions based on distance to points in a specific subset of the plane. These objects, which are mainly called as seeds of the plane, are just finitely many points in the plane. For each seed of the plane, there is an appropriate region, called a Voronoi cell, consisting of all points of the plane closer to that seed than to any other. The Voronoi diagram of a set of points is dual to its Delaunay triangulation. In football, these seeds represent the different positions of the players and the plane is referring to the instance of the field. A Voronoi diagram actually indicates the space that each player controls in the sense that if the ball is in this field then this player is closer to it. Hence missing data prohibit the creation of such plots. Imputed positions of the players can help better represent such situations. In Figure 6 one can see such a plot for a particular time stamp from the match considered. In Figure 6, one can see at the left part the Voronoi diagram from the available data and the right part based on the data after the imputation.



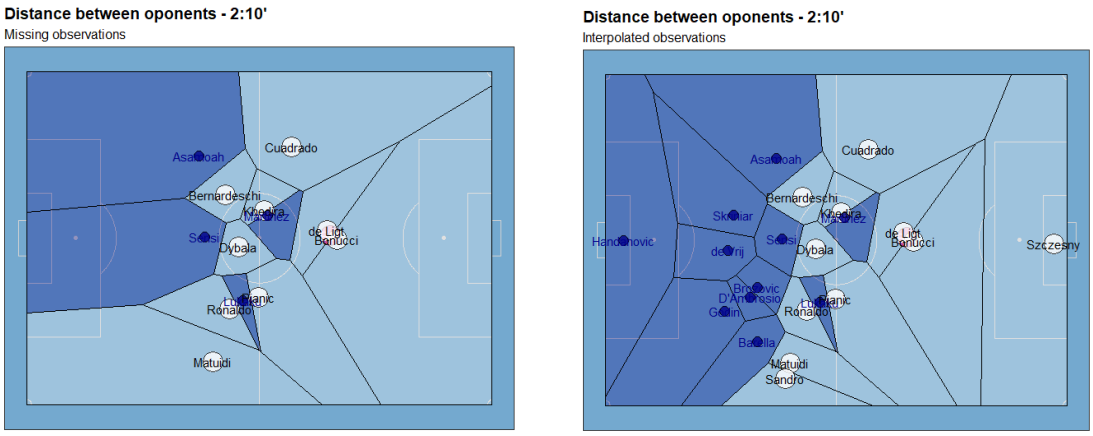


Figure 6: Players movement in real time

4.2.3. AVERAGE FORMATION LINE

A valuable match-level metric also, that is frequently monitored after post-match analyses, concerns the average formation line (AFL) of the teams. This metric corresponds to the average position on the long axis from the own goal of a team to the opponent's goal of the other team including all team events with the ball. By using this indicator, teams can better understand time intervals where their team put more pressure on the opposing team and instances when the team was in a more defending role. Figure 7 presents the AFL of the two teams. The left plot uses only the available players and hence estimates badly the AFL since some players are missing. The right panel is based on interpolated data using the Stine approach. Regarding the differences generated, it is clear that by using the initial data, the home team during the 1st half presents a more attacking formation system, while in the 2nd plot) does not seem to be so much forward to the opponent's goal. On the other hand, as far as it concerns the 2nd half, the home team based on the interpolated data seems to have a more attacking behavior, while regarding the plot with the actual data it reflects a more conservative formation, as it seems to be playing quite further back.

