ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Master Thesis Business Analytics and Quantitative Marketing

The relation between network metrics and team performance in football.

Name Student:	Eveline Mathol	Supervisor:	dr. P Wan
Student ID number:	387097	Second assessor:	dr. M van de Velden

July 2020

Abstract

In this paper we investigate the relation between network metrics and team performance. We also investigate whether network metrics have predictive power for predicting the number of goals scored and match outcome. Several regression and classification methods are implemented excluding and including the network metrics as predictor variables. These include the Random Forest, Extreme Gradient Boosting, Support Vector Machine (SVM) and Lasso Regression. We use the largest open collection football dataset provided by Wyscout in combination with collected features. Matches from the Premier League 2017-18 season are analysed. Network metrics appear to be related to team performance. Teams with a better performance have in general greater values for the clustering coefficient, largest eigenvalue, algebraic connectivity, position of x, mean degree, average change in x and closeness score, and smaller values for the average shortest path. Incorporating network metrics does not improve predictive performance much. For the prediction of the number of goals scored the Random Forest provides the best results, with a Mean Absolute Error of 0.887 and 0.882 excluding and including network metrics, respectively. The SVM has the best predictive performance for the match outcome with an accuracy of 0.567 and 0.578 excluding and including the network metrics. It still seems challenging to predict football match outcomes. We believe that network analysis can be useful to gain insights in more complex team behaviour and interactions. The knowledge on the relation between network metrics and team performance allows football professionals to adapt their team tactics for optimizing performance.

Keywords: football, passing networks, network metrics, team performance, Premier League, football match prediction, machine learning, random forest, support vector machine (SVM), gradient boosting.

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Contents

1 Introduction

Football is considered the most popular sport in the world. Over 250 million people regularly play football world-wide, 43% of the population consider themselves football fans and football has the largest television viewership (Nielsen Sports, 2018). Over the past years football analytics also gained more popularity. Advancements in big data and technology, such as automated sensing software, enabled high-fidelity data streams to be extracted for a match (Pappalardo et al., 2019b). The wealth of football data is enormous and offers many possibilities for analysis.

Football is a team sport with a highly variable and unpredictable nature. The performance of Leicester City in the Premier League 2015-16 season strongly emphasises this. After being promoted to the Premier League in 2014-15 and finishing 14th, Leicester City beat all odds and became the winner of the 2015-16 season. Football analytics is considered a complex task; global behaviour depends on dynamics of the interactions between two competing teams, constantly influencing each other (Cintia et al., 2015). Also, football is a low scoring sport (Duch et al., 2010). One of the teams can be significantly better, but the match can still end up in a draw. Performing football analysis can give insight in the team performance and their interactions. Recently, professional firms started to perform football analysis (Cintia et al., 2015). Important use cases include the construction of betting odds and the identification of talented young players by football scouts.

Despite the growing popularity of football analytics and the wealth of football data, the use of advanced performance metrics was still limited. Analysis was mostly restricted to simple metrics, including ball possession, pass accuracy and the number of shots on goal. More recently, network science is used in sports (Buldu et al., 2019). Network science is the study of connections or relationships between elements of phenomena, such biological and social phenomena (Watts, 2004). The connections between the elements can be represented as networks. Network science allows for studying more complex team interactions and behaviour in football. Using network science, passing networks for football matches can be created (Duch et al., 2010). Here, nodes represent the players of a team and edges the number of passes between the players. Network analysis enables among others the identification of key players, study of interactions between players and investigation of more complex network behaviour of a team (Rein and Memmert, 2016). Also, research showed that metrics coming from network science can be related to team performance (Pena and Touchette, 2012);(Clemente et al., 2015b);(Buldu et al., 2019).

In this paper we use network analysis to study the behaviour of football teams. The main goal is to investigate whether network metrics are related to the performance of teams. Passing networks are created for all matches in the Premier League 2017-18 season. Network metrics on player prominence, interconnectivity and spatial properties of the teams are studied. The analysis includes comparing network metrics according to match outcome, studying correlations between network metrics and goal statistics and identifying teams with similar network behaviour. Besides exploratory analysis, we also investigate whether network metrics have predictive power for match outcome. To do this, we implement several regression and classification models for the prediction of both the number of goals scored per team and match outcome.

Compared to previous research on using network metrics for football analysis and on the prediction of football match outcomes, this paper contributes to this field of research for the following reasons: i). it considers a large amount of network metrics; ii). it uses the largest open collection football dataset with an enormous amount of features; iii). it is the first to exploit the use of network metrics as predictor variables; and iv). it compares a wide range of regression and classification models for the prediction of football match outcomes.

This research is organized as follows. In Section 2 we give an overview of the existing literature on network analysis and predictive models in football. Section 3 consists of the data description, including overviews of the extracted features that will be used as predictors. In Section 4, the exploratory analysis including the comparison of network metrics and correlation analysis is described. Section 5 provides an overview of the implemented regression and classification methods for the predictive analysis. Section 6 and 7 present the results and discussion for the exploratory analysis. Section 8 and 9 present the results for the implemented predictive models and discusses their performances. We finish with limitations of this research in Section 10 and a conclusion in Section 11.

2 Literature

In this section we describe literature on network analysis used in football, with the focus on network metrics. After this, we provide an overview of literature that focused on the modelling of the number of goals scored and match outcome.

2.1 Network analysis

Sports analytics has been popular for a long time, but the application of network science to sports, and specifically to football, has only been widely studied since the previous decade (Buldu et al., 2019). First approaches focused on simple football statistics, such as pass accuracy and ball possession. These approaches are unable to capture individual performance, more complex dynamics and processes underlying team tactical behavior (Rein and Memmert, 2016). Using network science, the organization of a team can be considered as the result of the interaction between the players, creating passing networks (Buldú et al., 2018). With the network metrics derived from the passing networks, more complex team behaviour can be studied.

The first attempt to perform network analysis for football matches is performed by Gould and Gatrell (1979). For the FA Cup final between Liverpool and Manchester United in 1977, passing networks were created. Only in 2010, Duch et al. (2010) again considered passing networks. They introduced the *flow centrality* metric for the quantification of individual players. This metric allows for assessing the individual contribution of a player to team performance, which is considered challenging in football analytics. Gama et al. (2014) further explored passing networks, for identifying key players in attack phases and establishing preferential linkages between players.

Different types of network metrics were introduced and studied by researchers. Clemente et al. (2015a) proposed a set of network metrics for the studying of team properties and cooperation during attack phases. For five Portuguese League matches, the *network density*, *network heterogeneity* and *network centralisation* were analysed for identifying the strength and type of interactions between players. One finding is that the metrics were higher for the second half, showing decreased participation of the players. Cintia et al. (2015) developed the *H indicator*, a metric based on a set of pass-based performance metrics. They found that the H indicator has a strong correlation with the success of a team. Gonçalves et al. (2017) explored the relation between passing networks and the match outcome in youth elite association football in Portugal. They studied the *closeness* and *betweenness centrality* and showed that a lower passing dependency for a player (low betweenness score) and high intra-team well-connected passing relations (high closeness score) may optimize team performance. Aquino et al. (2019) examined among others the *clustering coefficient*, *eigenvector centrality*, *closeness centrality* and *betweenness centrality.* The metrics were compared for different match outcomes and playing formations using data of the FIFA 2018 World Cup. For match outcome, only the clustering coefficient seemed to have an effect: winning teams have in general larger clustering coefficients. Other papers investigating network metrics include the *in-degree* and *out-degree* for finding prominent tactical positions (Mendes et al., 2015); the *network density*, *total links* and *clustering coefficient* for comparing teams that performed better in the FIFA World Cup 2014 (Clemente et al., 2015b); the *clustering coefficient*, *network centroid*, *shortest path*, *algebraic connectivity* and *eigenvector centrality* to extract the unique style of F.C. Barcelona coached by Guardiola in the 2009-10 season (Buldu et al., 2019).

Although some papers relate network properties to match outcome, statistical modelling in this field is still rare. Wang et al. (2015) used a Bayesian latent model approach for automatically detecting tactical patterns of football teams. McHale and Relton (2018) implemented a generalised additive mixed model (GAMM) with covariates including position, distance and angle, to estimate the probability of a pass being successful. They combine this approach with the use of network centrality metrics to identify key players for the Premier League 2012-13 season. Buldú et al. (2018) provide a review on literature about passing networks and its challenges, including the dynamics, interaction between teams and time. For instance, to address the time challenge, some papers have investigated constructing passing networks with a sliding window instead of averaged over a match (Cotta et al., 2013);(Buldu et al., 2019). There appear to be many opportunities to extend current research and explore new methods, especially in combining modelling with the use of network metrics.

2.2 Modelling football match outcome

There are two distinct streams in the field of modelling football match outcomes: goal-based and result-based models. The first approach is focused on modelling the number of goals scored by both teams. The second approach models the probabilities of a win, draw or loss for the home team.

2.2.1 Statistical models

The first models developed for the prediction of the number of goals were statistical models. In particular, the number of goals was assumed to follow the Poisson distribution. The simplest case considers independence between the number of goals scored by competing teams. With this approach, the number of goals for the teams are based on two (conditionally) independent pairwise Poisson distributions. Early works using independent Poisson distributions include the work of Lee (1997) and Dyte and Clarke (2000). Lee (1997) expressed the mean as a linear combination of parameters for the home-team advantage, attack and defense strength. Dyte and Clarke (2000) modelled the number of goals conditional on the FIFA rating of each team and the match venue.

Although existence of correlation between the goals scored by competing teams had been proven, it was mostly ignored because of computational complexity. First approaches considering the dependence were proposed by Maher (1982), who implemented the bivariate Poisson distribution. This distribution accounts for (positive) dependencies between the number of goals scored by both teams. For further information on the bivariate Poisson distribution, we refer to the paper of Karlis and Ntzoufras (2003). Dixon and Coles (1997) extended the independent Poisson model of Maher (1982) by allowing among other things dynamic attack and defense parameters. They showed that two independent Poisson distributions do not correctly model the number of goals scored by both teams for all outcomes and include a correction parameter that adjusts the probabilities for interdependence. Rue and Salvesen (2000) incorporated the methods of Dixon and Coles (1997) and used a Bayesian framework to model the time-varying parameters. Groll et al. (2018a) provided evidence that instead of a bivariate Poisson distribution, two independent Poisson distributions can be used, if the two Poisson parameters contain enough informative covariates and thus already capture the correlation.

In the early 2000s, discrete choice models were developed to predict match outcome directly. Goddard (2005) compared the predictive performance of the goal-based and result-based approaches. Bivariate Poisson and ordered probit regressions were used for the modelling. The differences in predictive performance were relatively small, however, a hybrid method that combines a result-based dependent variable with goal-based team performance covariates provided the best results. Tsokos et al. (2019) used a Bradley-Terry model to predict match outcome. They considered various features, including days since previous match, points per match, and team rankings.

2.2.2 Machine learning models

In the beginning research focused mostly on the Poisson regression for the number of goals scored. More recently, machine learning approaches were investigated for both the prediction of football match outcomes and the number of goals. Schauberger and Groll (2018) implemented random forests and compared their predictive performance to more conventional regression methods. Using the data of the FIFA World Cup between 2002 and 2014, they modelled the number of goals and the match outcome, as well as a combination of both. For the match outcome, they also considered a variant of the random forest that takes into account the order of the outcomes. Covariates on economic factors of the country, team structure and home advantage were included. The random forests outperformed the conventional regression methods for the number of goals and ordinal match outcomes, and are close to or outperforming the predictions of bookmakers. Within the random forest methods, the forests that directly model the number of goals slightly outperformed those for the match outcomes. Groll et al. (2018a) further investigated the random forests on the same dataset. The methods are compared to ranking methods. In the end, including team ability parameters from the ranking methods as additional parameters to the random forests resulted in the best predictive performance.

Baboota and Kaur (2019) introduced another machine learning method for the prediction of football match results: gradient boosting trees. They compared this method to a Naive Bayes model, Support Vector Machine and random forest for the prediction of the match outcome. Features included covariates on the home and away form, and home and away streaks. The random forest and gradient boosting performed well, but were unable to outperform the bookmaker's predictions. Goller et al. (2018) focused on predicting the probabilities for the match outcome of the games of the German Football Bundesliga. They implemented a random forest that deals with the order of the match outcomes. A wide range of features is extracted from various data sources, including the market values, height and age of players, travel time, capacity of stadiums and information regarding European competitions.

The use of machine learning methods for the prediction of football match outcomes and the number of goals is still rare. Machine learning methods have just recently been implemented for this purpose. Also, the papers are quite limited in terms of features and datasets used for the modelling of match outcome and number of goals. There still seem to be many possibilities for exploring and extending machine learning methods and feature extraction.

3 Data

This section provides a detailed description on the data used for this research. First, we provide information about the available Wyscout datasets and collected data. After that, we present the extracted features from the data, separated in dependent variables and predictor variables. Tables with the different types of predictor variables are provided.

3.1 Wyscout dataset

The data used for this research is the largest open collection of football-logs ever released. A thorough data description is provided by Pappalardo et al. (2019b). The data has been collected and provided by Wyscout, an Italian football analysis company. Wyscout helps professionals to make data-driven decisions by providing tools for scouting, match and performance analysis. Clients of Wyscout include major football federations, scouting agencies and over 800 international clubs.

The data contains all spatio-temporal events occured during the matches from seven well-known football competitions. These competitions include five 2017-18 national football competitions in Europe: the Spanish first division (La Liga), Italian first division (Serie A), English first division (Premier League), German first division (Bundesliga) and French first division (Ligue 1). The other two competitions are the European Championship of 2016 and the World Cup of 2018, which are competitions between national teams. The number of matches, teams, players and events per competition are given in Table 1.

Table 1: Overview of the competitions and their number of matches, teams, players and events.

Competition	# Matches	# Teams	# Players	# Events
England	380	20	603	643,150
France	380	20	629	632,807
Germany	306	18	537	519,407
Italy	380	20	686	647,372
Spain	380	20	619	628,659
World Cup	64	32	736	101,759
European Championship	51	24	552	78,140

The data consists of seven datasets: competitions, matches, teams, players, events, coaches and referees. Each dataset is provided in Javascript Object Notation (JSON) format. The event dataset contains the most relevant information for this research. This data has been collected by video analysts using software that performs tagging (Pappalardo et al., 2019b). The tagging of a match consists of three main steps: 1) setting formations, 2) event tagging, and 3) quality control, which consists of an automatic check by an algorithm and a manual check by quality supervisors. The software in step 2 tags specific events in a match, and additional attributes to the event, such as the players performing the event, the exact location of the start and end of the event and tags about the accuracy of the event. As an example, we show a random observation of the event dataset below:

id	eventId	subeventId	tags	playerId	teamId	matchId	matchPeriod	positions	eventSec	eventName	subeventName
88178677	8	81	1801	83574	11944	1694390	1H	(13,31) (6,45)	37.14254	Pass	Hand pass

There are 10 types of events: duel, foul, free kick, goalkeeper leaving line, interruption, offside, others on the ball, pass, save attempt and shot. Table 2 gives insight on the proportions of the events, expressed in percentages of the total. The available subtypes and common tags per event are shown in Appendix A.

Table 2: Events distribution (% of the total) per competition.

Competition	Duel	Foul	Free Kick	Goalkeeper leaving line	Interruption	Offside	Others on the ball	Pass	Save attempt	Shot
England	27.47	1.27	5.66	0.20	4.28	0.24	7.94	51.10	0.52	1.31
France	27.08	1.61	6.15	0.19	4.40	0.24	8.03	50.44	0.54	1.32
Germany	27.76	1.67	5.98	0.18	4.18	0.23	7.79	50.34	0.54	1.33
Italy	25.92	1.54	5.87	0.17	4.17	0.25	8.06	52.11	0.55	1.36
Spain	27.37	1.74	6.06	0.20	4.14	0.30	7.69	50.70	0.54	1.27
World Cup	25.48	1.74	5.86	0.21	0	0.17	9.12	55.38	0.55	1.39
European Championship	27.00	1.70	6.20	0.22	0	0.24	6.55	55.92	0.63	1.53

From Table 2 we can see that the most frequently occurring event is a pass (over 50%), followed by a duel and others on the ball. The goalkeeper leaving line and an offside are in general the least occurring events.

The other datasets include information on the exact location of the competitions, matches, outcomes of the matches, substitutes, referees of the match and detailed information on the players, such as height, weight and age. An overview of the fields in the datasets is given in Appendix B. For a more detailed description of the datasets and their specific fields, we refer to the paper of Pappalardo et al. (2019b) and the Wyscout documentation at `https://apidocs.wyscout.com/`.

3.2 Collected data

Besides the information available in the Wyscout datasets, we collect other data that is relevant for this research. We are specifically considering factors that could influence team performance.

- **FIFA Rating** and **Wage** of the players. These covariates are present in the FIFA 18 Complete Player Dataset on `kaggle.com`, an online platform for machine learning and data science. The data has been crawled from the website `sofifa.com`. The rating of the players is created by EA Sports, the video game developer of the FIFA Football video games. The rating is based on a combination of international recognition and six scores for key statistics: pace, shooting, passing, dribbling, defending, and physical. Wage is expressed in euros per week. We believe that both the FIFA rating and the wage of the players in a team provide an indication of the strength and past performance of a team.

- **PlayeRank**. In addition to the overall FIFA rating, that is based on the half year before the start of the season, we add scores of the players during matches of the 2017-18 season. The PlayeRank score is a score between 0 and 1, with 0 indicating a terrible and 1 a very good performance (Pappalardo et al., 2019a). This score was created by the authors to create a widely accepted football performance metric for all its facets. With this score we are able to consider a more recent evaluation of the performance compared to the FIFA Rating and possibly identify temporary periods of performance.

- **European match**. Besides the national competitions of each country, some teams can also participate in European championships. The European championships considered here include the Champions League and Europa League. Teams qualify based on their ranking in the national competition and previous performance in the European championships. In general, the top one to four teams of a country can qualify. The Europa League is a ranking below the Champions League (UEFA, 2020). The matches of these championships take place during the regular national competitions. Thus, participating in these championships could affect the schedule and performance of the teams. We consider whether a team plays an European match in the week before or after the match. Schedules are taken from `worldfootball.net`

- **Ranking**. The ranking of a team is an indication of past performance and team strength. This dataset includes the ranking, points, and goal difference of the teams per matchweek. The information is taken from the official Premier League website `premierleague.com`.

- **Stadium capacity**. Data regarding the capacity of the stadiums is gathered from `worldfootbal.net`. Stadium capacity can have an impact on the atmosphere during a match. A greater amount of visitors may increase the home advantage. Also, teams with a larger stadium capacity will in general have more revenues and thus resources. By accounting for the ranking of teams, the stadium capacity allows for a 'big team' effect (Goddard, 2005). The big team will have a higher chance of winning, either because of the amount of visitors or because of the bigger amount of resources.

- **Distance stadiums**. As mentioned in the paper of Goddard (2005), geographic distance has a significant influence on match outcome. A small geographic distance could result in a greater competition between the teams because of a local derby, while a larger geographic distance can cause a home advantage due to travel difficulties and fatigue for players and supporters of the away team. Distances between the stadiums are calculated using the Google Maps API. This is available in the R package **gmapsdistance** (Melo et al., 2017). For the calculation of the distance, we consider transportation by car on a weekend day without any traffic.

3.3 Data selection and extraction

From the seven available competitions in the Wyscout dataset, we only select the competition of England: the Premier League season of 2017-18. This competition is considered the best league in the world in terms of popularity (highest television viewership) and competitive nature (many good teams instead of a few dominating teams). We believe that a similar approach for the modelling of match outcomes could be followed for the other competitions. The Premier League data has a total of 380 matches and 643,150 events.

Data features are extracted after combining the datasets shown in Appendix B. This is done by joining the datasets on the different IDs using SQL based commands. The collected datasets described in Section 3.2 also need to be joined. The dataset including the FIFA rating and wage requires the matching of the names of the players. When the names of the players are registered differently in the datasets, the matching cannot be done directly. To solve this we apply approximate string matching using Levenshtein distance (Levenshtein, 1966). The PlayeRanks data can be joined directly since it includes the match IDs. For the European match, ranking, stadium capacity and stadium distance datasets we join the datasets by manually adding the match, team, and stadium IDs.

The datasets provided by Wyscout and the collected data hardly contain any missing values. The final data created by joining the datasets has a total of 0.14% missing values, made up of 3.68% missing values for score attack and 4.74% missing values for the score defense from the PlayeRank dataset. Since the number of NA values is small, we replace these values using a simple median imputation.

3.4 Dependent variables

As described in Section 2.2, there are two different approaches for the modelling of football match outcomes: the modelling of the number of goals and the modelling of the match outcomes. We consider both approaches for the prediction of football match outcomes. In the following part, we briefly describe both dependent variables.

3.4.1 Goals

The modelling of the number of goals can be addressed as regression predictive modelling. Each match then corresponds to two different observations, one for each team. Figure 1 shows a barplot with the frequency of the number of goals scored by the home and away team. The number of goals is between zero and seven. In general the home team appears to score more goals. The home team has a higher frequency for two or more goals compared to the away team. The away team most frequently does not score any goals. There seems to be a home team advantage.

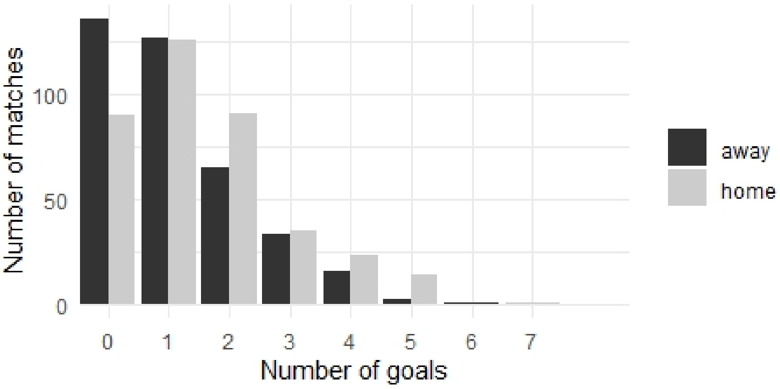


Figure 1: Frequency of number of goals scored by the home and away team.

3.4.2 Match outcome

For the second approach, the match outcome is used as dependent variable. There are three possible match outcomes: a win, loss, or draw for the home team. Since we consider the home team's perspective, each match corresponds to one observation. Table 3 shows the frequency of the match outcomes. There again seems to be a home team advantage.

Table 3: Match outcome distribution from the home team's perspective.

Win	Draw	Loss
164	96	100

3.5 Predictor variables

Different features are extracted from the combined datasets. These features are based on previous literature, common sense about effects on football performance and the investigation of the predictive power of network metrics. The tables in this section provide an overview of all extracted features categorized into different types. Besides the values of the extracted features, we also include the differences for all numeric variables between the competing teams as predictor variables, following the approach of (Groll et al., 2018b). The difference is taken from the perspective of the team under consideration for the prediction of the number of goals. For the prediction of match outcomes, the difference is taken from the home team's perspective. By doing this, we consider dynamics between the teams. For example, the variable for the difference in number of passes between the competing teams, assuming all other variables stay constant, describes the effect of a change in the number of passes of the opponent.

Table 4 includes the match-specific, team-specific, schedule-specific and location-specific features. The match-specific features contain a dummy for the home team to capture the home team advantage and a categorical variable for the referee. The team-specific features are only based on information about players that play during the match. Teams that are newly promoted are in general weaker than the other teams. Also, average wage and value of the players are measures for the team's strength. Day of the week and dummies for the teams playing in the European competition are included in the schedule-specific features. Playing in the European competition can affect the team's fatigue and mood. The location-specific features account for the fact that a large stadium capacity affects the atmosphere and that a large distance can cause travel difficulties.

Table 4: The short name, description and unit for the match-, team-, schedule- and location-specific features. Numerical features are displayed with their mean and standard deviation (SD), dummy features with their mean and categorical features with their median.

	Short name	Description	Unit	Mean (SD) or Median
Match-specific features				
1	Home	1 if the team is playing at home, 0 otherwise	dummy	0.5
2	Referee	Referee responsible for the match	categorical	M. Atkinson
Team-specific features				
3	Team	The team playing	categorical	-
4	Team opponent	The opponent of the team	categorical	-
5	Coach	The coach of the team	categorical	E. Howe & S. Dyche
6	Promoted	1 if the team was promoted to the league this season, 0 otherwise	dummy	0.15
7	Value	The average FIFA value of the players of the team playing in the match	numerical	77.51 (3.27)
8	Wage	The average wage of the players of the team playing in the match	numerical	78.70 (37.19)
9	Weight	The average weight of the players of the team playing in the match	numerical	77.27 (2.00)
10	Age	The average age of the players of the team playing in the match	numerical	27.23 (1.00)
11	Weight	The average weight of the players of the team playing in the match	numerical	182.78 (1.92)
12	Percentage left	Percentage of the players of the team playing in the match that are left-footed	numerical	0.25 (0.09)
13	Percentage native	Percentage of the players of the team playing in the match that are English	numerical	0.25 (0.13)
Schedule-specific features				
14	European match	1 if the team played an European match in the 7 days before the match	dummy	0.09
15	European match competition	1 if the opponent team played an European match in the 7 days before the match	dummy	0.09
16	Weekday	Day of the week	categorical	Saturday
Location-specific features				
17	Stadium capacity	Capacity of the stadium the match is played in	numerical	40487.75 (19870.79)
18	Distance stadiums	The distance between the stadiums of the team and the opponent team in kilometers	numerical	245.36 (138.50)

Table 5 contains features based on the previous three matches of a team and the rank of the team. Since the information from the match features is not known before the start of a match, these features are averaged over previous matches. By doing this, the features can be used in predictive analysis. This includes features on the attacking ability and defensive ability. Rank-specific features include the rank, points and goal difference of the team for the current matchweek.

Table 5: The short name, unit, mean and standard deviation (SD) of the features based on the previous matches and the rank-specific features.

	Short name	Unit	Mean (SD)
Previous match features			
1	Passes	numerical	423.44 (123.43)
2	Consecutive passes	numerical	271.07 (110.49)
3	Corners	numerical	5.16 (1.83)
4	Fouls	numerical	10.74 (2.16)
5	Shots	numerical	11.13 (3.62)
6	Yellow cards	numerical	1.58 (0.74)
7	Red cards	numerical	0.03 (0.10)
8	Score overall	numerical	0.01 (0.01)
9	Score attack	numerical	0.01 (0.01)
10	Score midfield	numerical	0.00 (0.01)
11	Score defense	numerical	0.01 (0.00)
12	Possesion	numerical	0.50 (0.06)
13	Freekicks	numerical	48.07 (4.81)
14	Offsides	numerical	2.06 (1.08)
15	Shots on goal	numerical	12.72 (2.45)
16	Save attempts	numerical	4.41 (1.68)
17	Saves	numerical	3.06 (1.32)
Rank-specific features			
18	Rank	numerical	10.50 (5.77)
19	Points	numerical	26.58 (18.10)
21	Goal difference	numerical	0.00 (18.94)

Finally, the network metrics displayed in Table 6 are features derived from the passing network. These metrics give insight in the interactions of a team and the team behaviour. We want to investigate whether these features have any predictive power for the modelling of the match outcome. Those metrics are also averaged over the past three matches, since they are not known before the start of a match. We again emphasise that for all numeric variables, both the value itself and the difference between the values of the competing teams are included as predictor variables.

After extracting all features, the final dataset contains 760 observations for the modelling of the number of goals scored by both teams and 380 observations for the modelling of match outcome. We will compare predictive models for the full set of predictor variables and excluding the network metrics. The number of predictors for these models are 73 and 119, respectively.

Table 6: The short name, unit, mean and standard deviation (SD) of the network features based on previous matches.

	Short name	Unit	Mean (SD)
Network features			
1	Clustering coefficient	numerical	3.42 (1.58)
2	Clustering coefficient local	numerical	0.30 (0.03)
3	Clustering coefficient global	numerical	0.83 (0.05)
4	Largest eigenvalue	numerical	29.02 (12.42)
5	Algebraic connectivity	numerical	6.74 (3.27)
6	Max eigencentrality	numerical	0.47 (0.03)
7	Sd eigencentrality	numerical	0.12 (0.01)
8	Mean eigencentrality	numerical	0.28 (0.01)
9	Closeness	numerical	0.18 (0.04)
10	Betweenness	numerical	8.34 (0.69)
11	Average shortest path	numerical	0.48 (0.16)
12	Position x	numerical	28.55 (4.34)
13	Position y	numerical	29.76 (4.02)
14	Dispersion	numerical	29.84 (0.87)
15	Dispersion position x	numerical	27.18 (1.08)
16	Average change in x	numerical	0.97 (0.51)
17	Average change in y	numerical	0.19 (0.43)
18	Closeness binary	numerical	0.08 (0.01)
19	Betweenness binary	numerical	3.19 (0.86)
20	Average shortest path binary	numerical	1.32 (0.09)
21	Mean degree	numerical	6.88 (0.77)
22	Sd degree	numerical	1.67 (0.21)
23	Max degree	numerical	9.16 (0.63)

4 Exploratory analysis

In this section we describe the exploratory analysis for investigating the relation between network metrics and team performance. We define the adjacency matrix, provide an overview of the studied network metrics, give information on the significance testing for comparisons between the network metrics and explain the correlation and cluster analysis.

4.1 Adjacency matrix

A passing network consists of nodes and edges, where nodes represent the players of the team and edges are weighted according to the number of passes occurring between the players. In this research, we only include successful passes for the weighted edges. A pass is considered successful when the end position of the pass of the sender is the same as the begin position of the event of the receiver following the pass. Unsuccessful passes are not included because we can not determine the intended receiver of the pass.

We study directed passing networks. Directed networks provide more information than undirected networks. They distinguish between performed and received passes by taking into acount the direction of a pass. The adjacency matrix is used to mathematically describe the passing networks. The adjacency matrix has size $n \times n$, where n refers to the number of players participating in a match. In this research we only consider sizes of 11 for easy comparison between networks. If a player is substituted, the new player takes over the node of the substituted player. The directed adjacency matrix is defined as follows:

$$A_{ij} = \begin{cases} \text{number of successful passes from player } i \text{ to player } j, & \text{for } i \neq j \\ 0, & \text{for } i{=}j. \end{cases} \tag{1}$$

The adjacency matrix can be used to gain insight in the interactions and performance of a team.

A passing network is made per match for each team. Thus, the network metrics under analysis are based on one match. We write a function in R for the creation of the adjacency matrix that has adjustable properties. It requires as input the match data, team, time period (first half, second half, whole match), a boolean for a directed passing network and a boolean for a weighted passing network. We also create visualizations of the passing network on a pitch, where the nodes and edges can be sized according to different network metrics.

4.2 Network metrics

Properties of the passing networks are studied using several network metrics. These metrics allow for analysing and quantifying the behaviour of football teams. In the end, we aim to find the relation between the network metrics and team performance. The network metrics can be divided into i). player prominence metrics; ii). interconnectivity metrics; and iii). spatial properties of the passing network.

4.2.1 Player prominence

To find the importance of players in the passing network, we study the out-degree and three different centrality scores for the players. The *out-degree* represents the total successful passed performed during the match per player. We include the mean, standard deviation and maximum value of the out-degree. The studied centrality scores are the eigenvector centrality, closeness centrality and betweenness centrality. The *eigenvector centrality* corresponds to the first eigenvector of the adjacency matrix. It is a measure of node importance; it takes into account the number of connections of a node and how well-connected the connections are. In our analysis, we include the mean, standard deviation and maximum of the eigenvector centrality scores. The *closeness centrality* considers how close a node is to other nodes in the passing network. It is the inverse sum of all distances to other nodes. Thus, for node v it is defined as

$$\frac{1}{\sum_i d(v,i)} \quad \text{for } i \neq v, \tag{2}$$

where $d(v,i)$ is the length of the shortest path from node i and node v. The *betweenness centrality* measures how well a node connects other players. It considers the number of shortest paths going through a node. The betweenness centrality for node v is defined as

$$\sum_{i \neq j \neq v} \frac{g_{ivj}}{g_{ij}} \quad \text{for } i \neq j \neq v, \tag{3}$$

where g_{ij} is the total number of shortest paths from node i to node j and g_{ivj} the number of those paths that pass through node v. Nodes with a high betweenness centrality score are important, since they connect other players through their shortest path (e.g. a player that connects the defense with the midfield) (Aquino et al., 2019). For the closeness and betweenness centrality, we only include the mean value over the players. We consider both the weighted and unweighted (binary) scores. For the unweighted scores, the distance of an edge is equal to one. For the weighted scores the distance is defined as the inverse of the number of passes, following the approach of Buldu et al. (2019). Thus, players that often pass to each other will have a shorter distance.

4.2.2 Interconnectivity

The interconnectivity metrics include the clustering coefficients, largest eigenvalue, algebraic connectivity and average shortest path. The *clustering coefficient* captures the degree to which players in a network tend to cluster together when passing the ball (Pena and Touchette, 2012). It reflects the local robustness of the network; when a player cannot access another player because of an opponent, there might be another player in the triangle for reaching the player. We consider the weighted, global and local clustering coefficient. The *weighted clustering coefficient* for node v proposed by Buldu et al. (2019) is defined as

$$\frac{\sum_{j,k} A_{vj} A_{jk} A_{vk}}{\sum_{j,k} A_{vj} A_{vk}}, \tag{4}$$

with A_{vj} the number of passes from player v to player j. The *global clustering coefficient* is the ratio of closed triplets (contains three edges) and all triplets (contains two or three edges). The *local clustering coefficient* of a node is the ratio of the number of edges between the neighbours of the node and the maximum possible number of edges between the neighbours of the node. The global and local clustering coefficient do not take into account the weights of the edges. The clustering coefficients are averaged over the players.

The *largest eigenvalue* refers to the largest eigenvalue λ_1 of the adjacency matrix. It is an indication of network strength, since it increases with the number of connections between the nodes. Networks with a higher number of passes, or networks where the well-connected players are connected between them, will have a higher λ_1 (Buldu et al., 2019). In order to investigate the existence of independent groups in the passing networks, we study the *algebraic connectivity.* It is the second smallest eigenvalue of the Laplacian matrix, which is defined as $D - A$, where D is the diagonal matrix of node degrees and A the adjacency matrix. The lower the algebraic connectivity, the clearer the existence of independent groups. Thus, a high algebraic connectivity implies a more interconnected team. Lastly, the *average shortest path* is an indication of how well-connected the players are. The shortest paths are calculated between all the nodes of the passing networks using Dijkstra's algorithm (Dijkstra et al., 1959). We consider both the weighted and unweighted (binary) passing networks. Again, the distances are defined as the inverse of the number of passes for the weighted passing networks.

4.2.3 Spatial properties

Finally, we also consider spatial properties of the passing networks. The dataset contains the (x,y) position for each event. The x coordinate indicates the nearness (in percentage) to the opponent's goal, while the y coordinate indicates the nearness (in percentage) to the right side of the field (Pappalardo et al., 2019b). As spatial properties we include the network centroid, dispersion around the centroid, average change in x and average change in y. The *network centroid* is the average (x,y) position of the passing network. This is an indication of the performance of teams during the match, since teams that are playing closer to their opponent's goal generally participate in more attacking phases during the match. The *dispersion* describes how spread around the network centroid the players pass. It shows how dense the playing area is of a team. Also, greater dispersion might indicate less involvement of all players in a team. The *average change in x* and *y* refers to the average (percentage) change in x and y coordinate for successful passes in the passing network. We are interested in whether these spatial properties are related to the performance of teams.

4.3 Significance testing

Comparisons between the various network metrics are made using statistical analysis. The comparison is performed according to teams and match outcome (win, draw, loss) in three different ways: comparing the means of the teams, comparing the means of the match outcomes, and comparing the win and loss means per team. With this comparison we want to examine whether network metrics are related to performance. To test for statistically significant differences between the means of the network metrics, several tests are used. The null hypothesis of these tests is that the means of the groups are the same, the alternative hypothesis is that at least one of the means is different.

First, the metrics are analysed for normality using the Shapiro-Wilk test (Shapiro and Wilk, 1965). After this, Levene's test is used for homogeneity of variance across groups (Levene, 1961). For metrics that have a normal distribution according to the Shapiro-Wilk test and equal variances according to Levene's test, the one-way ANOVA test is used (Howell, 2009). In the case of normality but not equal variances, the approximate method of Welch is used (Welch, 1947). The Kruskall-Wallis test is used as alternative to the one-way ANOVA test for metrics that do not have a normal distribution. (Kruskal and Wallis, 1952). This is a non-parametric test, which means that it does not assume anything about the underlying distribution. Thus, equal variances across the groups are not relevant in this case.

4.4 Correlation analysis

To assess dependence between network metrics and goal statistics, a correlation analysis is performed. For this analysis we consider the Pearson's correlations. Correlation coefficients are calculated for the number of goals, the number of conceded goals and the goal difference. Those can be associated to attack ability, defense ability and overall strength, respectively. Confidence intervals for the correlation coefficients are calculated using Fisher's z transformation (Fisher, 1915). With this transformation, the sampling distribution becomes normally distributed and 95% confidence intervals can be calculated. The strength of the correlation coefficients is interpreted according to Evans' classification (Evans, 1996): very weak (≤ 0.20); weak (between 0.20 and 0.40); moderate (between 0.40 and 0.60); strong (between 0.60 and 0.80); very strong (between 0.80 and 1).

4.5 Cluster analysis

Cluster analysis is performed to find groups of teams that behave similarly in terms of network metrics. We investigate whether these clusters are related to performance (the final ranking) of the teams. Clustering is the grouping of observations such that observations in the cluster are considered more similar compared to observations in other clusters. K-means clustering is the most popular method for clustering. Here, the observations are partitioned in k clusters and assigned to the closest cluster centroid. For further details on the k-means clustering algorithm, we refer to Lloyd (1982).

We use the means of all network metrics per match as data for the clustering. To determine the optimal size k, a plot with the total within-cluster sum of squares (WSS) for different values of k is created. The elbow approach is used to determine the optimal size for k. Here, when the elbow (bend) in the plot is chosen as optimal k, the WSS does not decrease much when adding another cluster. Teams are given the cluster to which most observations (matches) are assigned.

5 Predictive analysis

In this section we describe the predictive analysis for investigating the predictive power of the network metrics. An overview is given of the baseline methods, implemented regression and classification models, feature selection and model evaluation.

5.1 Overview implemented models

We consider two approaches for the predictive analysis: modelling the number of goals (goal-based) and modelling the football match outcomes (result-based). For the first approach, we want to predict the number of goals scored by both teams. Since the dependent variable is continuous, this is regression predictive modeling. The latter approach predicts whether a match results in a win, draw or loss for the home team. This is classification predictive modeling. We investigate whether the network metrics have any predictive power. For this purpose, predictive models with and without inclusion of the network metrics as predictor variables are compared. We implement several machine learning models, in particular linear regression and tree-based models. The predictive performance of the models is compared to various baseline methods. The implemented goal-based models are: linear regression, Poisson regression, Lasso and Ridge regression, Multivariate Adaptive Regression Spline, random forest and gradient boosting models. The implemented result-based models are the naive Bayes classifier, multinomial logistic regression, Support Vector Machine, random forest and gradient boosting models. The details of the models are explained below.

5.2 Baseline methods

5.2.1 Goal-based

For the goal-based models, we consider two baseline methods. The first method randomly samples the number of goals for both teams using the proportions of the number of goals from the training set. The second baseline method takes the average goals scored by the team in the previous three matches. The second baseline method is more advanced and expected to have better performance.

5.2.2 Result-based

For the modelling of match outcome, we consider three different baseline methods:

1. Randomly sampling a win, draw or loss based on the proportions in the training set.
2. The team with the highest rank in the matchweek wins, if the teams have an equal rank then the outcome is set to a draw. This is the most advanced baseline method.
3. The most frequent match outcome out of the previous three matches is selected for the home team. If the outcomes have the same frequency, then the outcome is set to a win for the home team.

5.3 Goal-based models

5.3.1 Linear regression

The linear regression is the most basic and commonly used regression model. It aims to find a linear relationship between a dependent variable and predictor variables. The general specification for a linear regression is given by

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i, \tag{5}$$

where y_i refers to the dependent variable, $\mathbf{x}_i$ the vector of predictor variables, $\boldsymbol{\beta}$ the parameter vector and ϵ_i the error term. Here, the dependent variable is the number of goals scored by one team. The model is fitted using the least squares method, which minimizes the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{N}(y_i - \mathbf{x}_i\boldsymbol{\beta})^2 = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{6}$$

Since we are not sure about the importance of all the predictor variables, variable selection is also performed. Variable selection eliminates irrelevant variables, hereby improving model interpretation, reducing model complexity and possibly decreasing the prediction error (Hastie et al., 2005). We perform two approaches: forward and backwards variable selection. During forward selection, the variable that improves the model fit the most is sequentially added. Backwards variable selection starts with all variables and then sequentially removes the variable that least improves the model fit. The best model is chosen based on the maximum value of the adjusted R^2.

5.3.2 Lasso and Ridge regression

Since the linear regression can have high variance, we also consider regularization methods. Two mainly used regularization methods are the Ridge regression (Hoerl and Kennard, 1970) and the Lasso regression (Tibshirani, 1996). The methods add a penalty to the error function which introduces some bias, but reduces variance. This prevents the model from overfitting and can improve predictive performance. Coefficients are shrunk to zero by minimizing a penalized residual sum of squares. In the case of Ridge regression, this penalized RSS is equal to

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2, \tag{7}$$

and for Lasso regression it is

$$RSS + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{8}$$

Here, p is the number of parameters in the model and λ is the tuning parameter which controls the strength of the penalty. Regularization is especially useful when the number of parameters is large or when there is multicollinearity. Lasso regression also enables us to perform variable selection, since coefficients can become exactly zero. To find the optimal value for λ, we perform five-fold cross-validation with a sequence of 100 λ values ranging between $\lambda = 10^{-3}, ..., 10^5$. The λ with the smallest cross-validation error is selected for the model. We fit the model using the **glmnet** package in R (Friedman et al., 2010).

5.3.3 Poisson regression

The Poisson regression is most widely used in literature for the modelling of the number of goals. A Poisson regression is a Generalized Linear Model used for count data. It assumes that the dependent variable follows the Poisson distribution and models the logarithm of the expected value as a linear combination of the predictor variables. Thus,

$$\log(E(y_i|\mathbf{x}_i)) = \mathbf{x}_i\boldsymbol{\beta}. \tag{9}$$

The number of goals is non-negative and discrete, and can be therefore be considered as count data. The Poisson regression assumes equal mean and variance of the dependent variable. When the variance is larger than the mean, also called over-dispersion, the quasi-Poisson model can be used. Here, the variance is modelled as the mean multiplied by a dispersion parameter. For further details and estimating methods regarding the Poisson and quasi-Poisson regression model, we refer to the work of Ver Hoef and Boveng (2007).

5.3.4 Multivariate Adaptive Regression Spline (MARS)

A more flexible, non-parametric regression technique popular for modelling high-dimensional data is MARS (Friedman et al., 1991). Advantages of MARS over the previously described regression models are that MARS automatically performs variable selection, finds knots and detects non-linear relationships and interactions between variables. The model has the form

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} a_m B_m(\mathbf{x}). \tag{10}$$

It is a weighted sum of basis functions $B_m(\mathbf{x})$. The basis function can be a hinge function or a product of two or more hinge functions. A hinge function is defined as $(\mathbf{x}-\mathbf{c})^+$ or $(\mathbf{c}-\mathbf{x})^+$, with

$$(\mathbf{x}-\mathbf{c})^+ = \begin{cases} x-c, & x>c, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad (\mathbf{c}-\mathbf{x})^+ = \begin{cases} c-x, & \text{if } x<c, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The model is build using the forward and backward stepwise procedure. The forward procedure sequentially adds pairs of basis functions, searching over all possible combinations. In the backward stepwise procedure, basis functions that contribute the least to the fit of the model are eliminated. For further details on MARS, we refer to the work of Friedman et al. (1991). We fit the MARS model using the **earth** package in R (Milborrow, 2019). A grid search is performed over two parameters: the degree of interactions (degree) and the number of retained terms (nprune). The parameter grid is defined in Table 7. Five-fold cross-validation is performed to find the best parameters.

Table 7: Parameter grid for MARS.

Parameter	Values
Degree	1, 2, 3
Nprune	2, 4, ..., 100

5.4 Result-based models

5.4.1 Naive Bayes Classifier

The first implemented result-based model is the naive Bayes classifier. We use this model since it is simple, fast and can perform well on complex data. The naive Bayes classifier is a classification algorithm based on Bayes' theorem. This theorem describes the posterior probability of an event using prior knowledge. Using Bayes' theorem, the probability of a class C_k given the feature vector X can be described as

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}. \tag{12}$$

The method is called naive because all features are assumed to be independent. Also, the Gaussian distribution is assumed for a feature vector $\mathbf{x}$ given the class C_k. That is,

$$P(X = \mathbf{x}|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(\mathbf{x}-\mu_k)^2}{2\sigma_k^2}}. \tag{13}$$

Here, μ_k and σ_k^2 refer to the mean and variance of the values of X for class C_k. Due to the simplifications of the naive Bayes classifier, we do expect the other classification methods to have better performance.

5.4.2 Multinomial logistic regression

We also implement the multinomial logistic regression. Similar to the linear regression model, the multinomial logistic regression is easy to implement, does not require parameter tuning and gives interpretable results. Multinomial logistic regression is a classification method that extends the logistic regression to multiclass outcomes. If we assume that there are J discrete outcomes, then these outcomes follow the multivariate Bernoulli distribution, with $P(y_i = j) = \pi_{ij}$ and $\sum_{j=1}^{J} \pi_{ij} = 1$. Since the probabilities are bounded between 0 and 1, and the probabilities have to sum to 1, the logistic function is chosen for modelling π_{ij}.

$$P(y_i = j|\mathbf{x}_i; \boldsymbol{\beta}_1, ...\boldsymbol{\beta}_l))) = \frac{\exp(\mathbf{x}_i\boldsymbol{\beta}_j)}{\sum_{l=1}^{J} \exp(\mathbf{x}_i\boldsymbol{\beta}_J))}, \quad \text{for } l = 1, ..., J. \tag{14}$$

Here, $\mathbf{x}_i$ is the vector of predictor variables and $\boldsymbol{\beta}_j$ the parameter vector for outcome j. For parameter identification, we impose outcome J as the base category and set $\boldsymbol{\beta}_J$ to zero. Now the model assumes that the log-odds are a linear combination of the predictor variables. The log-odds ratio for outcome j versus outcome J is defined as

$$\ell_{j|J} = \frac{\pi_{ij}}{\pi_{iJ}} = \mathbf{x}_i\boldsymbol{\beta}_j. \tag{15}$$

The parameters are estimated using Maximum Likelihood Estimation.

5.4.3 Support Vector Machine

The next model we implement popular for solving classification tasks is the Support Vector Machine (SVM). SVMs are known for being extremely effective when dealing with high-dimensional feature spaces (Baboota and Kaur, 2019). They find the separating margin hyperplane that gives maximal and equal distance to all the outcome classes (Boser et al., 1992). Observations close to the optimal separating hyperplane are called support vectors; others are considered irrelevant for finding the optimal separation. The objective function of the SVM consists of a part that maximizes the margin and a part that minimizes the classification error. Let the binary variable $y_i \in \{-1, 1\}$ and the training data $x_i = (x_i^{(1)}, \ldots, x_i^{(n)}) \in \mathbb{R}^n$, then the primal optimization problem is given by

$$\begin{aligned} \underset{w,b,\xi}{\text{Min}} \quad & \frac{1}{2}\boldsymbol{w}^\intercal\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^\intercal\phi(x_i) + b) \geq 1 - \xi, \quad i = 1, \ldots, N \\ & \xi_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{16}$$

Here, $\boldsymbol{w}$ represents the weight vector, ξ the slack variable to allow misclassifications and b the bias. The first part of the minimization function maximizes the margin, where the margin width between both hyperplanes is equal to $= \frac{2}{||\boldsymbol{w}||^2}$. The second part minimizes the sum of the distances for all misclassifications. C represents the cost parameter that determines the trade-off between margin width and classification error. $\phi(\cdot)$ is a non-linear function, related to the kernel trick. When the data is not linearly separable, the kernel transforms the data into a higher-dimensional space so that linear separation is possible. The kernel is equal to $K(x_i, x_j) = \phi(x_i)^\intercal \phi(x_j)$. The kernels that are most commonly used are the linear kernel ($x_i^\intercal x_j$), polynomial of degree d kernel $(\gamma x_i^\intercal x_j + r)^d$, sigmoid kernel ($\tanh\{\gamma x_i^\intercal x_j + r\}$) and radial basis function (RBF) kernel ($\exp\{-\gamma||x_i - x_j||^2\}$). The primal optimization problem is converted into a dual problem that can easily be solved by applying Lagrangian multipliers. The SVM can only deal with binary classification. To allow for multi-class classification, the one-against-one technique is used. In the one-against-one technique $\frac{k(k-1)}{2}$ binary classifiers are constructed, where k is the number of outcome classes (Karatzoglou et al., 2006). Here, each pair of outcome classes is trained against each other. The final class is then chosen based on a voting scheme, where the most frequently predicted class is selected as final class. For further details on the optimization of the SVM and the use of kernels, we refer to Vapnik (2013).