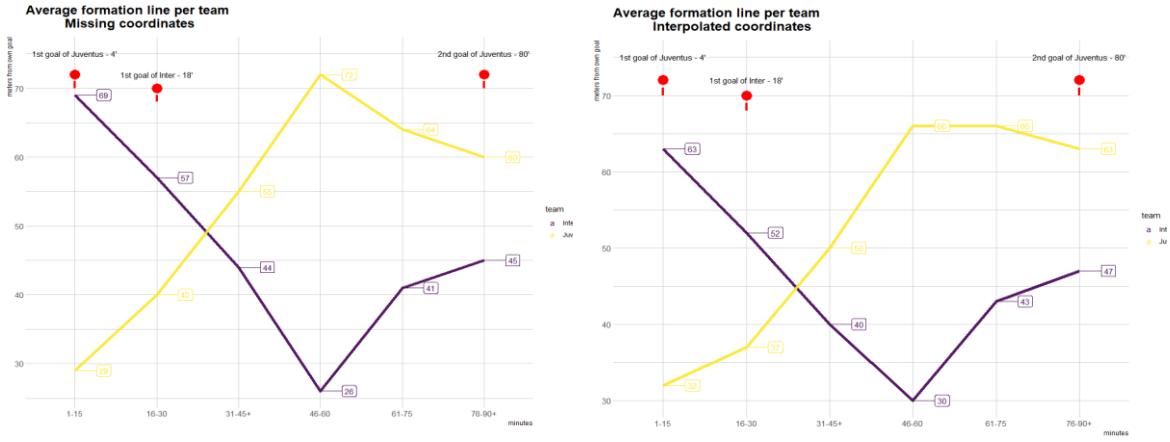


Figure 7: Average formation line per team using both actual and interpolated data

4.2.4. TEAM COMPACTNESS

Team compactness also, is another space-related concept that highlights the distances between one team's players (Santos and Penas, 2019). The idea is that players from one team keeping the biggest possible distance between themselves maintaining links between each other that will keep them in control of the space they occupy and action inside their occupied structure. Seven (7) key metrics were employed for both positional defensive and attacking plays where among others are included the average last defender's and first pressing player's lines, the average left and right lines, the average teams' depth and width and the defensive/attacking squares. By observing the defending and attacking squares of each approach (Figure 8), it is clear that for both positional plays, the home team has a more compact system when the interpolated coordinates are employed. By using the complete dataset also, the home team presents a more stable shape and secure space-control system, with narrowly defensive and attacking plays and a bigger concentration of the players in the inside zones of the pitch.

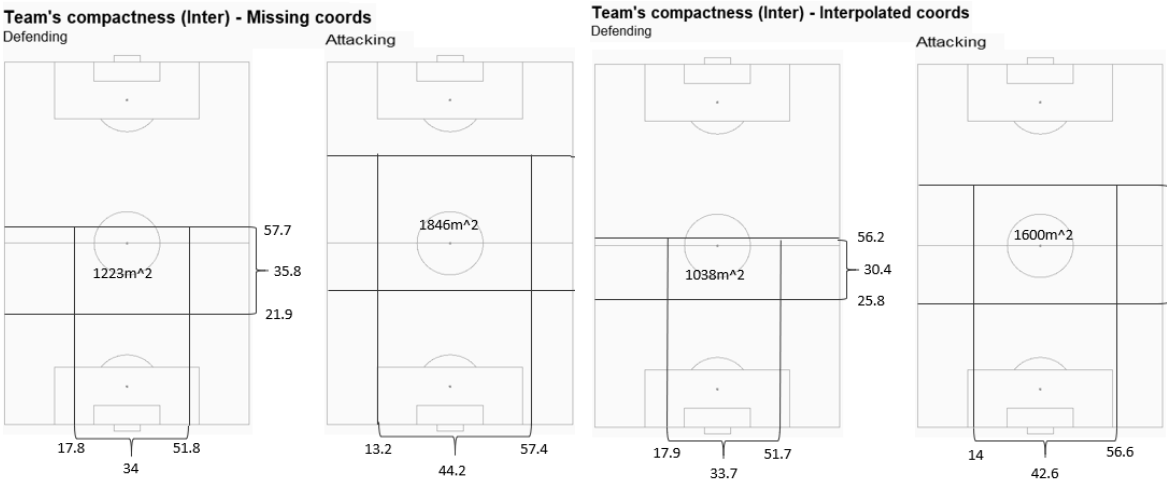


Figure 8: Home team's compactness using both actual positional before and after the interpolation procedure

5. CONCLUDING REMARKS AND FURTHER RESEARCH

In this paper we examined the case when tracking data based on broadcasting were used. Contrary to tracking data from a multicamera system, this is a cheap alternative producing less data. The scope of this paper was to examine whether interpolation methods can retrieve the missing information. The findings are very promising in the sense that the use of existing interpolation methods provided very satisfactory results, while several metrics used in practice calculated from the interpolated data was shown to have quite good performance in comparison with full data cases. Of course, this is rather a first step towards this research problem.