We implement the different kernels and perform a grid search to find the optimal parameters using five-fold cross-validation. The main parameters that require tuning are the cost parameter C and the γ parameter. A higher C will result in less classification errors, but comes with the risk of overfitting. The γ parameter determines how widespread the influence is of observations. If γ is large, more observations will be used for determining the optimal separating hyperplane. The grid search evaluates the following values for the two parameters: $C = 2^{-5}, 2^{-3}, ..., 2^{15}$ and $\gamma = 2^{-15}, 2^{-14}, ..., 2^{5}$. For the polynomial kernel we also consider values of 0, 1, ..., 6 for the degree parameter. We use the **e1071** package in R to model the SVM (Meyer et al., 2019). Attractive features of the SVM include the use of kernels for non-linear separation, the absence of local minima solutions and good prediction accuracy.

5.5 Goal-based and result-based models

5.5.1 Random forest

The first method implemented for both regression and classification predictive modelling is the random forest. It is a tree-based machine learning method developed by Breiman (2001). The random forest is an ensemble method that combines predictions of multiple decision trees into a single prediction. Decision trees are popular algorithms in machine learning. They are easy to use and interpret, but they are prone to overfitting and unstable due to high variance. Random forests are able to overcome these problems. In a random forest, predictions from multiple de-correlated decision trees are averaged (Hastie et al., 2005). To reduce variance, the trees are taught on bootstrap samples and aggregated (bagging). Bootstrapping is randomly sampling from the data with replacement. Bootstrap samples are considered to be independent and representative for the true underlying distribution. The results of the trees build on bootstrap samples are then aggregated. By combining these trees, the variance is reduced and the weak learners are turned into a strong learner. The variance of this bagged estimator is equal to

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2, \tag{17}$$

where ρ is the pairwise correlation and B the number of trees. Thus, variance can further be reduced by reducing the pairwise correlation ρ. Therefore, another source of randomness is introduced by splitting on a random subset of features. For each split in the trees, m of the p input variables are chosen at random as candidates for splitting. For regression, $m = p/3$ and for classification $m = \sqrt{p}$ is recommended (Breiman, 2001). The random forest algorithm can be summarized as follows:

1. Create B subsamples by sampling with replacement from the data.
2. For each subsample, create a decision tree to obtain an ensemble of B decision trees. Before each split in the decision tree, m input variables are randomly chosen as candidates for splitting.
3. Obtain the final prediction by taking the average prediction of the B trees (regression) or by taking the majority vote (classification).

Random forests are known to have good predictions on complex data, high speed and limited sensibility with respect to the choice of parameters. Model interpretability is more difficult compared to decision trees, however, the importance of variables can be examined by looking at the mean decrease in impurity measures such as the Gini index.

A randomized grid search is performed to find the optimal model. Contrary to a regular grid search, where every parameter combination is tried, a randomized grid search tries randomly selects parameter combinations. This decreases computation time, which enables it to search a larger parameter space (Bergstra and Bengio, 2012). We use the **ranger** package in R for implementing the random forest (Wright and Ziegler, 2015). The following parameter grid with five-fold cross-validation and 100 iterations is used:

Table 8: Parameter grid for random forest.

Parameter	Values
Estimators	100, 150, 200, ..., 500
Max depth	5, 10, 15, ..., 100, None
Min samples split	2, 5, 10, 15, 100
Max features	$p/3$, $\sqrt{p}$, log2(p)
Min samples leaf	1, 2, 5, 10
Bootstrap	True, False

The number of estimators refers to the number of trees in the forest. With a large number of trees, the forest can capture more information. However, a large number of trees can make the training time for the random forest very long. The maximum depth represents how deep each tree can get. The deeper the tree, the more information the tree can capture. The minimum samples split is the minimum samples required for a split at a decision node; the minimum samples leaf is the minimum number of samples required to be at an end node; the maximum features represents the number of randomly chosen features m to split on for each tree.

5.5.2 Gradient boosting

The final implemented model for both regression and classification is gradient boosting trees. Compared to the random forest, which performs bagging, gradient boosting is more focused on reducing bias than variance. Boosting is an ensemble algorithm developed by Freund and Schapire (1997).

It iteratively converts a set of weak learners into a strong learner by reweighting the weak learners based on the accuracy. A weak learner has an error rate that is slightly better than random guessing (Hastie et al., 2005). The most popular boosting algorithm is the AdaBoost.M1 algorithm for classification from Freund and Schapire (1997). This algorithm sequentially applies weak learners to modified versions of the data, where more weight is given to observations that are difficult to classify. The final prediction for a multiclass classification is

$$\hat{g}_M(\mathbf{x}) = \underset{y \in Y}{\operatorname{argmax}} \sum_{m=1}^{M} \alpha_m 1(g_m(\mathbf{x}) = y), \tag{18}$$

where α_m is the weight from weak classifier $g_m(x)$, based on the error of the classifier, and M the number of weak classifiers. Better learners are exponentially given more weight. For further details on the Adaboost algorithm we refer to Freund and Schapire (1997). As shown by Breiman (1999) and Friedman et al. (2000), the algorithm is similar to a forward stagewise additive model with exponential loss function

$$L(y, f(\mathbf{x})) = \exp(-yf(\mathbf{x})). \tag{19}$$

Gradient boosting was proposed by Friedman (2001) and is focused on the idea of boosting as optimizing a loss function using a gradient descent procedure. The AdaBoost algorithm is a special case of gradient boosting that uses the exponential loss function. Gradient boosting is more flexible since the user can define the loss function. However, a problem with ensembles of trees, like the random forest and gradient boosting trees, is that training the trees can take long for large, high-dimensional datasets.

Chen and Guestrin (2016) propose a scalable machine learning system for tree boosting: Extreme Gradient Boosting (XGB). Over the past years this has been the winning algorithm in many machine learning challenges. The main difference between gradient boosting and XGB is that it has a regularized model formalization to control overfitting, which penalizes more complex models. XGB also uses some algorithmic optimizations and implementations that improve execution speed and model performance. These include automatic sparse data optimization, parallel and distributed computation and an effective cache-aware block structure for out-of-core tree learning.

In this research we implement two different boosting algorithms: Gradient Boosting Trees (**sklearn.ensemble** module from Python's **scikit-learn** package) and XGB Regressor (**XGBoost** package from Python). We perform a randomized search with five-fold cross-validation and 100 iterations on the following grid for Gradient Boosting Trees:

Table 9: Parameter grid for Gradient Boosting Trees.

Parameter	Values
Estimators	100, 150, 200, ..., 500
Max depth	3, 5, 7, 15, 25, 30, 50
Min samples split	2, 5, 10
Max features	auto, sqrt, log2
Min samples leaf	1, 2, 4

For XGB we use the following grid:

Table 10: Parameter grid for XGB.

Parameter	Values
Estimators	100, 150, 200, ..., 500
Max depth	3, 5, 7, 15, 25, 30, 50
Min samples split	2, 5, 10
Min samples leaf	1, 2, 4
λ	0.01, 0.015, 0.025, 0.05, 0.075, 0.1, 1
α	0, 0.1, 0.5, 1
Min child weight	0.001, 0.01, 1, 3, 5, 7
Colsample by tree	0.6, 0.7, 0.8, 0.9, 1
Subsample	0.6, 0.7, 0.8, 0.9, 1

The α and λ parameters control the L1 and L2 regularization term on the weights (Chen and Guestrin, 2016). These regularization parameters can help reduce model complexity and increase performance. The minimum child weight corresponds to minimum number of instances needed to be in each node. The colsample by tree is the same as the max features used in the random forest and gradient boosting trees. It is the fraction of variables to be randomly sampled for each tree. Subsample is the fraction of randomly sampled instances used to build each tree.

5.6 Feature selection and multicollinearity threshold

Feature selection can improve model interpretability, increase predictive performance and reduce model complexity. We include the option of using no feature selection, using lasso regression as feature selection or using the correlation method as feature selection. With no feature selection, we include all variables for prediction. Lasso regression, as described in Section 5.3.2, performs feature selection by shrinking coefficients to zero. The correlation method excludes features that have a correlation of lower than 0.1 with the dependent variable.

We also include the option to use a multicollinearity threshold. Multicollinearity is the phenomenon of a predictor variable being highly correlated to one or more other predictor variables, resulting in a relatively large standard error (Allen, 1997). Incorporating highly correlated variables in a regression model thus leads to unstable regression coefficients and undermining of the statistical significance of the predictor variables. As a solution to multicollinearity, we include the option of removing highly correlated variables. If the correlation between two numeric predictor variables is larger than 0.75, one of the predictor variables is arbitrarily removed from the data.

5.7 Model evaluation

To evaluate the models, the data is split into a train set (75%) and a test set (25%). Since many predictor variables are based on averages over the past three matches, those predictor variables will be less well-defined for the first three matchweeks. To deal with this problem, we consider the first two matchweeks as a warm-up period and remove them from the training data. The data consists of 380 matches over 38 matchweeks. For the goal-based models, each match results in two observations, while for the result-based models we have one observation per match. Table 11 provides an overview of the warm-up, train and test set for both modelling approaches.

Table 11: Overview of the matchweeks and number of observations for the warm-up, train and test set.

	Goal-based model	Result-based model
Warm up: matchweeks	1-2	1-2
Warm-up: number of observations	40	20
Train set: matchweeks	3-29	3-29
Train set: number of observations	540	270
Test set: matchweeks	30-38	30-38
Test set: number of observations	180	90

For parameter tuning of the models, cross-validation is used over the train set. Cross-validation splits the data into K folds, where the model is trained on $K-1$ folds and predictions are made for the remaining fold k. This process is repeated for $k = 1, ..., K$ and the test error is averaged. Cross-validation prevents the model from overfitting. The parameters that minimize the cross-validation error are selected for the model. The performances of the different models over the test set will be compared using either regression or classification evaluation metrics.

5.7.1 Regression evaluation metrics

For evaluating the predictive performance of the goal-based models, we use the Mean Squared Error (MSE) and Mean Absolute Error (MAE). The MSE takes the square of the error and is calculated by

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{20}$$

Thus, larger errors have a relatively larger effect on the score, making it sensitive to outliers. A metric that is less sensitive to outliers is the MAE. It is given by

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|. \tag{21}$$

All observations have equal weight in the MAE. The MAE has a more intuitive interpretation compared to the MSE.

5.7.2 Classification evaluation metrics

For the result-based models, we consider other evaluation metrics, including the accuracy, precision, recall and F1 score. For the precision, recall and F1 score we consider both the weighted score for all match outcomes and the weighted score only considering wins and losses. To clarify the definitions of these evaluation metrics, we use the confusion matrix for match outcomes.

Table 12: Confusion matrix for the match outcome.

	Predicted Class		
True Class	Win	Draw	Loss
Win	n_{11}	n_{12}	n_{13}
Draw	n_{21}	n_{22}	n_{23}
Loss	n_{31}	n_{32}	n_{33}

The accuracy is then defined as

$$\frac{n_{11} + n_{22} + n_{33}}{n_{11} + n_{12} + n_{13} + n_{21} + n_{22} + n_{23} + n_{31} + n_{32} + n_{33}}.$$

Precision and recall are calculated per class. As example, we illustrate the precision and recall for win outcomes. Precision is then defined as

$$\frac{n_{11}}{n_{11} + n_{21} + n_{31}}$$

and recall is defined as

$$\frac{n_{11}}{n_{11} + n_{12} + n_{13}}.$$

The precision is focused on how precise your predictions for a class are, while recall looks from the perspective of the actual class outcome. For example, if we do not often predict a win compared to the actual number of wins, but all predicted wins are actual wins, the precision is high and the recall low. The F1 score finds a balance between precision and recall. It is defined as

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

for each class. Combining the metrics for the classes is done by weighting the scores based on the proportions of the classes.

6 Results Exploratory Analysis

6.1 Adjacency matrix

The adjacency matrix is calculated for all matches and teams in the dataset. As an example, we provide the adjacency matrices of the passing networks for a match between Manchester City (ended first place) and Brighton & Hove Albion (ended 15th place) on May 9, 2018. The final score of this match was 3 - 1. For Manchester City and Brighton & Hove Albion, the adjacency matrices are given in Table 13 and 14, respectively.

Table 13: Adjacency matrix for the passing network of Manchester City during the match Manchester City - Brighton & Hove Albion.

To/of						**Players**						
Players	Bravo	Laporte	Kompany	Touré	Danilo	Zinchenko	Sané	Gündoğan	de Jesus	Bernardo Silva	Fernandinho	**Passes performed**
Bravo	-	0	3	0	0	2	0	1	0	0	1	7
Laporte	3	-	16	10	2	13	3	11	0	1	12	71
Kompany	5	11	-	18	14	19	0	13	5	7	7	99
Touré	0	10	19	-	6	21	8	6	7	6	8	91
Danilo	0	9	15	2	-	4	3	6	0	7	5	51
Zinchenko	1	13	19	17	4	-	9	7	4	0	8	82
Sané	0	1	0	6	1	8	-	9	4	0	6	35
Gündoğan	0	9	15	12	6	13	3	-	0	1	10	69
de Jesus	0	0	4	6	2	1	2	5	-	2	4	26
Bernardo Silva	0	1	3	7	12	1	2	3	0	-	7	36
Fernandinho	1	8	7	9	6	12	7	12	3	7	-	72
Passes received	10	62	101	87	53	94	37	73	23	31	68	
Total interactions	17	133	200	178	104	176	72	142	49	67	140	**1278**

The total number of interactions of Manchester City is quite high (1278 interactions). Also, the amount of passes received and performed differs largely per player. Kompany has the largest amount of interactions (200), with 101 passes received and 99 passes performed. Kompany and Zinchenko are the players with the most interactions, with 19 passes played to Zinchenko and 19 passes received from Zinchenko. The goalkeeper, Bravo, has the lowest amount of interactions.

Table 14: Adjacency matrix for the passing network of Brighton & Hove Albion during the match Manchester City - Brighton & Hove Albion.

To/of	Players										
Players	Pröpper	Stephens	Saltor	Knockaert	Duffy	Groß	Izquierdo	Dunk	Ryan	Ulloa	Bong	**Passes performed**
Pröpper	-	3	1	1	0	3	0	1	0	1	1	11
Stephens	2	-	3	2	4	3	0	2	3	1	5	25
Saltor	4	1	-	9	4	5	0	3	3	1	0	30
Knockaert	3	6	8	-	1	5	0	0	0	0	0	23
Duffy	1	2	3	1	-	2	0	3	3	1	1	17
Groß	1	5	9	2	2	-	0	0	0	4	0	23
Izquierdo	1	1	0	1	0	0	-	0	0	0	2	5
Dunk	0	3	2	1	2	2	1	-	0	0	2	13
Ryan	1	1	1	0	0	0	0	0	-	0	0	3
Ulloa	1	4	1	1	0	3	1	2	0	-	1	14
Bong	0	1	0	1	0	0	3	2	0	2	-	9
Passes received	14	27	28	19	13	23	5	13	9	10	12	
Total interactions	25	52	58	42	30	46	10	26	12	24	21	**346**

Brighton & Hove Albion only has a total of 346 successful interactions. The player with the most interactions is Saltor with 58 passes received and performed. The amount of interactions is much lower compared to Manchester City, indicating that the number of interactions might be related to performance of the teams. Also, a few players barely perform any successful passes. Izquierdo, the left winger, has the least successful interactions for Brighton & Hove Albion during this match.

6.2 Visualization passing networks

Passing networks are visualized using R. We again analyse the match between Manchester City and Brighton & Hove Albion. Figure 2 and 3 present the undirected passing networks of the total match for Manchester City and Brighton & Hove Albion. The position of the nodes is the average position of a player for a successful pass. The nodes are sized according to node out-degree, described in Section 4.2.1. Edge size is based on the number of successful passes between two players. Both are rescaled to improve readability. Here, the minimum number of passes for an edge to be created is set to three.

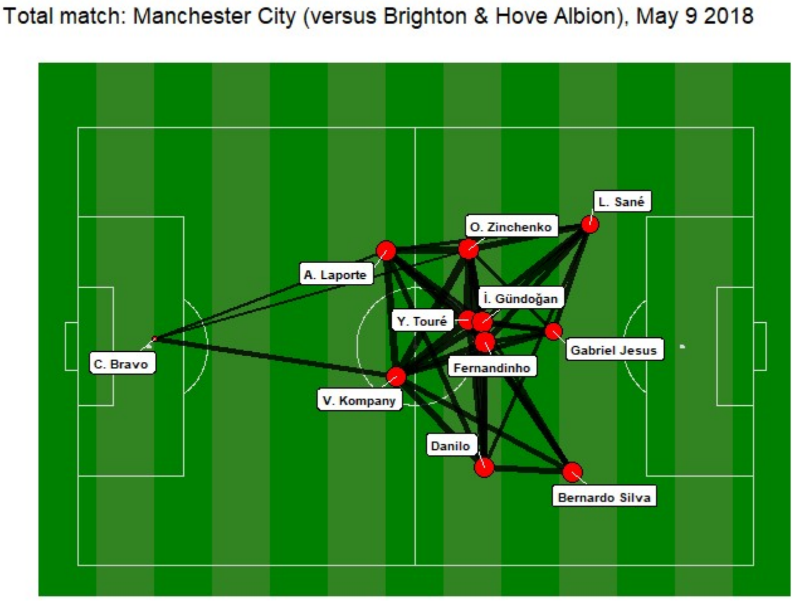


Figure 2: Undirected passing network for Manchester City during the match Manchester City - Brighton & Hove Albion.

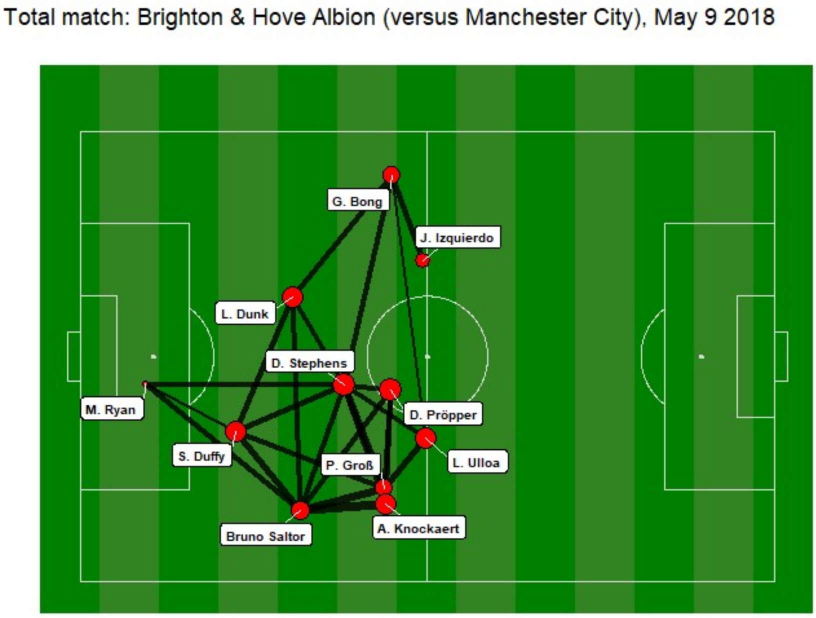
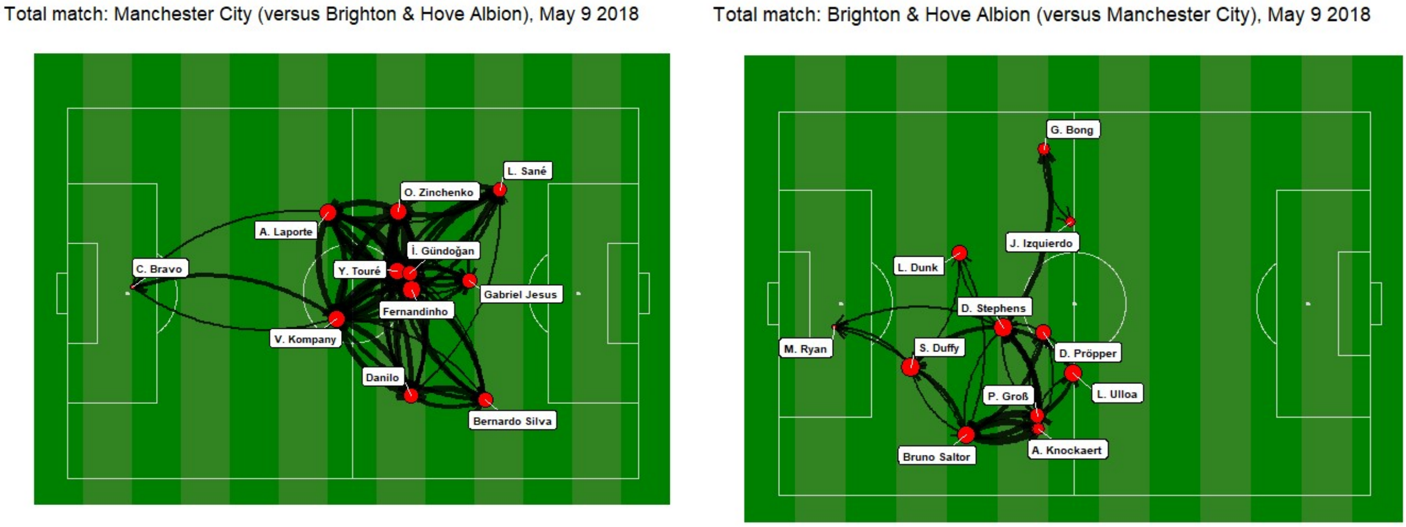


Figure 3: Undirected passing network for Brighton & Hove Albion during the match Manchester City - Brighton & Hove Albion.

These visualizations again show that the players of Manchester City are better connected compared to Brighton & Hove Albion. Also, the average position of the players of Manchester City is mostly on the half of the opponent and the playing area is more dense, while Brighton & Hove Albion's average position is closer to their own goal and their average position is more spread around. We also considered directed passing networks. The directed passing networks of both teams are shown in Figure 4. The direction of a pass is indicated by an arrow. These figures also show that the players of Manchester City have more interactions between them. Visualizations of the undirected and directed passing networks by half for this specific match are given in Appendix C.



(a) Manchester City.

(b) Brighton & Hove Albion.

Figure 4: Directed passing networks for both teams during the Manchester City - Brighton & Hove Albion match.

6.3 Comparison network metrics

Statistical tests are performed for comparing the network metrics according to match outcome, team, and win or loss per team. Table 15 presents the means (SD) of the network metrics for a win, draw and loss, the statistical test performed and the corresponding p-value. The eigencentrality metrics, the betweenness score and the average change in x are not statistically different for the match outcomes. For the other network metrics, at least one of the means is different. Among others, the clustering coefficient, largest eigenvalue, algebraic connectivity and position of x are higher for winning teams compared to losing teams. These values are 4.11 versus 2.94 for clustering coefficient, 34.57 versus 25.31 for largest eigenvalue, 8.04 versus 5.94 for algebraic connectivity and 30.47 versus 27.15 for position of x. Also, winning teams have a smaller average shortest path (0.42 versus 0.51) and a larger mean degree (7.12 versus 6.75). For draws, the mean values are mostly closer to the mean loss values. The average change in y is largest for draw outcomes.

Table 15: Mean (SD) values, p-value and corresponding statistical test for the comparison of the network metrics according to a win, draw and loss match outcome.

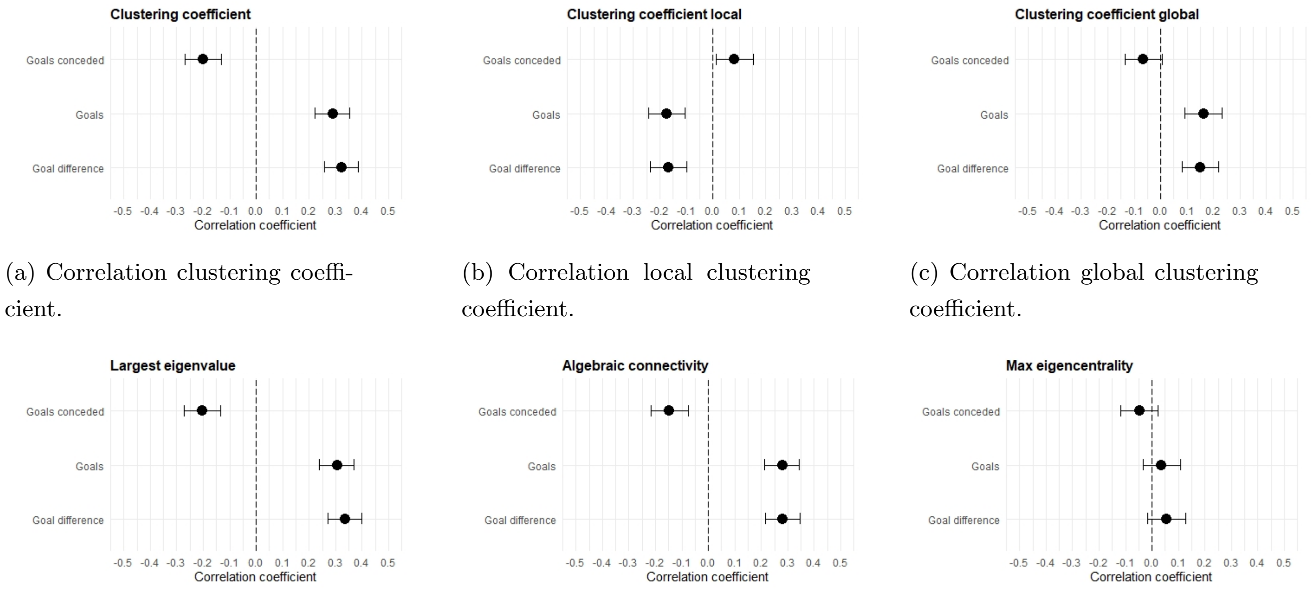
Network features	Win	Draw	Loss	p-value	test	Difference
Number of matches	281	198	281			
Clustering coefficient	4.11 (2.39)	3.09 (1.56)	2.94 (1.47)	<0.001	Kruskall-Wallis	Yes
Clustering coefficient local	0.29 (0.04)	0.31 (0.05)	0.31 (0.04)	<0.001	Kruskall-Wallis	Yes
Clustering coefficient global	0.84 (0.08)	0.82 (0.07)	0.83 (0.07)	<0.001	Kruskall-Wallis	Yes
Largest eigenvalue	34.57 (18.15)	26.43 (12.06)	25.31 (11.42)	<0.001	Kruskall-Wallis	Yes
Algebraic connectivity	8.04 (4.68)	6.15 (3.59)	5.94 (3.25)	<0.001	Kruskall-Wallis	Yes
Max eigencentrality	0.47 (0.04)	0.47 (0.04)	0.46 (0.05)	0.560	Kruskall-Wallis	**No**
Sd eigencentrality	0.12 (0.02)	0.12 (0.02)	0.12 (0.02)	0.219	One-way analysis of means	**No**
Mean eigencentrality	0.28 (0.01)	0.28 (0.01)	0.28 (0.02)	0.505	Kruskall-Wallis	**No**
Closeness	0.20 (0.06)	0.17 (0.05)	0.17 (0.04)	<0.001	Kruskall-Wallis	Yes
Betweenness	8.37 (1.19)	8.43 (1.13)	8.23 (1.09)	0.124	Kruskall-Wallis	**No**
Average shortest path	0.42 (0.20)	0.50 (0.22)	0.51 (0.20)	<0.001	Kruskall-Wallis	Yes
Position x	30.47 (6.05)	27.99 (5.07)	27.15 (4.96)	<0.001	Approximate method of Welch	Yes
Position y	31.15 (5.56)	28.82 (4.63)	29.12 (4.68)	<0.001	Approximate method of Welch	Yes
Dispersion	29.84 (1.35)	30.04 (1.31)	29.73 (1.36)	0.043	Kruskall-Wallis	Yes
Dispersion position x	27.27 (1.60)	27.59 (1.76)	26.84 (1.84)	<0.001	Kruskall-Wallis	Yes
Average change x	0.99 (0.67)	1.00 (0.77)	0.94 (0.70)	0.516	One-way analysis of means	**No**
Average change y	0.18 (0.68)	0.30 (0.72)	0.11 (0.75)	0.023	One-way analysis of means	Yes
Closeness binary	0.08 (0.01)	0.08 (0.01)	0.08 (0.01)	<0.001	Kruskall-Wallis	Yes
Betweenness binary	2.91 (1.14)	3.36 (1.10)	3.33 (1.09)	<0.001	Kruskall-Wallis	Yes
Average shortest path binary	1.29 (0.12)	1.34 (0.11)	1.33 (0.11)	<0.001	Kruskall-Wallis	Yes
Mean degree	7.12 (1.06)	6.73 (0.92)	6.75 (0.94)	<0.001	Kruskall-Wallis	Yes
Sd degree	1.62 (0.34)	1.70 (0.31)	1.69 (0.32)	0.018	One-way analysis of means	Yes
Max degree	9.28 (0.88)	9.11 (0.87)	9.10 (0.90)	0.007	Kruskall-Wallis	Yes

The network metrics per team are given in Appendix D. All network metrics are statistically different based on a 5% significance level. The means and standard deviation differ largely per team. For easier comparison we also provide boxplots of the network metrics, ordered by final ranking of the teams. Appendix E presents the final ranking of the teams and Appendix F shows the boxplots. The teams with a better final ranking have in general a higher clustering coefficient, lower local clustering coefficient, higher global clustering coefficient, higher largest eigenvalue, higher algebraic connectivity, higher closeness score, lower average shortest path, higher x and y position, higher average change in x and a higher mean degree compared to lower ranked teams. The eigencentrality metrics, dispersion, average change in y and maximum and standard deviation for degree appear to be more similar for the teams.

Appendix G contains the season plots for the network metrics of teams that ended high in final ranking (Manchester City, Manchester United), middle (Newcastle United) and low (Stoke City, West Bromwhich Albion). For all matches in the season, the mean values are plotted per team. These plots show that Manchester City, followed by Manchester United, mostly has the highest values for the clustering coefficient, largest eigenvalue, algebraic connectivity, closeness, position of x and mean degree, and the lowest for average shortest path. Newcastle United seems closer to the lowest performing teams in terms of metrics, Stoke City and West Bromwich Albion. Finally, Appendix H shows the network metrics according to win and loss per team. Most network metrics do not differ for a win or loss. West Bromwich Albion, that finished last in the season, differs mostly in network metrics for a win and a loss.

6.4 Correlation analysis

Figure 5 represents the correlations between the individual network metrics and the number of goals, number of conceded goals and goal difference. Those can be associated to attack ability, defense ability and overall strength, respectively. The clustering coefficient, largest eigenvalue, algebraic connectivity, closeness and position of x have the largest positive (negative) correlation for number of goals and goal difference (goals conceded). The average shortest path has a large negative correlation with the number of goals and goal difference. For the other networks metrics, there seems to be a very weak relationship or no relationship at all.



(a) Correlation clustering coefficient.

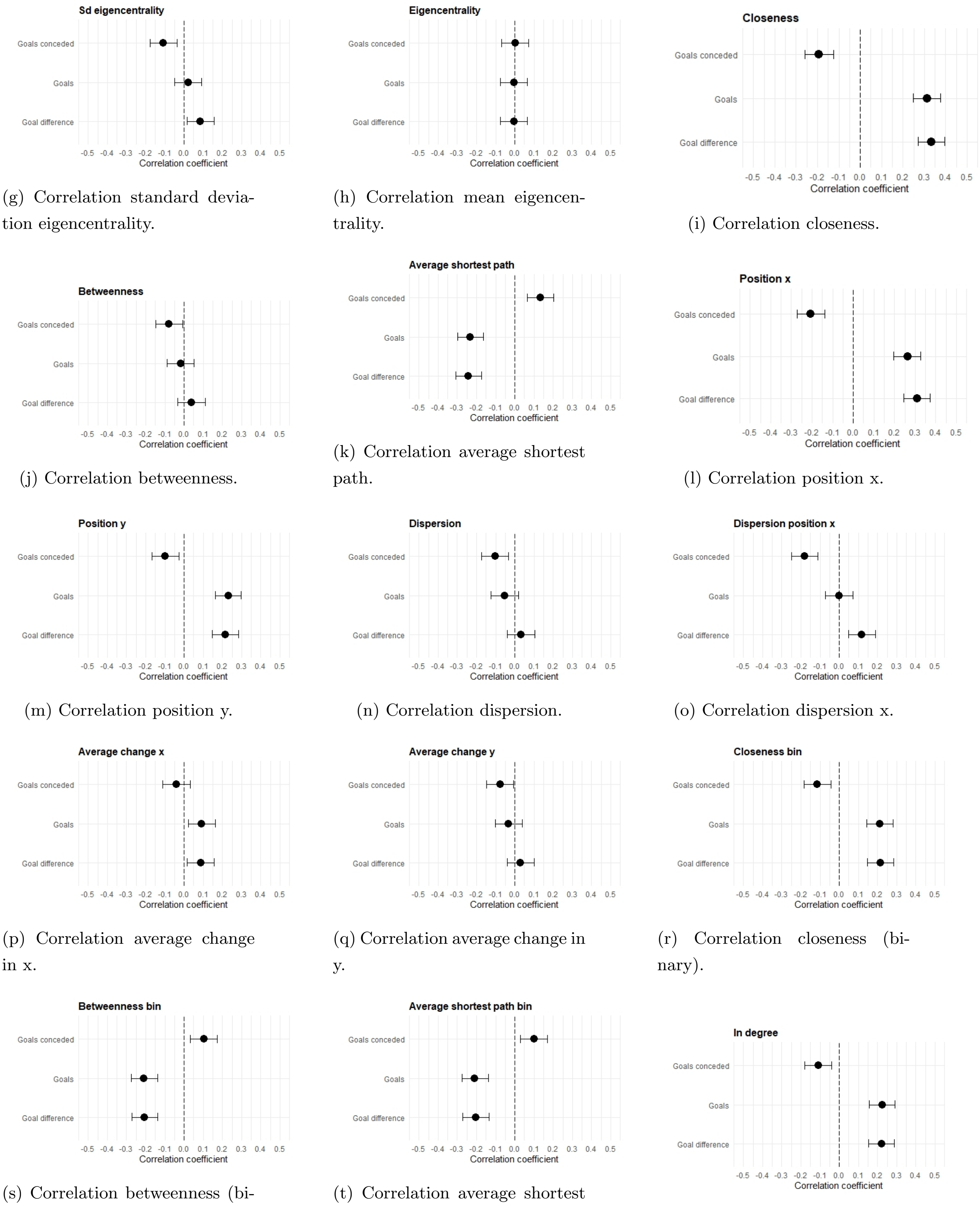
(b) Correlation local clustering coefficient.

(c) Correlation global clustering coefficient.

(d) Correlation largest eigenvalue.

(e) Correlation algebraic connectivity.

(f) Correlation maximum eigencentrality.



(g) Correlation standard deviation eigencentrality.

(h) Correlation mean eigencentrality.

(i) Correlation closeness.

(j) Correlation betweenness.

(k) Correlation average shortest path.

(l) Correlation position x.

(m) Correlation position y.

(n) Correlation dispersion.

(o) Correlation dispersion x.

(p) Correlation average change in x.

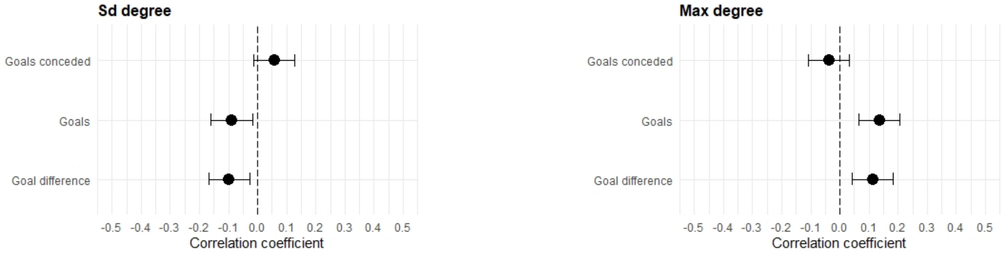
(q) Correlation average change in y.

(r) Correlation closeness (binary).

(s) Correlation betweenness (binary).

(t) Correlation average shortest path (binary).

(u) Correlation mean degree.



(v) Correlation standard deviation degree.

(w) Correlation maximum degree.

Figure 5: Correlation coefficients (95% confidence intervals) between the number of conceded goals, the number of scored goals and the goal difference, and the individual network metrics.

6.5 Cluster analysis

The elbow plot showed that the optimal number of clusters is four. The assigned cluster, cluster probability and final ranking for the Premier League teams are reported in Table 16. Table 17 shows the mean values for the network metrics per cluster. Previous analyses showed that among others the clustering coefficient, largest eigenvalue, algebraic connectivity, closeness, average shortest path, position of x and mean degree mostly differ per team and/or are related to goal statistics. The first cluster mostly contains matches of Manchester City, the winner of the season. The above mentioned metrics are relatively higher for this cluster, with an exception for the average shortest path that is relatively lower. The second cluster contains mostly matches from the teams with a final ranking between two and seven. In general, the metrics that are high (low) for the first cluster are also high (low) for the second cluster. The third cluster has the lowest mean values for the network metrics. This cluster contains mostly matches for the lowest ranked teams, with approximately 63% of the matches of Stoke City and 68% of the matches of West Bromwich Albion assigned to it. The mean networks metrics for the fourth cluster are roughly in between the mean values for the second and the third cluster.

Table 16: Assigned cluster, cluster probability and final ranking for the teams in the Premier League 2017-18 season.

Team	Name	Cluster	Cluster probability	Final ranking
1625	Manchester City	1	60.526	1
1609	Arsenal	2	47.368	6
1610	Chelsea	2	55.263	5
1611	Manchester United	2	47.368	2
1624	Tottenham Hotspur	2	52.632	3
1612	Liverpool	2	47.368	4
1646	Burnley	3	55.263	7
1639	Stoke City	3	63.158	19
1633	West Ham United	3	47.368	13
1613	Newcastle United	3	60.526	10
1627	West Bromwich Albion	3	68.421	20
1631	Leicester City	4	55.263	9
1651	Brighton & Hove Albion	4	68.421	15
1628	Crystal Palace	4	50.000	11
1673	Huddersfield Town	4	47.368	16
1623	Everton	4	55.263	8
10531	Swansea City	4	63.158	18
1619	Southampton	4	47.368	17
1644	Watford	4	60.526	14
1659	AFC Bournemouth	4	55.263	12

Table 17: Mean values for the network metrics per cluster.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**
Clustering coefficient	8.092	5.005	1.635	3.097
Clustering coefficient local	0.272	0.281	0.332	0.298
Clustering coefficient global	0.914	0.883	0.767	0.839
Largest eigenvalue	64.610	42.113	14.804	26.661
Algebraic connectivity	13.742	9.615	3.793	6.320
Max eigencentrality	0.467	0.470	0.470	0.462
Sd eigencentrality	0.135	0.128	0.118	0.121
Mean eigencentrality	0.272	0.275	0.279	0.277
Closeness	0.290	0.231	0.128	0.177
Betweenness	8.487	8.449	8.293	8.282
Average shortest path	0.212	0.296	0.715	0.434
Position x	39.323	33.362	22.860	28.597
Position y	37.190	34.304	24.398	30.307
Dispersion	29.539	30.240	29.174	30.236
Dispersion position x	27.211	27.669	26.642	27.388
Average change in x	1.618	1.349	0.571	0.976
Average change in y	0.246	0.152	0.134	0.227
Closeness binary	0.086	0.083	0.071	0.078
Betweenness binary	1.754	2.298	4.303	3.025
Average shortest path binary	1.175	1.230	1.434	1.303
Mean degree	8.246	7.716	5.853	7.002
Sd degree	1.555	1.670	1.660	1.695
Max degree	9.885	9.725	8.420	9.330

The clustering results are visualized in Figure 6. Principal components analysis (PCA) is performed for plotting the observations according to the first two principal components that explain the majority of variance. Cluster 3 and 4 have the most overlapping attributes.

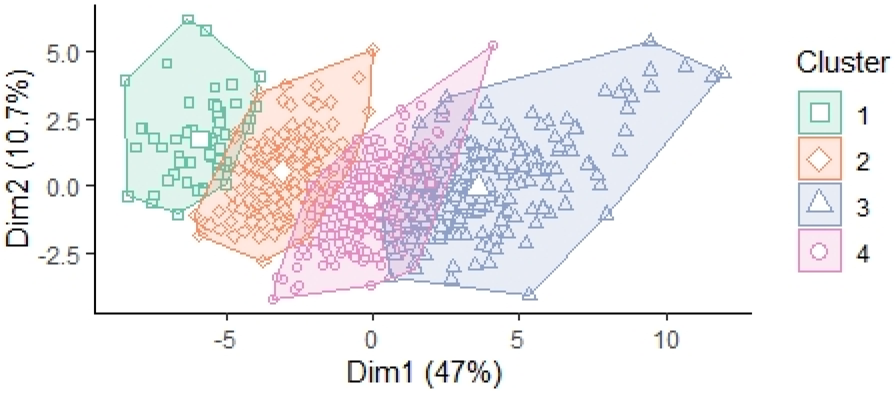


Figure 6: Visualization of the *k*-means clusters using PCA.

7 Discussion Exploratory Analysis

The adjacency matrix can be useful for identifying key players and pairs of players with a high amount of interactions. Coaches and trainers can gain insight in the performance of players and quantify their contribution to the team. Also, analysing the adjacency matrix over time allows for assessing progress of players and the variability of relations between players. Especially the visualizations of the passing network may provide useful insights. Besides the number of successful passes, the visualization displays the average x and y position of the players. These positions indicate how dense the playing area is and can also be related to performance of the team. In our example we clearly see that Manchester City, the winner of the match, plays closer to the goal of Brighton & Hove Albion. The visualization allows for easier interpretation of the different interactions in a passing network and easier comparison between multiple passing networks (first half versus second half, over time, between teams).

Comparing the network metrics according to match outcome, team, and win or loss per team, we found clear differences. Greater values for the clustering coefficient, largest eigenvalue, algebraic connectivity, position of x and mean degree were observed for winning teams compared to losing teams. Also, the average shortest path for winning teams is smaller than for losing teams. The boxplots with the network metrics per team ordered by final ranking are in line with these findings. Teams with a higher final ranking have in general a higher clustering coefficient, higher largest eigenvalue, higher algebraic connectivity, higher closeness, lower average shortest path, higher position of x and y, higher average change in x and higher mean degree compared to lower ranked teams. The high clustering coefficient and largest eigenvalue imply a robust and strong network, where well-connected players are connected between them. The other network metrics imply that winning teams play closer to the opponent's goal and have a more interconnected team where the nodes are closer together. Furthermore, correlation analysis showed that the clustering coefficient, largest eigenvalue, algebraic connectivity, closeness and position of x are weakly positive related to the number of goals and goal difference, and that there is a weak negative relation with the number of conceded goals. This suggests that stronger teams with better attack and defense abilities have in general larger values for the above mentioned metrics.

We also found that some network metrics are less related to the performance of a team. The eigencentrality, betweenness and the average change in x do not differ per match outcome. Correlation analysis also showed no dependence between goal statistics and the eigencentrality, betweenness, dispersion, and average change in x and y. Finally, cluster analysis revealed four distinct groups of observations based on the network metrics. They are again related to the final ranking, and thus performance of the team. The matches of Manchester City are mostly assigned to one cluster, with the best values for the previously mentioned network metrics. This implies that Manchester City has a unique playing style which is clearly related to their good performance.

8 Results Predictive Analysis

In this section we present the results for the goal-based and result-based models. We provide the performance metrics for the models and show variable importance plots for the random forest models.

8.1 Goal-based models

Regression models are implemented for the prediction of the number of goals scored per team. The models under consideration are linear regression (LR), linear regression with forward variable selection (LR forward), linear regression with backward variable selection (LR backward), Poisson regression, quasi-Poisson regression, Lasso regression, Ridge regression, MARS, random forest (RF), gradient boosting trees (GB) and extreme gradient boosting (XGB). The models are run for the three feature selection methods (no feature selection, Lasso regression, the correlation method) with and without the multicollinearity threshold, resulting in 66 different model combinations. We also include the two baseline methods, described in Section 5.2. Models are built on both the set of predictor variables excluding and including the network metrics. The MAE and MSE are reported in Table 18 for the models excluding the network metrics as predictor variables. For each model, we only report the best feature selection and multicollinearity combination, ordered by MAE.

Table 18: The lowest Mean Absolute Error (MAE) and Mean Squared Error (MSE) per model.

Method	**Feature selection**	**Multicollinearity threshold**	**MAE**	**MSE**
RF	None	False	0.887	1.296
GB	Lasso	False	0.889	1.297
MARS	None	False	0.898	1.4
XGB	None	True	0.913	1.388
Lasso regression	Lasso	True	0.922	1.362
Ridge regression	Lasso	True	0.941	1.393
LR	Lasso	False	0.963	1.476
Baseline 2	-	-	1.037	1.727
LR backward	Lasso	True	1.099	2.004
LR forward	Lasso	False	1.102	1.979
Poisson regression	Lasso	False	1.265	2.664
quasi-Poisson regression	Lasso	False	1.265	2.664
Baseline 1	-	-	1.367	3.478

The model with the lowest MAE and MSE is the random forest, with a MAE of 0.887 and MSE of 1.296. The random forest has an almost similar performance to the gradient boosting trees, with a MAE of 0.889 and MSE of 1.297. The RF, GB, XGB, Lasso regression, MARS, Ridge regression and LR all outperform the baseline methods. Their predictive performance appears to be quite similar. The first baseline method is the worst performing method, with a MAE of 1.367 and MSE of 3.478. Baseline 2 is performing slightly better, which is as we expected since it uses past team information.

Table 19 presents the MAE and the MSE for the models including the network metrics as predictor variables. The best model is again the random forest, with a MAE of 0.882 and MSE of 1.269. Similar as previously, the RF, GB, XGB, Lasso regression, MARS and Ridge regression have a better predictive performance than the baseline methods. Comparing the predictive performance of the models with and without the network metrics, we find that the results are quite similar. The random forest including the network metrics has a MAE that is 0.005 smaller and a MSE that is 0.027 smaller than the random forest excluding the network metrics. For the other models including the network metrics, the MAE and MSE are also lower, but the differences are very small.

Table 19: The lowest Mean Absolute Error (MAE) and Mean Squared Error (MSE) per model. Network metrics are included as predictor variables.

Method	**Feature selection**	**Multicollinearity threshold**	**MAE**	**MSE**
RF	Correlation	False	0.882	1.269
GB	Lasso	True	0.883	1.286
XGB	None	True	0.893	1.297
Lasso regression	None	False	0.889	1.272
MARS	None	False	0.898	1.4
Ridge regression	Lasso	True	0.912	1.338
Baseline 2	-	-	1.037	1.727
LR forward	Lasso	False	1.109	1.981
LR backward	Lasso	True	1.15	2.143
LR	Correlation	True	1.16	1.93
quasi-Poisson regression	Lasso	True	1.27	2.653
Poisson regression	Lasso	False	1.27	2.655
Baseline 1	-	-	1.35	3.206

Figure 7 shows the variable importance of the 25 most important predictors for the random forest model excluding network metrics.