Some limitations of our approaches are also present. To start with, we have used only one match as a proof-of-concept case so a more detailed examination would have been much more insightful. Also note that we base our approach in existing interpolation methods which perhaps do not take in full account the particular data at hand. So, an interesting problem is whether better and more suitable for the problem interpolation is possible. Also, the missingness mechanism needs further investigation. Here implicitly we assume missing at random mechanism but one may argue that

players further away from the ball are more probably missing, so the mechanism can be somewhat informative. This deserves some more investigation. Finally, note that it would have been interesting to measure the effect of the interpolation to other metrics used in football based on tracking data. In this paper while we demonstrated the potential with some of them, it is perhaps the case that for other metrics more sophisticated interpolation may be needed.

To sum up, we believe that since broadcast-based tracking data are simple to obtain and very cheap their usage will be of increasing interest in the near future and thus we expect increasing interest and perhaps improved methodologies to obtain and analyze them.

ACKNOWLEDGMENTS

The authors would like to thank two anonymous referees for detailed comments that helped us improving the manuscript.

REFERENCES

Altman, N. S. (1992). "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression." *The American Statistician*, 46(3), 175–185.

Bialkowski A., Lucey P., Carr P., Matthews I. (2016). Discovering Team Structures in Soccer from Spatiotemporal Data. *IEEE Transactions on Knowledge and Data Engineering* 28(10), 2596-2605

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20, 37–46.

Diquigiovanni J., Scarpa B. (2018). Analysis of association football playing styles: An innovative method to cluster networks. *Statistical Modelling* 19(1) 28–54.

Dvorak J., Junge A., Chomiak J., Graf-Baumann T., Peterson L., Rösch D., Hodgson R. (2000). Risk factor analysis for injuries in football players. Possibilities for a prevention program. *American Journal of Sports Medicine*. 2000;28(5 Suppl): S69-74.

Grothendieck G., Zeileis A. (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software* 14(i06).

Horton M., Gudmundsson J., Chawla S., Estephan J. (2014). Automated Classification of Passing in Football. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp 319-330, Springer, Cham.

Ho, T.K. (1995). "Random decision forests", In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pp. 278-282 vol.1.

Karlis D., Ntzoufras I. (2003). Analysis of sports data by using bivariate Poisson models. *Journal of the Royal Statistical Society:* Series D (The Statistician), 52(3), 381-393.

Link D., Lang S., Seidenschwarz P. (2016). Real Time Quantification of Dangerousity in Football Using Spatiotemporal Tracking Data. *PLoS ONE* 11(12): e0168768.

Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 35(7), 1704-1716

Mohr M., Krustrup P., Bangsbo J. (2003). Match performance of high-standard soccer players with special reference to development of fatigue. *Journal of Sports Sciences* 21(7):519-28.

Moritz S., Beielstein T.B. (2017). imputeTS: Time Series Missing Value Imputation in R. *The R Journal* 9(1), 207.

Mortensen, J. (2020). Statistical methods for tracking data in sports. *Doctoral dissertation, Science: Department of Statistics and Actuarial Science, Simon Fraser University, Canada.*

Power P., Ruiz H., Wei X., Lucey P. (2017). Not All Passes Are Created Equal: Objectively Measuring the Risk and Reward of Passes in Soccer from Tracking Data. *The 23rd ACM SIGKDD International Conference.* Paper in Conference Proceedings (pp. 1605-1613).

Rampinini E., Coutts A. J., Castagna C., Sassi E., Impellizzeri F. M. (2007). Variation in Top Level Soccer Match Performance. *International Journal of Sports Medicine*, 28(12), 1018-1024.

Rein R., Memmert D. (2016). Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus* 5(1) 1-13.

Santos P. M., Penas C. L. (2019). Defensive positioning on the pitch in relation with situational variables of a professional football team during regaining possession. *Human Movement* 20(2):50-56.

Shaw L., Glickman M. (2019). Dynamic analysis of team strategy in professional football. *Barca Sports Analytics Summit 13.* Paper in Conference Proceedings

Stineman, R. W. (1980). A consistently well-behaved method of interpolation. *Creative Computing*, 6(7), 54-57.

https://www.skillcorner.com Internet Document

https://uppsala.instructure.com/courses/28112 Internet Document

https://medium.com/skillcorner/a-new-world-of-performance-insight-from-video-tracking-technology-f0d7c0deb767 Internet Document

https://sportlogiq.com/en/ Internet Document