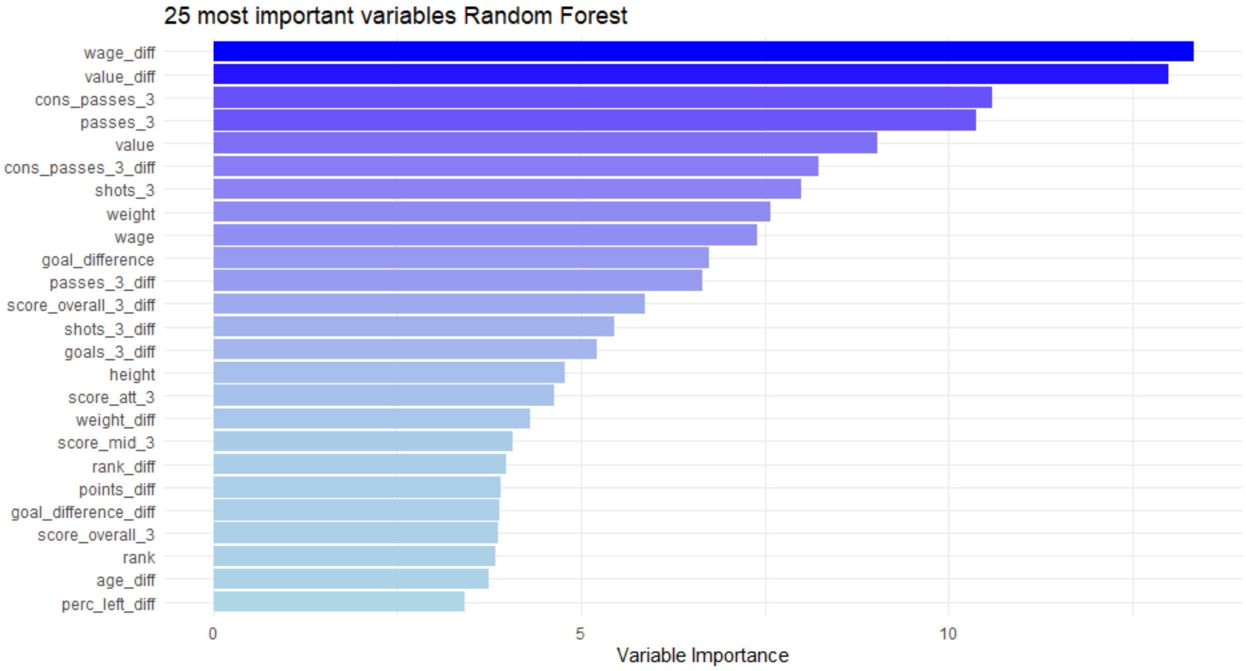


Figure 7: Variable importance plot for the Random Forest model excluding network metrics.

The wage and value difference between the competing teams are the most important variables for the prediction of the number of goals scored. The average passes and consecutive passes over the past three matches are also considered important. For the random forest including network metrics, the 25 most important variables are shown in Figure 8. Besides the wage and value differences, there are many network metrics present in the most important variables. The algebraic connectivity over the past three matches is most important. Other network metrics include the closeness scores, the binary average shortest path and the clustering coefficient.

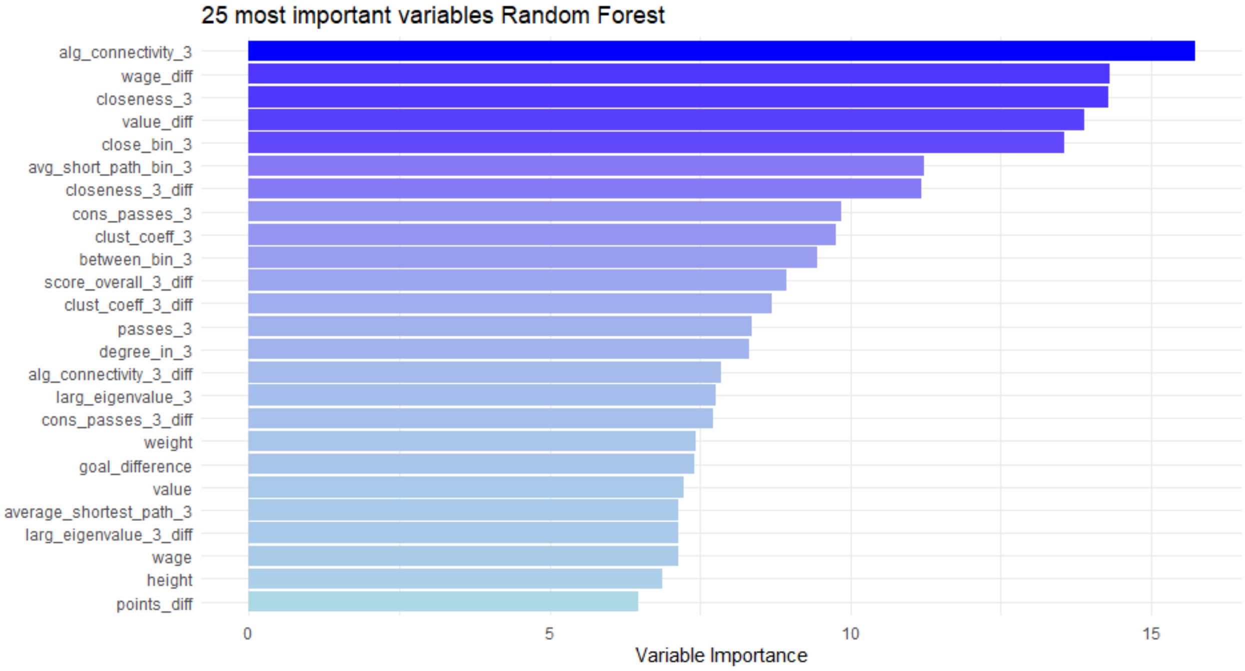


Figure 8: Variable importance plot for the Random Forest model including network metrics.

8.2 Result-based models

For the modelling of match outcome, different classification models were implemented. The predictive performance of the methods on the set of predictor variables excluding and including the network metrics are given in Table 20 and Table 21, respectively. The tables are ordered by accuracy and weighted F1. For the classification models excluding the network metrics as predictor variables, the best performing models in terms of accuracy are the SVM with a sigmoid and linear kernel. Their accuracy is equal to 0.567 with a 95% confidence interval between 0.458 and 0.671. The Naive Bayes model has the highest weighted F1 score. When only considering win and loss outcomes, the SVM with a sigmoid kernel has the highest F1 score. The F1 scores for the win and loss are in general higher than the weighted F1 score that also takes into account draws. All implemented models, except the XGB and Multinomial model, outperform the baseline methods. However, when considering the confidence intervals for the classification methods, most overlap. The second baseline method has a quite good performance, which was as expected. The baseline method does not predict any draws, but when only considering wins and losses it has a weighted F1 of 0.601. The third baseline method has the worst predictive performance.

Table 20: Predictive performance metrics for the best feature selection and multicollinearity threshold combination per model, excluding network metrics. Performance metrics include the accuracy, weighted F1, weighted precision and weighted recall.

Method	**Feature selection**	**Multicollinearity threshold**	**Accuracy**	**CI lower**	**CI upper**	**Weighted F1**	**Weighted precision**	**Weighted recall**	**Win loss F1**	**Win loss precision**	**Win loss recall**
SVM linear	None	False	0.567	0.458	0.671	0.542	0.566	0.567	0.623	0.624	0.671
SVM sigmoid	None	False	0.567	0.458	0.671	0.507	0.55	0.567	0.627	0.612	0.714
Naive Bayes	Correlation	False	0.544	0.436	0.65	0.559	0.592	0.544	0.616	0.675	0.571
GB	None	False	0.544	0.436	0.65	0.538	0.615	0.544	0.593	0.702	0.586
RF	Correlation	False	0.544	0.436	0.65	0.531	0.542	0.533	0.559	0.602	0.614
SVM radial	None	True	0.544	0.436	0.65	NA	NA	0.544	0.596	0.598	0.7
SVM polynomial	None	True	0.533	0.425	0.639	0.452	0.607	0.533	0.555	0.637	0.671
Baseline 2	-	-	0.533	0.425	0.639	NA	NA	0.533	0.601	0.536	0.686
Baseline 1	-	-	0.511	0.404	0.618	0.512	0.517	0.511	0.536	0.547	0.529
XGB	Lasso	True	0.511	0.404	0.618	0.5	0.501	0.511	0.568	0.565	0.586
Multinomial	Lasso	True	0.511	0.404	0.618	0.497	0.512	0.511	0.557	0.563	0.586
Baseline 3	-	-	0.367	0.268	0.475	0.377	0.396	0.367	0.413	0.448	0.386

Figure 9 shows the confusion matrices for the models with the highest accuracy: the SVM with a linear kernel and the SVM with a sigmoid kernel. The SVM linear predicts more draws compared to the SVM sigmoid, while the SVM sigmoid predicts more wins. The confusion matrices for the other models can be found in Appendix I.

	Predicted Class		
True Class	Win	Draw	Loss
Win	32	3	4
Draw	14	4	2
Loss	12	4	15

(a) SVM linear.

	Predicted Class		
True Class	Win	Draw	Loss
Win	35	0	4
Draw	17	1	2
Loss	14	2	15

(b) SVM sigmoid.

Figure 9: Confusion matrices for the two best performing models in terms of accuracy: SVM linear sigmoid and SVM sigmoid.

Table 21 shows the results for the classification models including the network metrics. From this table we can see that the SVM sigmoid has the highest accuracy; the random forest has the highest weighted F1. However, we again see that the 95% confidence intervals for the accuracy also overlap. When only considering win and loss outcomes, the SVM sigmoid has the highest weighted F1. The second baseline method outperforms many of the implemented models. Figure 10 shows the confidence matrices for the SVM sigmoid and random forest. The SVM sigmoid does not predict any draws, but correctly predicts more wins and losses.

Table 21: Predictive performance metrics for the best feature selection and multicollinearity threshold combination per model. Network metrics are included as predictor variables. Performance metrics include the accuracy, weighted F1, weighted precision and weighted recall.

Method	Feature selection	Multicollinearity threshold	Accuracy	CI lower	CI upper	Weighted F1	Weighted precision	Weighted recall	Win loss F1	Win loss precision	Win loss recall
SVM sigmoid	Lasso	True	0.578	0.469	0.681	NA	NA	0.578	0.649	0.584	0.743
RF	Correlation	False	0.544	0.436	0.650	0.532	0.555	0.544	0.594	0.618	0.614
GB	None	True	0.533	0.425	0.639	0.522	0.558	0.533	0.569	0.612	0.586
SVM radial	None	True	0.533	0.425	0.639	NA	NA	0.533	0.584	0.566	0.686
Baseline 2	-	-	0.533	0.425	0.639	NA	NA	0.533	0.601	0.536	0.686
Naive Bayes	None	True	0.511	0.403	0.618	0.51	0.509	0.511	0.582	0.579	0.586
Multinomial	Lasso	True	0.489	0.382	0.597	0.476	0.481	0.489	0.563	0.561	0.586
XGB	Lasso	True	0.5	0.393	0.607	0.493	0.504	0.5	0.55	0.566	0.557
SVM linear	Lasso	False	0.5	0.393	0.607	0.467	0.472	0.5	0.562	0.55	0.614
SVM polynomial	Lasso	False	0.5	0.393	0.607	NA	0.489	0.5	0.548	0.629	0.643
Baseline 1	-	-	0.433	0.329	0.542	0.433	0.434	0.433	0.502	0.507	0.5
Baseline 3	-	-	0.367	0.268	0.475	0.377	0.396	0.367	0.413	0.448	0.386

	Predicted Class		
True Class	Win	Draw	Loss
Win	32	0	7
Draw	15	0	5
Loss	11	0	20

(a) SVM sigmoid.

	Predicted Class		
True Class	Win	Draw	Loss
Win	30	6	3
Draw	11	6	3
Loss	12	6	13

(b) Random Forest.

Figure 10: Confusion matrices for the two best performing models in terms of accuracy: SVM sigmoid and Random Forest.

The 25 most important variables for the random forest model excluding network metrics are shown in Figure 11. The wage difference between the competing teams is the most important variable, followed by the difference in average number of shots, overall score and number of passes over the past three matches.

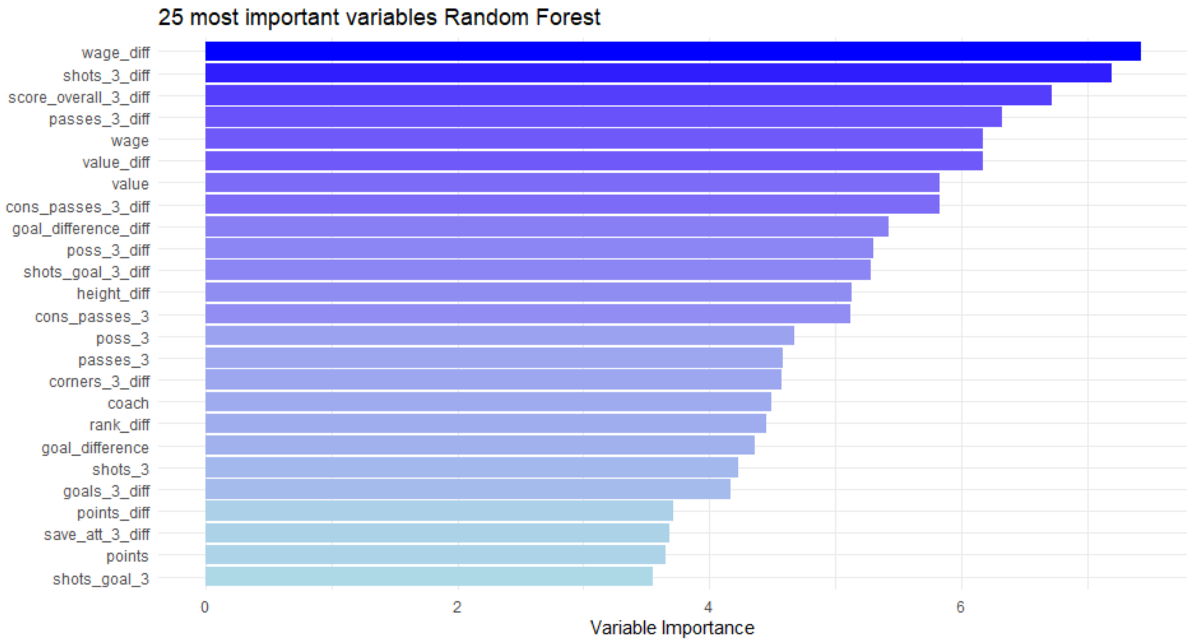


Figure 11: Variable importance plot for the Random Forest model excluding network metrics.

For the random forest model including network metrics, the 25 most important variables are shown in Figure 12. The difference in wage between the competing teams is again the most important variable, followed by the difference in average overall score, number of shots and possession over the past three matches and the difference in value between the teams. The 25 most important variables do include some network metrics, however, their variable importance is as not as high as the previously mentioned variables.

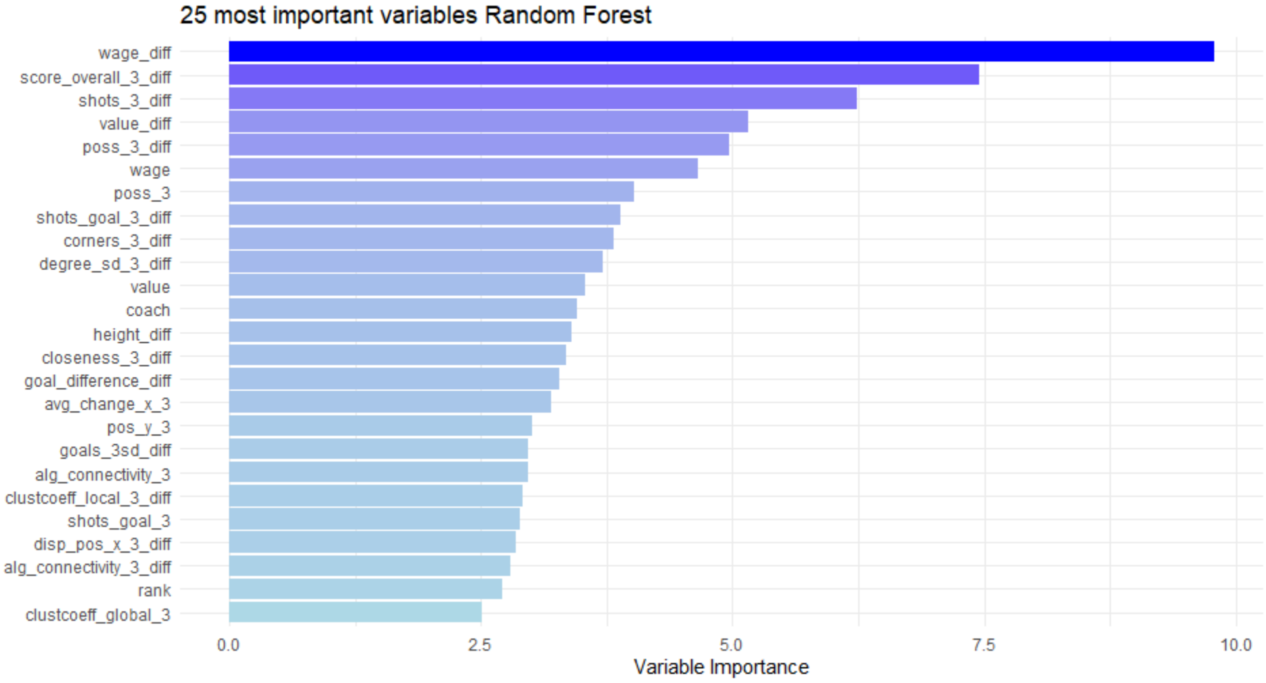


Figure 12: Variable importance plot for the Random Forest model including network metrics.

9 Discussion Predictive Analysis

For the prediction of the number of goals scored per team, several regression models were implemented. Performance was compared between the models including and excluding the network metrics as predictor variables. Overall, the random forest has the best predictive performance. On average, the random forest has predictions that are approximately 0.15 closer to the actual number of goals compared to the best baseline method. For both the models with and without the network metrics, the random forest, gradient boosting, extreme gradient boosting, Lasso regression, MARS and Ridge regression all outperform the two baseline methods. Differences are however very small. The models do have better performance than the linear and Poisson regression models. This could be caused by the fact that tree-based models can better capture non-linear relationships and feature interactions. Also, it seems that the Lasso and Ridge regression and MARS can better deal with the non-linear relationships, feature interactions and noisy features that might be present in the data. The Poisson distribution does not seem to fit the number of scored goals well.

Incorporating the network metrics as predictor variables does not improve predictive performance much for the goal-based models. The MAE is 0.005 smaller for the random forest with the network metrics. Even though the performance is slightly better, the difference seems negligible. Model complexity increases when incorporating additional variables, which can possibly lead to overfitting, an increase in computational speed and noise when the variables are irrelevant. Therefore, we believe that not incorporating the network metrics as predictor variables warrants the quality of the regression models.

For the prediction of the match outcome, we also compared classification methods excluding and including the network metrics as predictor variables. Without network metrics, the SVM with a linear kernel has the best predictive performance for all match outcomes, while the SVM with a sigmoid kernel has the best performance when only considering win and loss outcomes. The performance metrics for wins and losses are higher compared to the metrics also considering draws. This suggests that the models have more difficulty with the prediction of draws. Incorporating the network metrics as predictor variables does not improve predictive performance much. The accuracy of the SVM with a sigmoid kernel is 0.011 higher, but the weighted F1 score is lower when incorporating network metrics. The win loss F1 score is in fact 0.026 higher when incorporating the network metrics. This suggests that the models including network metrics might be better at predicting win and loss outcomes but worse at predicting draws. Most implemented classification models have higher predictive performance metrics than the baseline methods, but the 95% confidence intervals for the accuracy mostly overlap. Also, the second baseline method already has quite good predictive performance. Overall, it seems challenging to predict match outcome and incorporating network metrics does not boost predictive performance. The simple Naive Bayes model already provides relatively good results.

The random forest models for both regression and classification allow for measuring variable importance. For the goal-based models, the difference in the wage and values of the players between the competing teams are considered very important. These variables might provide a good indication of the strength of a team. When incorporating network metrics, the algebraic connectivity is the most important variable. Thus, the existence of independent groups in the team is considered important. The closeness scores also seem important for the prediction of the number of goals; the average closeness, binary closeness and difference in closeness between the competing teams are included in the top most important variables. Even though incorporating the network metrics does not improve the predictive performance much for the goal-based models, many network metrics are included in the 25 most important variables. Contrary to the goal-based models, the most important variables of the random forest for the prediction of match outcome do not contain many network metrics. Again the difference in wage between the competing teams is most important. Perhaps for the prediction of number of goals scored the network metrics have more predictive power than for the prediction of match outcome.

10 Limitations

There are a several limitations in this study that could be addressed in future research. First of all, we only analyse matches from the 2017-18 season of the Premier League. It would be interesting to expand the dataset to include more seasons of the Premier League, competitions of other countries, or competitions between national teams such as the World Cup. Perhaps more variation in number of goals scored or match outcome could be captured by using more data, improving model results. Second, we believe that there could be some adjustments made to the adjacency matrix. We do not consider the difficulty of a pass. It can occur that a team makes less passes, but that these passes are more difficult regarding for example the number of opponents nearby and distance. Considering pass difficulty when creating the adjacency matrix can provide a more realistic view of the performance of individual players and teams. Also, players are replaced by their substitutes to keep the passing network size constant. This makes it harder to actually study the behaviour of a player that does not play the full match. Here, the adjacency matrix could be weighted by dividing by the minutes played per player so that the edges represent the number of passes per minute.

We also see some possibilities for further research regarding the predictive analysis. There could be experimented with the number of matches that features are averaged over. Features not known before the beginning of a match, including the network metrics and previous match features, are averaged over the past three matches. It could also be useful to be able to make predictions for matches further into the future. In this research we only consider predictions for matches that will take place the same week. Also, some papers in the literature consider the ordinal aspect of the match outcome. For match outcomes, a win is better than a draw and a draw is better than a loss. These models might outperform the currently used classification models. An interesting model to implement would be the ordinal forest (Hornung, 2019). Lastly, we suggest exploring betting strategies and calculating betting returns for the different prediction methods. This enables quantifying the actual improvements of prediction models over betting odds. Also, including betting odds as predictor variables might improve predictive performance.

Topics that have not been thoroughly addressed in the field of network science for football analysis include dynamics of the passing networks and the interactions between teams. Buldú et al. (2018) give an extensive review of the challenges of using network science for football analysis. In our research, the passing networks are averaged over the whole match. Considering the dynamics of the passing networks, for example by using a sliding window, has not much been investigated. This allows for analysing the evolution of the network metrics and performance of a team over a match. We also do not consider interactions between the two competing teams. It would be very interesting to construct two interacting passing networks. This results in multilayer networks, where the intra-layer links would be composed of the passes only within each team, while the inter-layer links would be based on ball recovery/losses (Buldú et al., 2018). This allows for studying the adaptability of the teams to its opponent, which is not possible when studying the passing networks separately.

11 Conclusion

In this research we have performed network analysis to study the behaviour of football teams. The main goal was to investigate the relation between network metrics and team performance. We also investigated whether network metrics have predictive power for the prediction of number of goals and match outcome. The largest open collection football dataset provided by Wyscout was used in combination with collected features. Matches from the Premier League 2017-18 season were analysed, which includes 380 matches for 20 teams.

First, we compared network metrics according to match outcome and team. We found that winning teams have greater values for the clustering coefficient, largest eigenvalue, algebraic connectivity, position of x and mean degree, and a smaller value for the average shortest path. Also, teams with a higher final ranking have a larger closeness score and average change in x. Most of these metrics are positively related to the number of goals and goal difference, and negatively related to the number of conceded goals. Cluster analysis revealed four distinct groups of observations based on the network data. Especially Manchester City appears to have a unique playing style that is related to their good performance. Overall, teams seem to have different behaviour that can be characterized by network metrics and most of these network metrics are clearly related to team performance.

To investigate the predictive power of the network metrics, we implemented several regression (goal-based) and classification (result-based) models for the prediction of the number of goals and match outcome. The random forest provides the best predictive performance for the prediction of the number of goals scored with a MAE of 0.887 excluding network metrics and 0.882 including network metrics. For the prediction of the match outcome, the SVM provides the best performance with an accuracy of 0.567 and 0.578 excluding and including the network metrics, respectively. For both modelling approaches, incorporating the network metrics as predictor variables does not improve predictive performance much. Even though some network metrics are considered important by the random forest, we believe that the network metrics do not have enough predictive power to be included as predictor variables. Also, the implemented models outperform the baseline methods, but the differences are very small. It still seems like a challenging task to predict outcomes in soccer.

Overall, we believe that using network analysis to study the behaviour of football teams can be very useful. It allows for identifying key players, quantifying the contribution of players to the team and assessing the variability of interactions between players. Also, with the knowledge of the relation between the network metrics and performance, football professionals can adapt their team tactics to optimize performance. For future research on this topic, we suggest expanding the dataset to more seasons or other competitions, experimenting with dynamics and interactions between competing teams in passing networks and further exploring the modelling of football match outcomes by considering the ordinal aspect of the data, including betting odds and making predictions for matches further into the future.

References

Michael Patrick Allen. The problem of multicollinearity. *Understanding regression analysis*, pages 176–180, 1997.

Rodrigo Aquino, João Cláudio Machado, Filipe Manuel Clemente, Gibson Moreira Praça, Luiz Guilherme C Gonçalves, Bruno Melli-Neto, João Victor S Ferrari, Luiz H Palucci Vieira, Enrico F Puggina, and Christopher Carling. Comparisons of ball possession, match running performance, player prominence and team network properties according to match outcome and playing formation during the 2018 fifa world cup. *International Journal of Performance Analysis in Sport*, 19 (6):1026–1037, 2019.

Rahul Baboota and Harleen Kaur. Predictive analysis and modelling football results using machine learning approach for english premier league. *International Journal of Forecasting*, 35(2):741–755, 2019.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

Leo Breiman. Prediction games and arcing algorithms. *Neural computation*, 11(7):1493–1517, 1999.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Javier M Buldú, Javier Busquets, Johann H Martínez, José L Herrera-Diestra, Ignacio Echegoyen, Javier Galeano, and Jordi Luque. Using network science to analyse football passing networks: Dynamics, space, time, and the multilayer nature of the game. *Frontiers in psychology*, 9:1900, 2018.

Javier M Buldu, J Busquets, Ignacio Echegoyen, et al. Defining a historic football team: Using network science to analyze guardiola's fc barcelona. *Scientific reports*, 9(1):1–14, 2019.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

Paolo Cintia, Fosca Giannotti, Luca Pappalardo, Dino Pedreschi, and Marco Malvaldi. The harsh rule of the goals: Data-driven performance indicators for football teams. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2015.

Filipe Manuel Clemente, Micael Santos Couceiro, Fernando Manuel Lourenço Martins, and Rui Sousa Mendes. Using network metrics in soccer: a macro-analysis. *Journal of human kinetics*, 45(1):123–134, 2015a.

Filipe Manuel Clemente, Fernando Manuel Lourenço Martins, Dimitris Kalamaras, P Del Wong, and Rui Sousa Mendes. General network analysis of national soccer teams in fifa world cup 2014. *International Journal of Performance Analysis in Sport*, 15(1):80–96, 2015b.

Carlos Cotta, Antonio M Mora, Juan Julián Merelo, and Cecilia Merelo-Molina. A network analysis of the 2010 fifa world cup champion team play. *Journal of Systems Science and Complexity*, 26(1): 21–42, 2013.

Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

Mark J Dixon and Stuart G Coles. Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46 (2):265–280, 1997.

Jordi Duch, Joshua S Waitzman, and Luís A Nunes Amaral. Quantifying the performance of individual players in a team activity. *PloS one*, 5(6), 2010.

David Dyte and Stephen R Clarke. A ratings based poisson model for world cup soccer simulation. *Journal of the Operational Research society*, 51(8):993–998, 2000.

James D Evans. *Straightforward statistics for the behavioral sciences.* Thomson Brooks/Cole Publishing Co, 1996.

Ronald A Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2): 337–407, 2000.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Jerome H Friedman et al. Multivariate adaptive regression splines. *The annals of statistics*, 19(1): 1–67, 1991.

José Gama, Pedro Passos, Keith Davids, Hugo Relvas, João Ribeiro, Vasco Vaz, and Gonçalo Dias. Network analysis and intra-team activity in attacking phases of professional football. *International Journal of Performance Analysis in Sport*, 14(3):692–708, 2014.

John Goddard. Regression models for forecasting goals and match results in association football. *International Journal of forecasting*, 21(2):331–340, 2005.

Daniel Goller, Michael C Knaus, Michael Lechner, Gabriel Okasa, et al. Predicting match outcomes in football by an ordered forest estimator. *Economics Working Paper Series*, (1811), 2018.

Bruno Gonçalves, Diogo Coutinho, Sara Santos, Carlos Lago-Penas, Sergio Jiménez, and Jaime Sampaio. Exploring team passing networks and player movement dynamics in youth association football. *PloS one*, 12(1), 2017.

Peter Gould and Anthony Gatrell. A structural analysis of a game: the liverpool v manchester united cup final of 1977. *Social Networks*, 2(3):253–273, 1979.

Andreas Groll, Thomas Kneib, Andreas Mayr, and Gunther Schauberger. On the dependency of soccer scores–a sparse bivariate poisson model for the uefa european football championship 2016. *Journal of Quantitative Analysis in Sports*, 14(2):65–79, 2018a.

Andreas Groll, Christophe Ley, Gunther Schauberger, and Hans Van Eetvelde. Prediction of the fifa world cup 2018-a random forest approach with an emphasis on estimated team ability parameters. *arXiv preprint arXiv:1806.03208*, 2018b.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Roman Hornung. Ordinal forests. *Journal of Classification*, pages 1–14, 2019.

David C Howell. *Statistical methods for psychology.* Cengage Learning, 2009.

Alexandros Karatzoglou, David Meyer, and Kurt Hornik. Support vector machines in r. *Journal of statistical software*, 15(9):1–28, 2006.

Dimitris Karlis and Ioannis Ntzoufras. Analysis of sports data by using bivariate poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(3):381–393, 2003.

William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

Alan J Lee. Modeling scores in the premier league: is manchester united really the best? *Chance*, 10(1):15–19, 1997.

Howard Levene. Robust tests for equality of variances. *Contributions to probability and statistics. Essays in honor of Harold Hotelling*, pages 279–292, 1961.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Michael J Maher. Modelling association football scores. *Statistica Neerlandica*, 36(3):109–118, 1982.

Ian G McHale and Samuel D Relton. Identifying key players in soccer teams using network analysis and pass difficulty. *European Journal of Operational Research*, 268(1):339–347, 2018.

Rodrigo Azuero Melo, David Zarruk, Maintainer Rodrigo Azuero Melo, and XML Imports RCurl. Package ‘gmapsdistance’. *The Comprehensive R Archive Network (CRAN)*, 2017.

Rui Sousa Mendes, Filipe Manuel Clemente, and Fernando Manuel Lourenço Martins. Network analysis of portuguese team on fifa world cup 2014. *E-balonmano. com: Revista de Ciencias del Deporte*, 11(2):225–226, 2015.

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, Friedrich Leisch, CC Chang, and CC Lin. e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien, 2018. *R package version*, pages 1–7, 2019.

Maintainer Stephen Milborrow. Package 'earth'. *R Software package*, 2019.

Nielsen Sports. World football report, 2018. `https://nielsensports.com/wp-content/uploads/2014/12/Nielsen_World-Football-2018-6.11.18.pdf`.

Luca Pappalardo, Paolo Cintia, Paolo Ferragina, Emanuele Massucco, Dino Pedreschi, and Fosca Giannotti. Playerank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10 (5):1–27, 2019a.

Luca Pappalardo, Paolo Cintia, Alessio Rossi, Emanuele Massucco, Paolo Ferragina, Dino Pedreschi, and Fosca Giannotti. A public data set of spatio-temporal match events in soccer competitions. *Scientific data*, 6(1):1–15, 2019b.

Javier López Pena and Hugo Touchette. A network theory analysis of football strategies. *arXiv preprint arXiv:1206.6904*, 2012.

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

Havard Rue and Oyvind Salvesen. Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(3):399–418, 2000.

Gunther Schauberger and Andreas Groll. Predicting matches in international football tournaments with random forests. *Statistical Modelling*, 18(5-6):460–482, 2018.

Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Alkeos Tsokos, Santhosh Narayanan, Ioannis Kosmidis, Gianluca Baio, Mihai Cucuringu, Gavin Whitaker, and Franz Király. Modeling outcomes of soccer matches. *Machine Learning*, 108(1): 77–95, 2019.

UEFA. `uefa.com`, 2020. Accessed: 13-05-2020.

Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.

Jay M Ver Hoef and Peter L Boveng. Quasi-poisson vs. negative binomial regression: how should we model overdispersed count data? *Ecology*, 88(11):2766–2772, 2007.

Qing Wang, Hengshu Zhu, Wei Hu, Zhiyong Shen, and Yuan Yao. Discerning tactical patterns for professional soccer teams: an enhanced topic model with applications. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2197–2206, 2015.

Duncan J Watts. The "new" science of networks. *Annu. Rev. Sociol.*, 30:243–270, 2004.

Bernard L Welch. The generalization ofstudent's' problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35, 1947.

Marvin N Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *arXiv preprint arXiv:1508.04409*, 2015.

Appendices

A Event types in the Wyscout dataset

Table 22: Event types with their subevents and most common tags.

Event	Subevent	Tags
Pass	Simple pass, high pass, head pass, smart pass, launch, cross, hand pass	Accurate, not accurate, interception, assist, right foot, left foot, blocked key pass, own goal
Foul	Foul, hand foul, late card foul, out of game foul, protest, stimulation, time lost foul, violent foul	No card, yellow, red, second yellow
Shot	Shot	Goal, accurate, not accurate, block, opportunity, assist
Duel	Air duel, ground attacking duel, ground defending duel, ground loose ball duel	Accurate, not accurate, lost, won, sliding tackle
Free kick	Corner, free kick, free kick cross, free kick shot, goal kick, penalty, throw in	Accurate, not accurate, high, opportunity, assist, goal
Offside	Offside	
Save attempt	Save attempt, reflexes	Accurate, not accurate, goal, counter attack
Others on the ball	Acceleration, Clearance, Touch	Accurate, not accurate, interception, counter attack
Goalkeeper leaving line		Accurate, not accurate, opportunity, right foot, left foot
Interruption	Ball out of field, whistle	

B Available Wyscout datasets

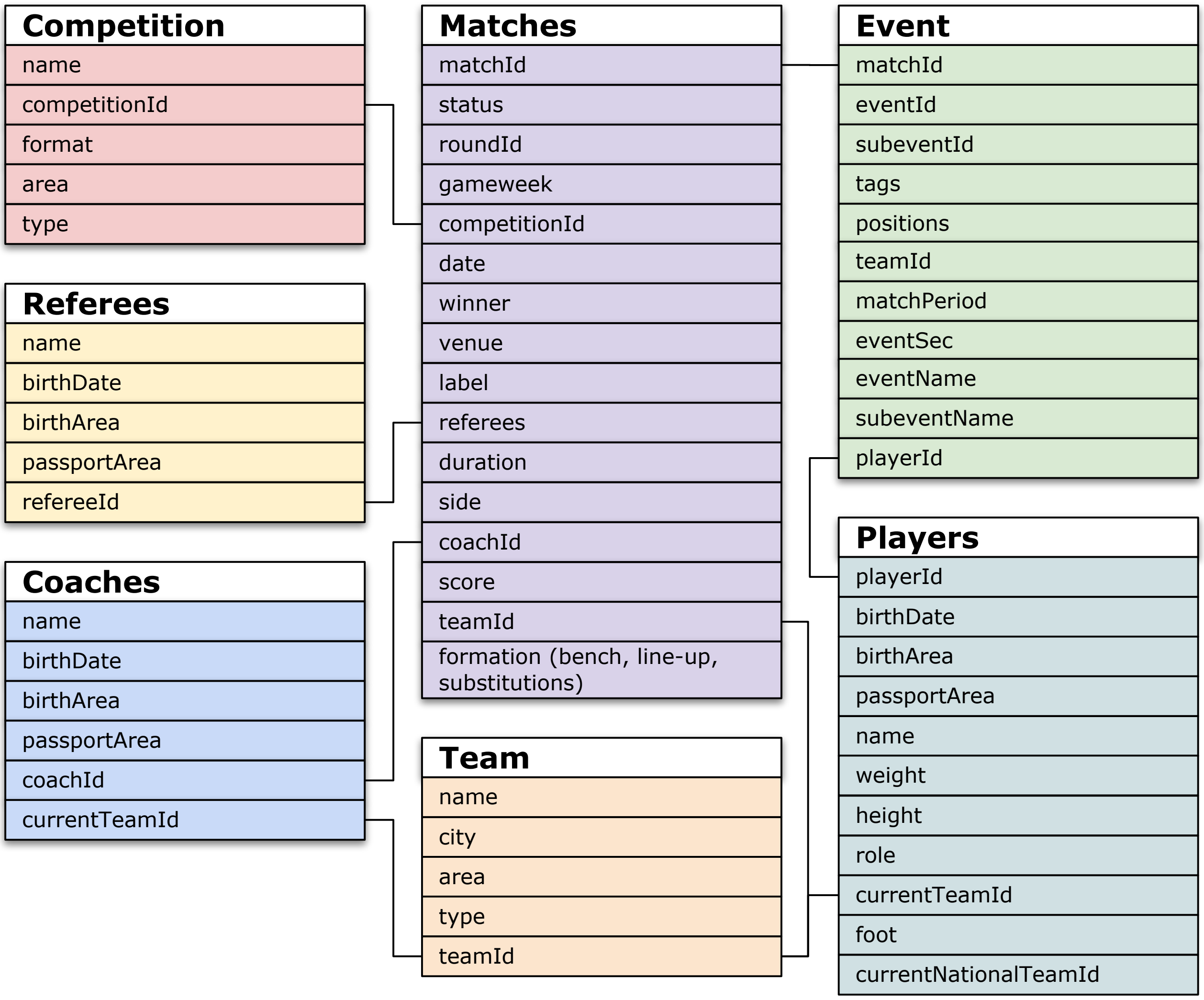
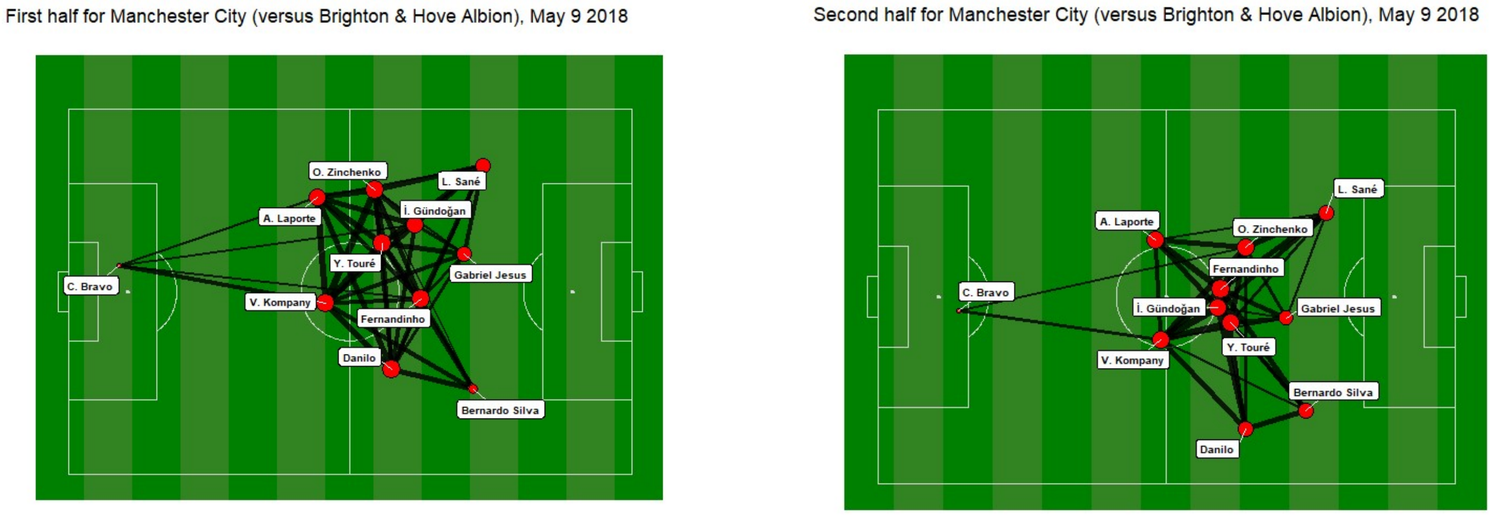


Figure 13: An overview of the available Wyscout datasets.

C Visualization passing network



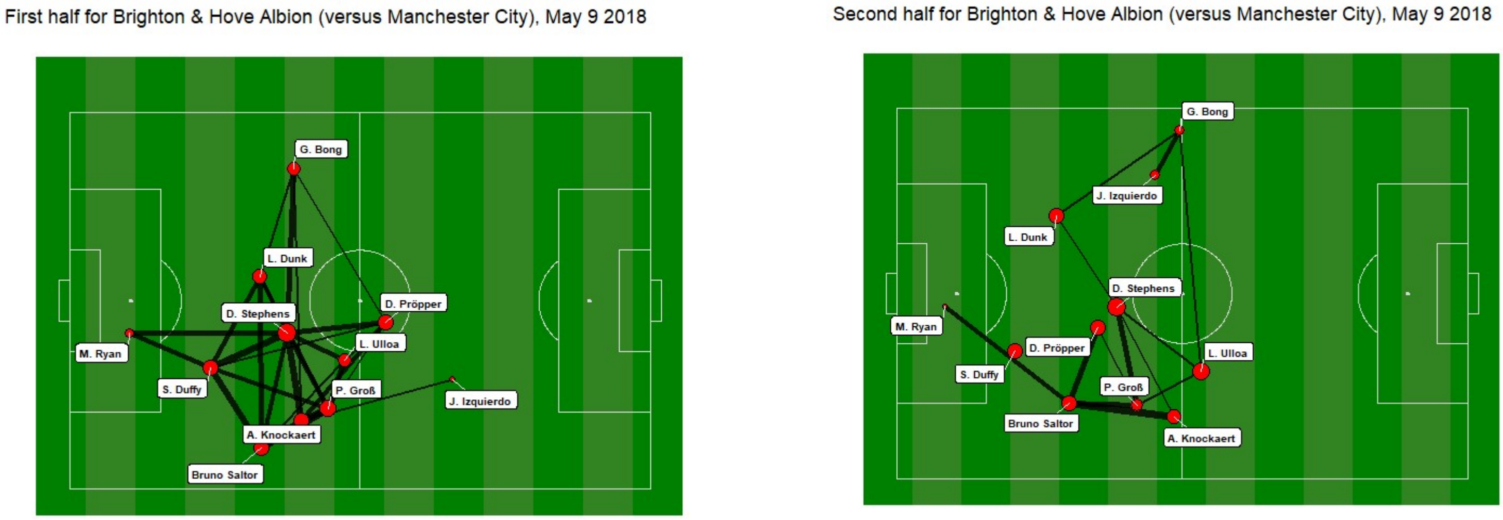
First half for Manchester City (versus Brighton & Hove Albion), May 9 2018

Second half for Manchester City (versus Brighton & Hove Albion), May 9 2018

(a) First half.

(b) Second half.

Figure 14: Undirected passing network for the first and second half of Manchester City during the Manchester City - Brighton & Hove Albion match.



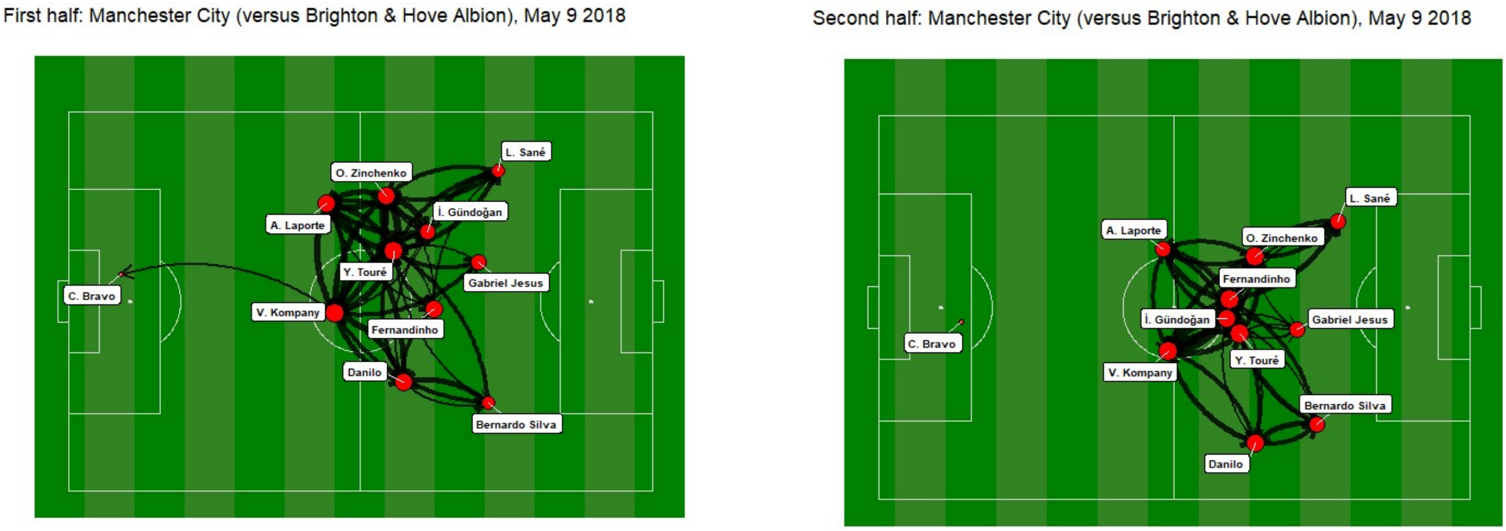
First half for Brighton & Hove Albion (versus Manchester City), May 9 2018

Second half for Brighton & Hove Albion (versus Manchester City), May 9 2018

(a) First half.

(b) Second half.

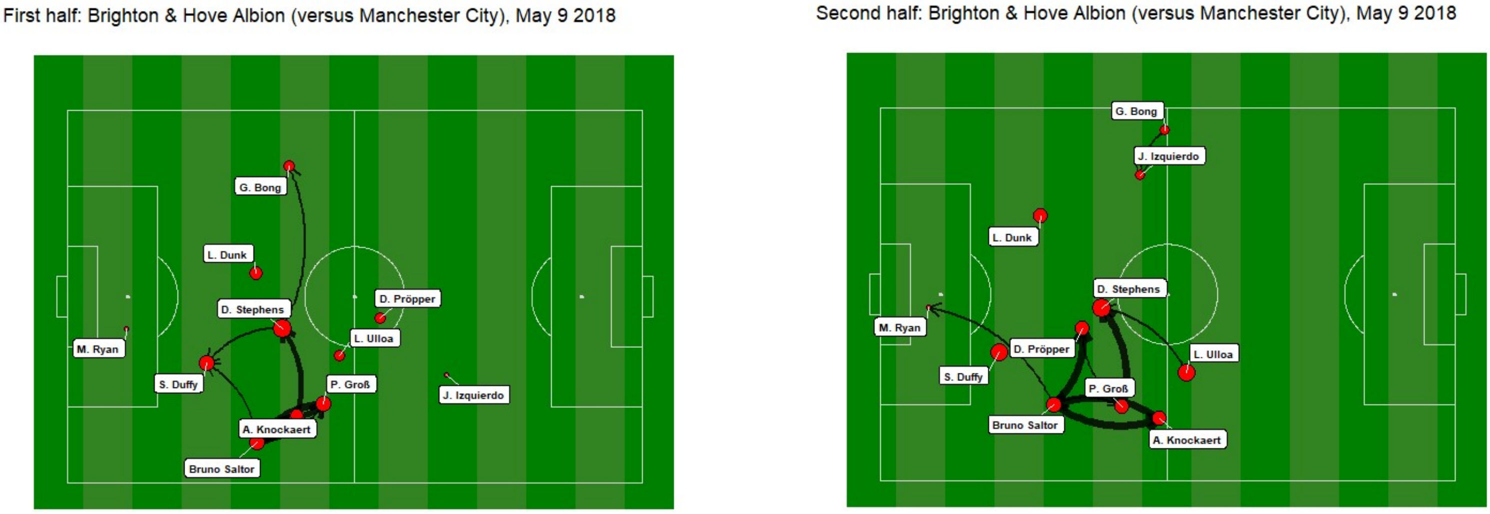
Figure 15: Undirected passing network for the first and second half of Brighton & Hove Albion during the Manchester City - Brighton & Hove Albion match.



(a) First half.

(b) Second half.

Figure 16: Directed passing network for the first and second half of Manchester City during the match Manchester City - Brighton & Hove Albion.



(a) First half.

(b) Second half.

Figure 17: Directed passing network for the first and second half of Brighton & Hove Albion during the match Manchester City - Brighton & Hove Albion.

D Comparison network metrics according to team.

Table 23: Mean (SD) values, p-value and corresponding statistical test for the comparison of the network metrics per team.

	Swansea City	Arsenal	Chelsea	Manchester United	Liverpool	Newcastle United	Southampton	Everton	Tottenham Hotspur	Manchester City	West Bromwich Albion
Clustering coefficient	3.02 (1.18)	5.34 (1.77)	4.77 (1.90)	4.41 (1.77)	5.32 (1.96)	2.34 (1.13)	3.27 (1.32)	2.56 (0.89)	4.69 (1.71)	7.27 (2.49)	2.14 (0.86)
Clustering coefficient local	0.31 (0.05)	0.28 (0.02)	0.28 (0.03)	0.28 (0.03)	0.28 (0.03)	0.30 (0.03)	0.31 (0.04)	0.31 (0.04)	0.28 (0.02)	0.27 (0.02)	0.33 (0.05)
Clustering coefficient global	0.83 (0.07)	0.89 (0.03)	0.86 (0.05)	0.84 (0.08)	0.88 (0.04)	0.78 (0.08)	0.86 (0.04)	0.83 (0.06)	0.89 (0.04)	0.92 (0.03)	0.79 (0.08)
Largest eigenvalue	25.66 (8.98)	44.96 (13.02)	39.50 (13.55)	36.84 (13.89)	44.83 (14.88)	19.97 (9.20)	28.73 (10.35)	22.55 (7.37)	39.87 (12.06)	59.22 (16.54)	18.42 (6.22)
Algebraic connectivity	6.32 (2.62)	11.09 (4.19)	10.26 (3.72)	7.28 (3.39)	11.20 (3.68)	5.51 (2.27)	4.62 (2.48)	6.26 (2.76)	8.79 (3.06)	14.47 (4.18)	3.93 (2.15)
Max eigencentrality	0.47 (0.04)	0.45 (0.04)	0.48 (0.04)	0.47 (0.05)	0.47 (0.04)	0.48 (0.05)	0.45 (0.03)	0.46 (0.04)	0.45 (0.03)	0.46 (0.04)	0.48 (0.04)
Sd eigencentrality	0.12 (0.02)	0.12 (0.02)	0.12 (0.02)	0.12 (0.02)	0.13 (0.02)	0.12 (0.02)	0.12 (0.02)	0.12 (0.02)	0.12 (0.02)	0.13 (0.02)	0.12 (0.02)
Eigencentrality	0.28 (0.01)	0.28 (0.01)	0.28 (0.01)	0.28 (0.01)	0.27 (0.01)	0.28 (0.01)	0.28 (0.01)	0.28 (0.01)	0.28 (0.01)	0.28 (0.01)	0.28 (0.01)
Closeness	0.17 (0.04)	0.24 (0.04)	0.22 (0.04)	0.22 (0.05)	0.23 (0.05)	0.15 (0.04)	0.18 (0.03)	0.16 (0.03)	0.22 (0.04)	0.29 (0.05)	0.14 (0.03)
Betweenness	8.66 (1.19)	7.83 (1.06)	8.89 (1.40)	8.22 (0.98)	8.85 (1.27)	8.14 (1.16)	8.13 (1.06)	8.21 (0.99)	7.95 (1.10)	8.09 (1.28)	8.24 (1.09)
Average shortest path	0.49 (0.24)	0.29 (0.08)	0.31 (0.11)	0.37 (0.17)	0.30 (0.10)	0.61 (0.24)	0.46 (0.13)	0.53 (0.20)	0.32 (0.07)	0.23 (0.05)	0.63 (0.19)
Position x	28.07 (4.58)	34.75 (4.86)	31.11 (4.43)	31.97 (5.01)	32.85 (4.39)	24.52 (4.44)	29.54 (4.02)	27.03 (3.66)	32.90 (4.35)	37.64 (4.75)	24.30 (4.29)
Position y	32.06 (3.72)	34.82 (3.16)	34.04 (3.73)	32.73 (4.92)	33.45 (4.22)	27.38 (4.67)	29.59 (3.07)	28.43 (3.80)	33.95 (3.06)	35.62 (3.64)	25.69 (4.15)
Dispersion	30.44 (1.43)	29.98 (0.92)	30.02 (1.28)	30.04 (1.03)	29.54 (0.89)	29.54 (1.43)	30.19 (1.13)	29.82 (1.34)	30.26 (0.95)	29.65 (1.61)	29.39 (1.96)
Dispersion position x	26.57 (2.03)	27.79 (1.17)	27.42 (1.79)	27.37 (1.46)	26.94 (1.32)	26.47 (1.58)	27.73 (1.42)	27.31 (1.60)	27.90 (1.40)	27.31 (2.08)	26.86 (2.35)
Average change x	0.51 (0.67)	1.75 (0.56)	1.44 (0.62)	1.14 (0.77)	1.26 (0.62)	1.02 (0.49)	1.04 (0.63)	0.62 (0.57)	1.37 (0.38)	1.27 (0.56)	0.56 (0.61)
Average change y	0.33 (0.69)	-0.07 (0.51)	0.12 (0.73)	-0.01 (0.57)	0.29 (0.49)	0.26 (0.88)	0.22 (0.84)	0.13 (0.65)	0.21 (0.60)	0.23 (0.54)	-0.04 (0.94)
Closeness binary	0.08 (0.01)	0.08 (0.00)	0.08 (0.00)	0.08 (0.01)	0.08 (0.00)	0.07 (0.01)	0.08 (0.00)	0.08 (0.00)	0.08 (0.00)	0.09 (0.00)	0.07 (0.01)
Betweenness binary	3.27 (1.34)	2.11 (0.42)	2.57 (0.66)	2.89 (1.20)	2.29 (0.53)	3.87 (1.24)	3.00 (0.66)	3.36 (0.96)	2.28 (0.53)	1.62 (0.44)	3.82 (1.07)
Average shortest path binary	1.33 (0.13)	1.21 (0.04)	1.26 (0.07)	1.29 (0.12)	1.23 (0.05)	1.39 (0.12)	1.30 (0.07)	1.34 (0.10)	1.23 (0.05)	1.16 (0.04)	1.39 (0.11)
Mean degree	6.84 (0.99)	7.90 (0.41)	7.46 (0.61)	7.21 (0.97)	7.72 (0.53)	6.28 (0.92)	7.05 (0.62)	6.74 (0.79)	7.72 (0.53)	8.38 (0.44)	6.23 (0.86)
Sd degree	1.75 (0.30)	1.65 (0.35)	1.61 (0.32)	1.64 (0.34)	1.54 (0.28)	1.71 (0.30)	1.75 (0.32)	1.65 (0.26)	1.57 (0.38)	1.54 (0.33)	1.69 (0.23)
Max degree	9.24 (0.85)	9.79 (0.41)	9.66 (0.58)	9.32 (0.90)	9.84 (0.44)	8.92 (1.08)	9.26 (0.64)	9.13 (0.81)	9.53 (0.60)	9.95 (0.23)	8.79 (0.74)

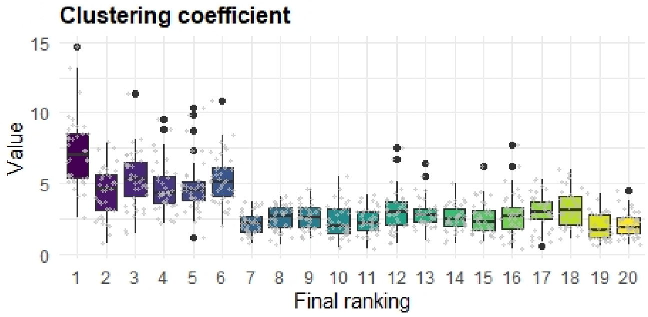
Crystal Palace		West Ham United	Stoke City	Watford	Burnley	Brighton & Hove Albion	AFC Bournemouth	Huddersfield Town	p	test	Difference
2.37 (0.90)	2.65 (0.93)	2.55 (1.12)	1.94 (0.96)	2.98 (1.05)	2.23 (0.78)	2.57 (0.89)	3.07 (1.42)	2.80 (1.52)	<0.001	Approximate method of Welch	Yes
0.79 (0.08)	0.84 (0.05)	0.80 (0.07)	0.78 (0.07)	0.81 (0.05)	0.80 (0.06)	0.82 (0.06)	0.81 (0.07)	0.80 (0.09)	<0.001	Kruskal-Wallis	Yes
0.31 (0.05)	0.32 (0.05)	0.31 (0.04)	0.34 (0.06)	0.30 (0.04)	0.32 (0.05)	0.31 (0.04)	0.30 (0.04)	0.32 (0.05)	<0.001	Approximate method of Welch	Yes
20.98 (7.21)	23.15 (7.46)	22.45 (8.37)	17.36 (7.51)	24.85 (7.32)	19.63 (6.01)	22.80 (7.00)	25.89 (9.81)	22.82 (10.97)	<0.001	Approximate method of Welch	Yes
4.78 (2.39)	4.46 (2.39)	5.04 (2.77)	4.60 (2.98)	4.69 (1.98)	4.50 (2.20)	5.52 (2.51)	5.50 (2.02)	6.59 (3.27)	<0.001	Approximate method of Welch	Yes
0.46 (0.04)	0.46 (0.04)	0.48 (0.05)	0.47 (0.04)	0.46 (0.09)	0.45 (0.04)	0.46 (0.05)	0.47 (0.04)	0.48 (0.04)	<0.001	Kruskal-Wallis	Yes
0.12 (0.02)	0.12 (0.02)	0.13 (0.02)	0.12 (0.02)	0.12 (0.03)	0.12 (0.02)	0.11 (0.02)	0.14 (0.02)	0.13 (0.02)	<0.001	One-way analysis of means	Yes
0.28 (0.01)	0.28 (0.01)	0.28 (0.01)	0.28 (0.01)	0.27 (0.05)	0.28 (0.01)	0.28 (0.01)	0.27 (0.01)	0.27 (0.01)	<0.001	Kruskal-Wallis	Yes
0.15 (0.04)	0.16 (0.03)	0.16 (0.03)	0.14 (0.04)	0.17 (0.03)	0.15 (0.03)	0.16 (0.03)	0.16 (0.03)	0.16 (0.05)	<0.001	Approximate method of Welch	Yes
8.43 (1.05)	8.26 (1.03)	8.25 (1.00)	8.29 (0.99)	8.35 (1.10)	8.16 (0.95)	8.73 (1.12)	8.85 (1.27)	8.17 (1.04)	<0.001	Kruskal-Wallis	Yes
0.57 (0.20)	0.53 (0.14)	0.53 (0.16)	0.67 (0.24)	0.50 (0.13)	0.60 (0.14)	0.51 (0.15)	0.51 (0.17)	0.58 (0.28)	<0.001	Kruskal-Wallis	Yes
26.47 (4.34)	26.29 (3.78)	27.39 (3.91)	24.85 (4.95)	27.53 (3.54)	25.95 (3.34)	26.57 (3.33)	26.57 (4.72)	25.60 (5.21)	<0.001	One-way analysis of means	Yes
26.57 (4.34)	27.18 (4.25)	27.46 (3.59)	24.93 (4.97)	29.69 (4.03)	26.48 (3.65)	29.92 (4.18)	27.76 (3.92)	28.11 (4.22)	<0.001	Approximate method of Welch	Yes
29.53 (1.33)	29.42 (1.08)	29.77 (1.05)	28.94 (1.80)	30.64 (1.40)	29.54 (1.12)	30.26 (0.92)	29.82 (1.48)	30.20 (1.27)	<0.001	Approximate method of Welch	Yes
27.73 (1.72)	26.78 (1.55)	27.21 (1.76)	27.01 (2.12)	27.73 (1.46)	27.09 (1.35)	26.66 (1.66)	27.11 (1.92)	26.91 (2.19)	<0.001	Approximate method of Welch	Yes
1.03 (0.60)	1.05 (0.48)	0.65 (0.71)	0.73 (0.72)	0.95 (0.63)	0.38 (0.76)	0.91 (0.62)	1.23 (0.56)	0.58 (0.78)	<0.001	One-way analysis of means	Yes
0.24 (0.88)	0.29 (0.69)	0.03 (0.69)	0.23 (0.74)	0.23 (0.81)	-0.01 (0.73)	0.37 (0.64)	0.45 (0.82)	0.20 (0.69)	0.037	Approximate method of Welch	Yes
0.07 (0.01)	0.08 (0.00)	0.08 (0.01)	0.07 (0.01)	0.07 (0.01)	0.07 (0.00)	0.08 (0.00)	0.08 (0.00)	0.07 (0.01)	<0.001	Kruskal-Wallis	Yes
3.83 (1.11)	3.30 (0.82)	3.48 (0.88)	4.11 (1.16)	3.52 (0.67)	3.64 (0.72)	3.41 (0.76)	3.58 (1.01)	3.68 (1.35)	<0.001	Kruskal-Wallis	Yes
1.39 (0.11)	1.33 (0.08)	1.35 (0.09)	1.41 (0.12)	1.35 (0.07)	1.37 (0.07)	1.34 (0.08)	1.36 (0.10)	1.37 (0.15)	<0.001	Kruskal-Wallis	Yes
6.31 (0.86)	6.80 (0.71)	6.49 (0.81)	6.04 (0.96)	6.55 (0.56)	6.38 (0.72)	6.63 (0.67)	6.54 (0.80)	6.38 (1.28)	<0.001	Approximate method of Welch	Yes
1.75 (0.30)	1.74 (0.35)	1.63 (0.28)	1.58 (0.37)	1.79 (0.34)	1.58 (0.30)	1.66 (0.32)	1.88 (0.30)	1.66 (0.37)	<0.001	One-way analysis of means	Yes
8.79 (0.81)	9.13 (0.62)	8.84 (0.89)	8.45 (1.08)	9.18 (0.61)	8.71 (0.90)	8.82 (0.93)	9.11 (0.89)	8.89 (1.25)	<0.001	Kruskal-Wallis	Yes

E Final ranking

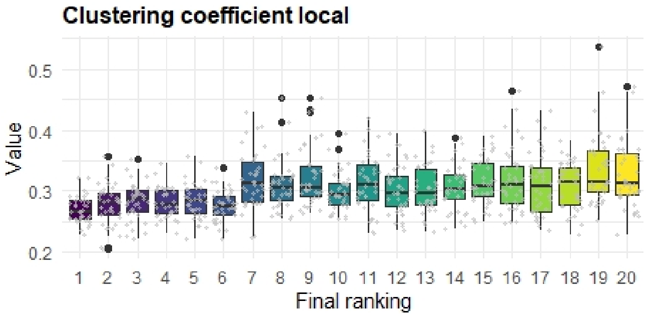
Table 24: Final ranking Premier League 2017-2018 season.

Team Id	**Name**	**Rank**
1625	Manchester City	1
1611	Manchester United	2
1624	Tottenham Hotspur	3
1612	Liverpool	4
1610	Chelsea	5
1609	Arsenal	6
1646	Burnley	7
1623	Everton	8
1631	Leicester City	9
1613	Newcastle United	10
1628	Crystal Palace	11
1659	AFC Bournemouth	12
1633	West Ham United	13
1644	Watford	14
1651	Brighton & Hove Albion	15
1673	Huddersfield Town	16
1619	Southampton	17
10531	Swansea City	18
1639	Stoke City	19
1627	West Bromwich Albion	20

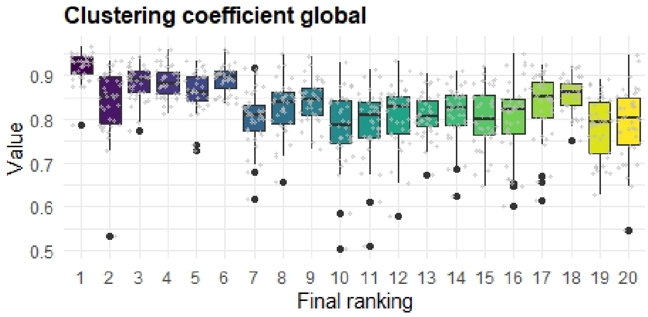
F Boxplots



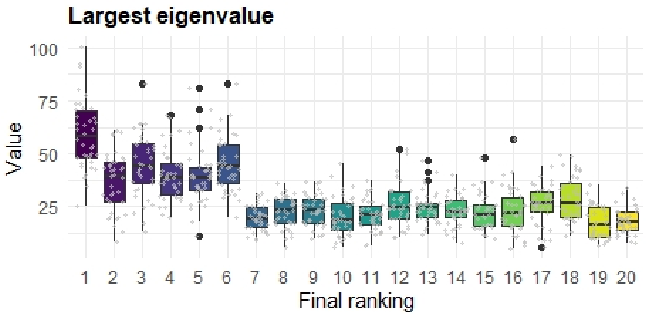
(a) Boxplot clustering coefficient.



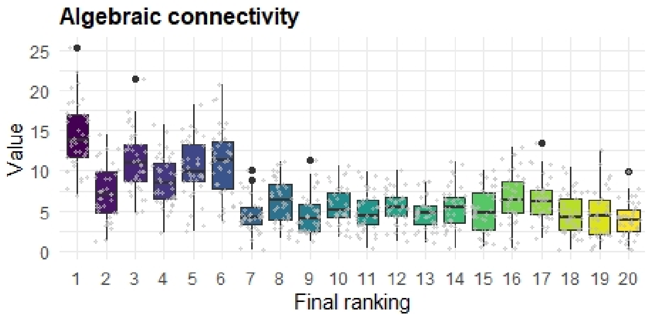
(b) Boxplot local clustering coefficient.



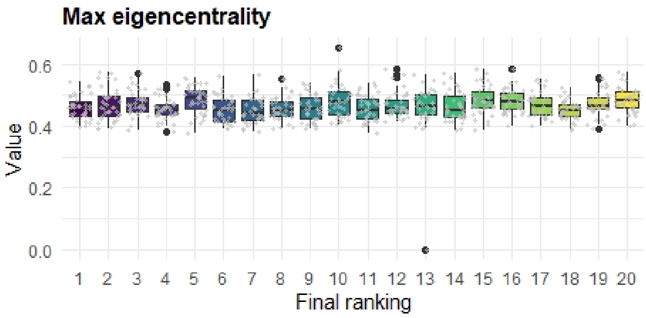
(c) Boxplot global clustering coefficient.



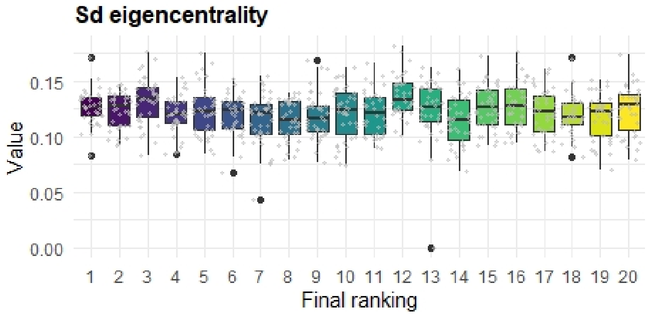
(d) Boxplot largest eigenvalue.



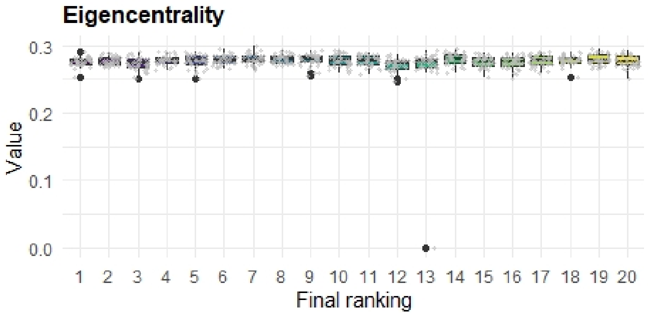
(e) Boxplot algebraic connectivity.



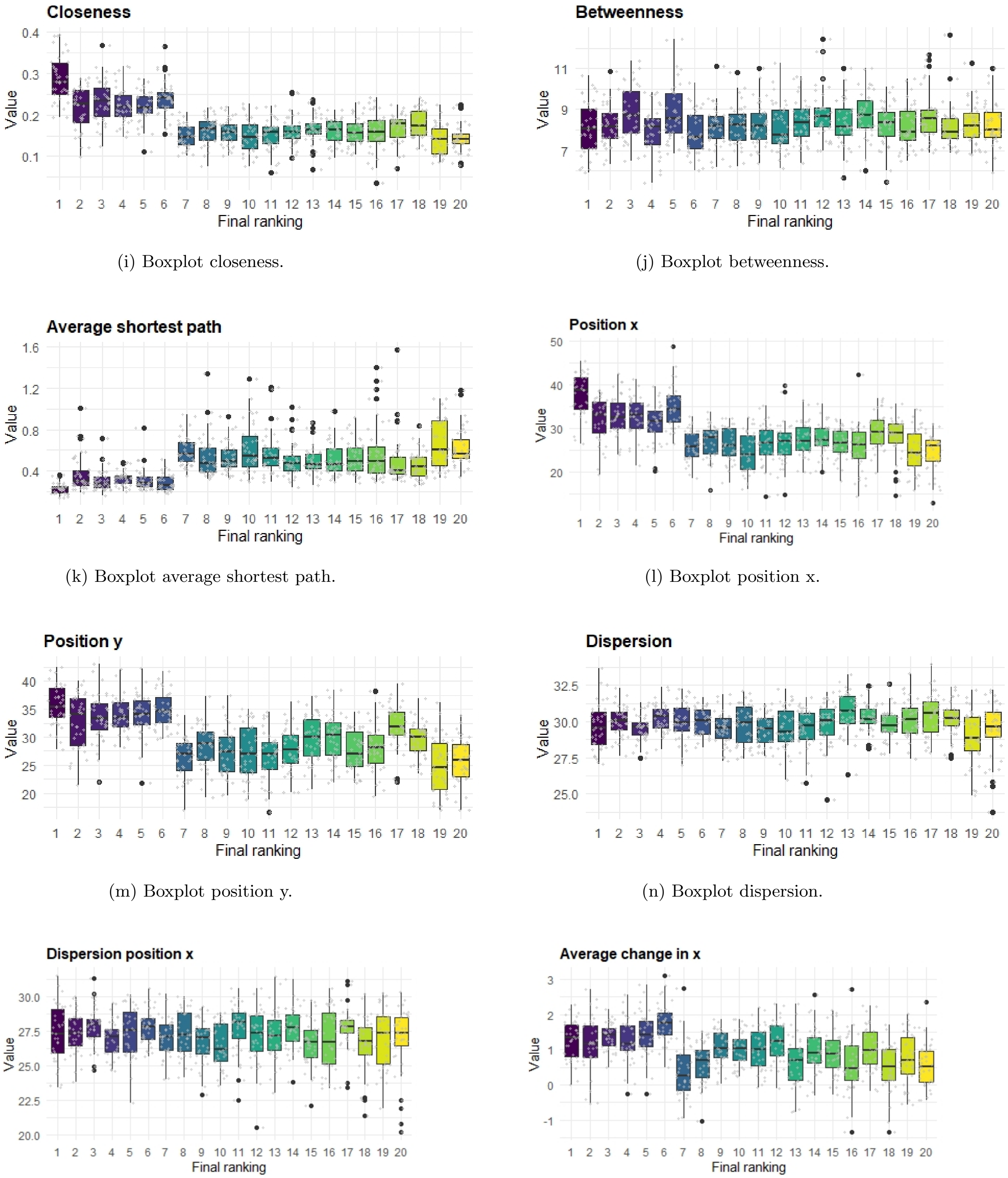
(f) Boxplot maximum eigencentrality.



(g) Boxplot standard deviation eigencentrality.



(h) Boxplot mean eigencentrality.



(i) Boxplot closeness.

(j) Boxplot betweenness.

(k) Boxplot average shortest path.

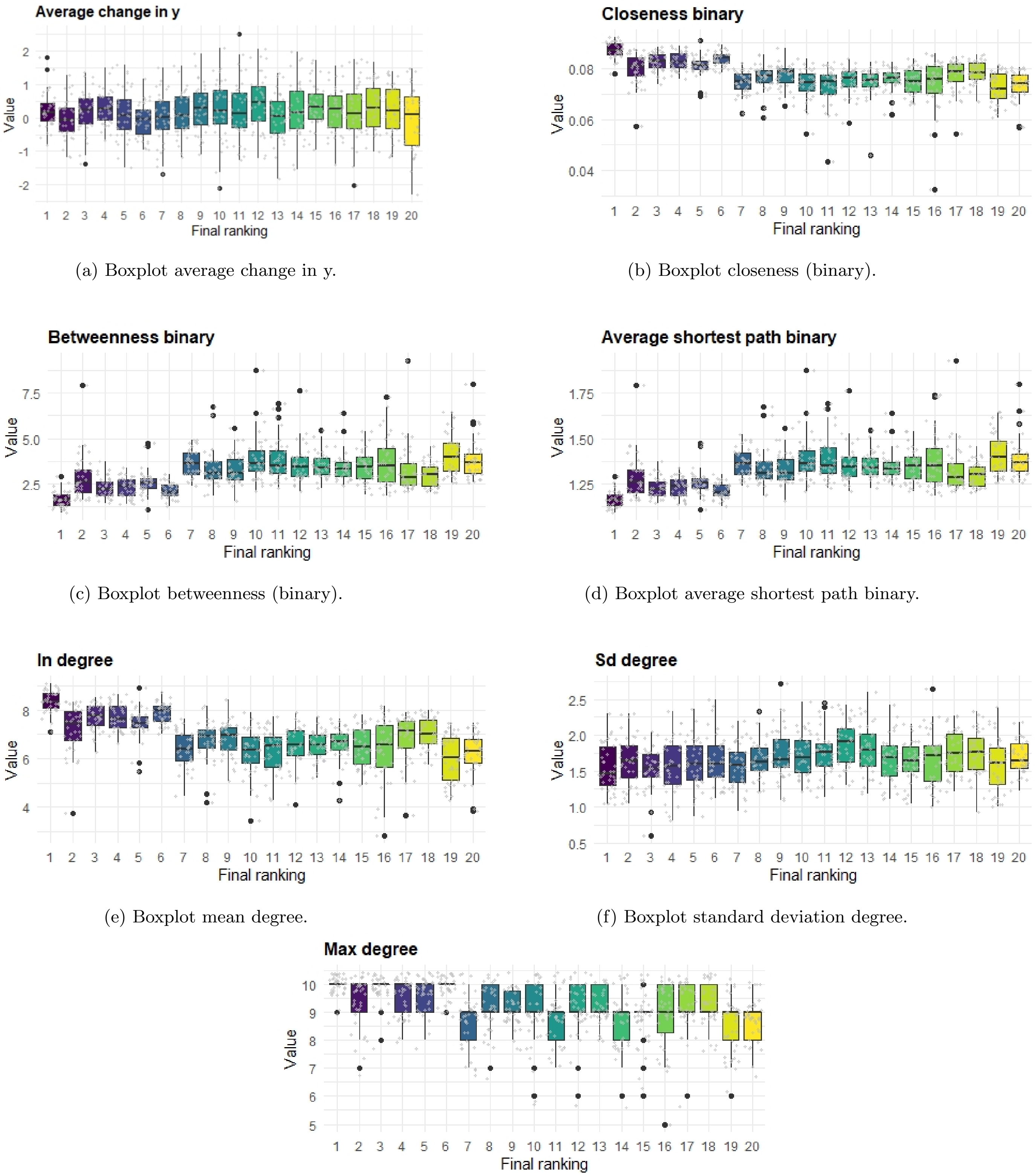
(l) Boxplot position x.

(m) Boxplot position y.

(n) Boxplot dispersion.

(o) Boxplot dispersion position x.

(p) Boxplot average change in x.



(a) Boxplot average change in y.

(b) Boxplot closeness (binary).

(c) Boxplot betweenness (binary).

(d) Boxplot average shortest path binary.

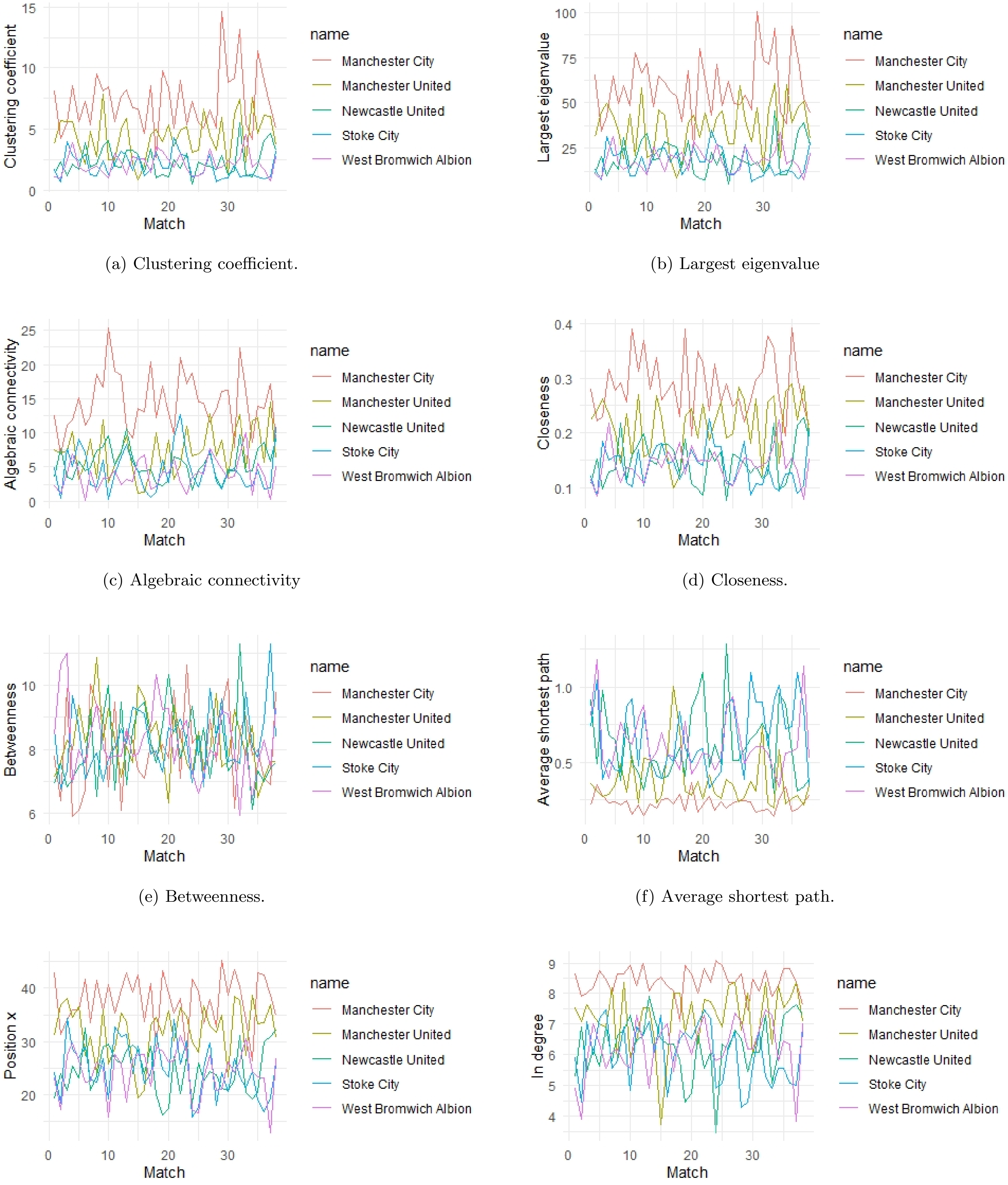
(e) Boxplot mean degree.

(f) Boxplot standard deviation degree.

(g) Boxplot maximum degree.

Figure 19: Boxplot for the network metrics per match grouped by team (rank). The rank is given in Appendix B.

G Season plots



(a) Clustering coefficient.

(b) Largest eigenvalue

(c) Algebraic connectivity

(d) Closeness.

(e) Betweenness.

(f) Average shortest path.

(g) Position x.

(h) Mean degree

Figure 20: Mean plots for several network metrics throughout the season. Values are reported for Manchester City, Manchester United, Newcastle United, Stoke City and West Bromwhich Albion, who ended 1st, 2nd, 10th, 19th and 20th in the season, respectively.

H Comparison network metrics according to team and win and loss outcome.

Table 25: Mean (SD) values for the comparison of the network metrics for the win and loss per team.

	Clustering coefficient			Clustering coefficient global			Clustering coefficient local			Largest eigenvalue			Algebraic connectivity			Max eigencentrality			Sd eigencentrality			Eigencentrality			Closeness			Betweenness			Average shortest path			Position x		
Team	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff
10531	2.954	3.166	No	0.839	0.831	No	0.329	0.289	Yes	25.129	26.599	No	5.641	6.513	No	0.477	0.467	No	0.13	0.119	No	0.274	0.279	No	0.165	0.174	No	9.572	8.166	Yes	0.478	0.458	No	28.267	28.242	No
1609	5.547	5.104	No	0.894	0.89	No	0.272	0.284	No	46.925	42.577	No	12.043	9.506	No	0.446	0.461	No	0.118	0.124	No	0.279	0.277	No	0.241	0.237	No	7.951	7.892	No	0.271	0.314	No	35.399	33.654	No
1610	5.07	4.094	No	0.859	0.873	No	0.284	0.289	No	41.612	34.849	No	10.393	9.841	No	0.484	0.471	No	0.129	0.113	No	0.274	0.281	No	0.223	0.209	No	9.079	8.635	No	0.311	0.327	No	32.197	29.257	Yes
1611	4.248	4.828	No	0.84	0.864	No	0.278	0.288	No	36.242	39.387	No	6.883	9.232	No	0.473	0.467	No	0.123	0.121	No	0.277	0.278	No	0.213	0.225	No	8.078	8.429	No	0.383	0.36	No	31.951	31.703	No
1612	5.558	5.661	No	0.891	0.888	No	0.285	0.275	No	46.561	48.271	No	11.957	8.927	No	0.479	0.462	No	0.133	0.134	No	0.273	0.273	No	0.241	0.233	No	8.782	9.295	No	0.287	0.299	No	33.135	34.597	No
1613	2.158	2.453	No	0.765	0.785	No	0.306	0.295	No	18.185	21.076	No	5.694	5.423	No	0.497	0.464	No	0.128	0.115	No	0.275	0.28	No	0.139	0.153	No	7.898	8.2	No	0.683	0.603	No	23.538	24.926	No
1619	3.926	3.031	No	0.868	0.858	No	0.306	0.31	No	33.849	27.72	No	5.085	4.879	No	0.443	0.444	No	0.119	0.118	No	0.279	0.28	No	0.202	0.173	No	8.097	7.959	No	0.405	0.476	No	31.516	29.219	No
1623	3.041	2.46	No	0.823	0.842	No	0.295	0.318	No	25.989	22.367	No	5.902	6.523	No	0.464	0.444	No	0.122	0.109	Yes	0.278	0.283	Yes	0.174	0.164	No	8.758	7.845	Yes	0.453	0.523	No	28.088	27.405	No
1624	4.894	3.764	No	0.889	0.88	No	0.285	0.286	No	41.197	33.662	No	8.994	6.682	No	0.465	0.455	No	0.123	0.115	No	0.277	0.281	No	0.221	0.208	No	8.138	7.491	No	0.317	0.365	No	32.9	31.914	No
1625	7.641	5.397	No	0.919	0.94	No	0.265	0.309	Yes	61.758	46.905	No	14.631	16.603	No	0.459	0.49	No	0.126	0.132	No	0.276	0.274	No	0.296	0.259	No	8.045	9.841	No	0.221	0.252	No	38.449	30.69	No
1627	1.359	2.359	Yes	0.706	0.827	Yes	0.362	0.322	No	12.566	20.398	Yes	1.574	4.33	Yes	0.486	0.477	No	0.13	0.125	No	0.273	0.276	No	0.113	0.149	Yes	8.36	8.077	No	0.804	0.587	No	21.465	24.966	No
1628	2.587	2.14	No	0.794	0.793	No	0.294	0.329	No	23.147	18.731	No	5.331	4.616	No	0.458	0.461	No	0.119	0.125	No	0.279	0.276	No	0.158	0.145	No	8.339	8.595	No	0.484	0.631	No	28.209	24.881	No
1631	2.852	2.092	Yes	0.846	0.839	No	0.306	0.344	No	24.843	19.171	No	5.475	3.758	No	0.465	0.456	No	0.119	0.116	No	0.279	0.28	No	0.169	0.143	Yes	8.383	8.192	No	0.485	0.61	No	27.756	23.815	Yes
1633	2.258	2.888	No	0.78	0.814	No	0.31	0.322	No	20.419	24.784	No	5.37	5.754	No	0.476	0.487	No	0.126	0.13	No	0.276	0.274	No	0.158	0.167	No	8.433	8.25	No	0.552	0.491	No	27.103	27.777	No
1639	1.735	2.047	No	0.76	0.797	No	0.336	0.33	No	15.556	18.385	No	4.067	5.108	No	0.471	0.472	No	0.113	0.119	No	0.281	0.278	No	0.13	0.143	No	7.689	8.32	No	0.723	0.628	No	22.238	25.729	No
1644	3.129	3.059	No	0.813	0.816	No	0.315	0.302	No	25.69	25.266	No	4.901	4.732	No	0.455	0.459	No	0.123	0.127	No	0.277	0.258	No	0.178	0.161	No	7.848	8.416	No	0.467	0.501	No	27.587	27.334	No
1646	2.226	2.437	No	0.798	0.806	No	0.307	0.304	No	19.135	22.036	No	4.126	4.369	No	0.448	0.445	No	0.114	0.116	No	0.281	0.279	No	0.145	0.159	No	7.936	8.155	No	0.604	0.551	No	26.231	26.684	No
1651	2.589	2.465	No	0.817	0.807	No	0.299	0.308	No	22.954	21.569	No	5.291	5.492	No	0.449	0.459	No	0.108	0.116	No	0.282	0.28	No	0.163	0.155	No	9.057	8.692	No	0.481	0.539	No	28.105	25.462	No
1659	2.808	3.113	No	0.797	0.802	No	0.316	0.288	No	23.982	25.794	No	5.151	5.507	No	0.483	0.46	No	0.137	0.134	No	0.271	0.272	No	0.156	0.164	No	9.202	8.358	No	0.537	0.525	No	25.99	25.955	No
1673	2.437	2.754	No	0.791	0.808	No	0.316	0.319	No	20.53	22.231	No	6.296	6.711	No	0.489	0.477	No	0.128	0.125	No	0.275	0.276	No	0.158	0.157	No	8.468	7.795	No	0.581	0.578	No	25.51	24.881	No

	Position y			Dispersion			Dispersion position x			Average change x			Average change y			Closeness binary			Betweenness binary			Average shortest path binary			In degree			Sd degree			Max degree		
Team	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff	Win	Loss	Diff
10531	33.294	32.01	No	31.329	30.139	No	26.8	26.456	No	0.371	0.689	No	0.241	0.365	No	0.076	0.078	No	3.386	3.026	No	1.339	1.303	No	6.648	7	No	1.83	1.752	No	9.125	9.381	No
1609	35.159	34.333	No	30	29.8	No	27.865	27.401	No	1.691	1.695	No	-0.198	-0.044	No	0.084	0.083	No	2.01	2.217	No	1.201	1.222	No	7.99	7.79	No	1.602	1.713	No	9.842	9.692	No
1610	34.431	33.665	No	29.965	29.858	No	27.509	26.904	No	1.371	1.34	No	0.13	-0.213	No	0.081	0.082	No	2.654	2.473	No	1.265	1.247	No	7.394	7.527	No	1.625	1.577	No	9.571	9.7	No
1611	32.804	32.981	No	29.986	29.992	No	27.32	27.08	No	1.092	1.108	No	-0.058	-0.181	No	0.08	0.082	No	2.876	2.571	No	1.288	1.257	No	7.229	7.506	No	1.619	1.519	No	9.32	9.286	No
1612	34.131	34.28	No	29.431	29.602	No	26.675	26.899	No	1.092	1.673	No	0.174	0.193	No	0.083	0.084	No	2.212	2.182	No	1.221	1.218	No	7.792	7.836	No	1.485	1.501	No	9.81	9.8	No
1613	26.26	27.992	No	29.474	29.427	No	26.308	26.393	No	1.191	0.977	No	0.54	0.113	No	0.073	0.074	No	4.083	3.823	No	1.408	1.382	No	6.061	6.364	No	1.716	1.665	No	8.5	9	No
1619	30.945	29.301	No	29.984	30.061	No	27.941	27.514	No	1.083	1.023	No	0.61	0.225	No	0.08	0.079	No	2.753	2.949	No	1.275	1.297	No	7.273	7.074	No	1.656	1.689	No	9.143	9.188	No
1623	29.912	28.765	No	30.169	29.673	No	27.394	27.278	No	0.606	0.793	No	0.215	0.099	No	0.077	0.077	No	3.224	3.248	No	1.322	1.325	No	6.839	6.855	No	1.642	1.597	No	9.077	9.2	No
1624	34.458	33.155	No	29.984	30.993	Yes	27.371	28.776	No	1.311	1.183	No	0.314	0.122	No	0.083	0.082	No	2.273	2.455	No	1.227	1.245	No	7.727	7.545	No	1.56	1.64	No	9.565	9.286	No
1625	36.073	32.132	No	29.538	27.39	Yes	27.142	25.056	No	1.266	1.124	No	0.159	-0.035	No	0.088	0.085	No	1.58	1.864	No	1.158	1.186	No	8.423	8.136	No	1.503	1.677	No	9.938	10	No
1627	25.098	26.129	No	29.554	29.409	No	26.229	26.905	No	0.043	0.834	Yes	-0.435	-0.112	No	0.066	0.076	Yes	4.939	3.388	No	1.514	1.341	Yes	5.152	6.617	Yes	1.718	1.694	No	8.333	8.842	No
1628	27.648	25.822	No	30.127	29.102	No	28.458	26.936	Yes	0.87	1.067	No	0.305	0.056	No	0.073	0.072	No	3.322	4.233	Yes	1.34	1.423	No	6.554	6.085	No	1.662	1.816	No	8.909	8.812	No
1631	27.416	25.14	No	29.165	29.234	No	26.939	26.468	No	0.895	1.144	No	0.085	0.395	No	0.078	0.075	No	3.098	3.606	No	1.31	1.361	No	6.977	6.533	No	1.608	1.771	No	9.167	9	No
1633	26.464	28.158	No	29.999	29.441	No	27.812	26.784	No	0.519	0.572	No	-0.207	-0.032	No	0.074	0.076	No	3.773	3.358	No	1.377	1.34	No	6.291	6.568	No	1.601	1.664	No	8.9	8.875	No
1639	22.419	26.118	No	27.45	29.389	No	25.429	27.284	No	0.696	0.723	No	0.419	0.147	No	0.07	0.074	No	4.091	3.904	No	1.422	1.392	No	5.779	6.263	No	1.536	1.642	No	8.286	8.684	No
1644	29.798	30.047	No	30.549	30.461	No	27.533	27.425	No	0.694	1.052	No	0.421	0.22	No	0.076	0.074	No	3.479	3.431	No	1.348	1.345	No	6.579	6.593	No	1.811	1.84	No	9.182	9.263	No
1646	25.946	28.216	No	29.553	29.754	No	27.329	26.771	No	0.403	0.282	No	-0.154	0.008	No	0.075	0.075	No	3.623	3.5	No	1.366	1.353	No	6.377	6.515	No	1.594	1.626	No	8.857	8.833	No
1651	29.951	30.503	No	30.321	30.074	No	27.171	25.838	No	0.74	0.947	No	0.71	0.135	Yes	0.076	0.075	No	3.404	3.568	No	1.34	1.359	No	6.667	6.483	No	1.599	1.625	No	8.444	8.812	No
1659	25.976	28.453	No	29.738	29.491	No	27.478	26.278	No	0.993	1.123	No	0.655	0.323	No	0.074	0.076	No	3.76	3.585	No	1.376	1.359	No	6.322	6.591	No	1.968	1.805	No	8.909	9.188	No
1673	26.634	28.391	No	30.584	29.865	Yes	27.897	26.255	No	0.359	0.578	No	0.263	0.03	No	0.074	0.074	No	3.889	3.531	No	1.389	1.363	No	6.242	6.469	No	1.678	1.648	No	9	8.947	No

I Confusion matrices for match outcome prediction.

I.1 Without network metrics

	Predicted Class		
True Class	Win	Draw	Loss
Win	21	13	5
Draw	4	9	7
Loss	4	8	19

(a) Naive Bayes.

	Predicted Class		
True Class	Win	Draw	Loss
Win	36	0	3
Draw	18	0	2
Loss	18	0	13

(b) SVM radial.

	Predicted Class		
True Class	Win	Draw	Loss
Win	30	11	10
Draw	8	8	10
Loss	1	1	11

(c) GB.

	Predicted Class		
True Class	Win	Draw	Loss
Win	30	5	4
Draw	10	6	4
Loss	11	7	13

(d) Random Forest.

	Predicted Class		
True Class	Win	Draw	Loss
Win	38	0	1
Draw	18	1	1
Loss	21	1	9

(e) SVM polynomial.

	Predicted Class		
True Class	Win	Draw	Loss
Win	28	7	4
Draw	8	5	7
Loss	12	6	13

(f) XGB.

	Predicted Class		
True Class	Win	Draw	Loss
Win	28	6	5
Draw	12	5	3
Loss	14	4	13

(g) Multinomial.

	Predicted Class		
True Class	Win	Draw	Loss
Win	22	10	7
Draw	7	9	4
Loss	13	9	15

(h) Baseline 1.

	Predicted Class		
True Class	Win	Draw	Loss
Win	17	9	13
Draw	6	6	8
Loss	8	13	10

(i) Baseline 3.

	Predicted Class		
True Class	Win	Draw	Loss
Win	26	0	13
Draw	11	0	9
Loss	9	0	22

(j) Baseline 2.

Figure 21: Confusion matrices for the implemented classification models excluding network metrics.

I.2 With network metrics

	Predicted Class		
True Class	Win	Draw	Loss
Win	35	0	4
Draw	17	0	3
Loss	18	0	13

(a) SVM radial.

	Predicted Class		
True Class	Win	Draw	Loss
Win	29	6	4
Draw	12	7	1
Loss	13	6	12

(b) GB.

	Predicted Class		
True Class	Win	Draw	Loss
Win	23	9	8
Draw	7	5	5
Loss	8	5	18

(c) Naive Bayes.

	Predicted Class		
True Class	Win	Draw	Loss
Win	24	5	10
Draw	10	6	4
Loss	11	5	15

(d) Multinomial.

	Predicted Class		
True Class	Win	Draw	Loss
Win	30	3	6
Draw	15	2	3
Loss	13	5	13

(e) SVM linear.

	Predicted Class		
True Class	Win	Draw	Loss
Win	27	7	5
Draw	10	6	4
Loss	11	8	12

(f) XGB.

	Predicted Class		
True Class	Win	Draw	Loss
Win	35	2	2
Draw	20	0	0
Loss	20	1	10

(g) SVM polynomial.

	Predicted Class		
True Class	Win	Draw	Loss
Win	24	7	8
Draw	8	4	8
Loss	9	11	11

(h) Baseline 1.

	Predicted Class		
True Class	Win	Draw	Loss
Win	17	9	13
Draw	6	6	8
Loss	8	13	10

(i) Baseline 3.

	Predicted Class		
True Class	Win	Draw	Loss
Win	26	0	13
Draw	11	0	9
Loss	9	0	22

(j) Baseline 2.

Figure 22: Confusion matrices for the implemented classification models including network metrics.