NTNU
Norwegian University of
Science and Technology

Beating the bookmakers

Using artificial neural networks to profit from football betting

Simon Borøy-Johnsen

Master of Science in Computer Science
Submission date: June 2017
Supervisor: Helge Langseth, IDI

Norwegian University of Science and Technology
Department of Computer Science

Summary

Artificial Neural Networks (ANNs) have throughout the years been used for several purposes. Problems spanning from image classification to text generation have all been subject to ANNs.

In this report, ANNs were used in order to predict the outcomes of football matches. Using data from the football statistics web site **www.whoscored.com**, ANNs were constructed in order to predict the outcomes of matches from two successive seasons of the English Premier League. The predictions were then used to decide whether or not to place bets on the outcomes, in an effort to generate a profit.

Several ANNs were constructed, utilizing data sources from player ratings to team characteristics. The networks were trained using simple back-propagation training. The predictions were then used together with odds from seven international bookmakers, trying to generate a profit from betting. Different money management (betting) strategies were applied, in order to highlight the importance of choosing correct bet sizes.

The results show that simple assumptions show promising results when predicting the outcome of a football match. The results also show that ANNs can indeed beat bookmakers in their own game, and gain a profit from football betting.

The report ends with the author's thoughts on how to further improve the profitability of the presented models.

Sammendrag

Nevrale nettverk har i en årrekke blitt brukt til flere forskjellige formål. Alt fra bildeklassifisering til tekstgenerering har blitt utprøvd ved hjelp av nevrale nettverk.

I denne rapporten ble nevrale nettverk brukt til å predikere resultatene til fotballkamper. Ved å bruke data fra fotballstatistikk-nettstedet **www.whoscored.com** , ble nevrale nettverk brukt for å predikere resultatene for kamper fra to sammenhengende sesonger av engelske Premier League. Prediksjonene ble deretter brukt til å avgjøre hvorvidt man skulle sette penger på resultatene, i et forsøk på å generere profitt.

Flere nevrale nettverk ble konstruert. Nettverkene brukte datakilder fra spillerrangeringer til lag-karakteristikker. De ble trent ved hjelp av simpel "backpropagation"-trening. Prediksjonene ble brukt sammen med odds fra syv internasjonale tippeselskap, i et forsøk på å generere profitt fra tipping. Forskjellige pengestyrings-strategier ble utprøvd, for å fremheve viktigheten av å velge riktige innsatsstørrelser.

Resultatene viser at enkle antagelser kan være nok for å nøyaktig predikere resultatene for fotballkamper. Resultatene viser også at nevrale nettverk kan brukes til å slå tippeselskapene i sitt eget spill: å tjene penger på fotballtipping.

Rapporten avsluttes med forfatterens tanker om hvordan å forbedre de presenterte resultatene ytterlige.

Preface

This master thesis has been carried out at the Department of Computer Science (IDI) at Norwegian University of Science and Technology (NTNU). The thesis was written in order to fulfill the graduation requirements of the Computer Science program at NTNU. I was engaged in researching and writing this study from August 2016 to June 2017.

I would like to thank all the people who have helped me carry out this thesis. Thanks to my supervisor, Helge Langseth, for his indispensable guidance and sincere opinions throughout this work. I would also like to thank two of my fellow students, Simen Selseng and Magnus Gundersen, for keeping me company while writing this thesis. Without them, I would not have made it through the long days at the study hall.

Last but not least, I would like to thank my fiancee, Marianne Gilje, for inspiring and supporting me throughout my studies and our eight years together.

I hope you enjoy your reading.

Simon Borøy-Johnsen
Trondheim, June 2017

Table of Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

AI Artificial Intelligence.

ANN Artificial Neural Network.

FNN Feedforward Neural Network.

HTML HyperText Markup Language.

IDI Department of Computer Science.

kNN k-Nearest Neighbor.

MAE Mean Absolute Error.

MSE Mean Squared Error.

NTNU Norwegian University of Science and Technology.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

ROI Return On Investment.

RPS Rank Probability Score.

SQL Structured Query Language.

SVM Support Vector Machine.

tanh Hyperbolic Tangent.

Chapter 1

Introduction

Association football (hereafter referred to as **football**) is one of the most popular sports in the world. The sport is played by millions of people all over the world, both for fun and professionally. Even more people enjoy watching the sport being played by professionals, in both national and international competitions (FIFA, 2014).

Several bookmakers all around the world allow placing bets on almost every aspect of a match: from the final outcome to which team will be awarded the next throw-in. Predicting the outcomes of football matches is a difficult task. The complexity of the sport, and the vast amount of variables affecting the outcome, increase the difficulty. Injuries, the physical and psychological shape of the players, referees and more; all contribute to the final outcome of a match.

Artificial Intelligence (AI) is as popular as ever, with tech giants such as Google and Microsoft pouring millions of dollars into acquiring AI companies (DeepMind (2017), Maluuba (2017)). This report will explore how AI, more specifically ANNs, can be used for predicting the outcomes of football matches. The prediction models produced will be tested in a real-life simulation, trying to beat the bookmakers in their own game: by profiting from betting.

1.1 Report goal and research questions

This section presents the goal of the report, as well as the research questions formed in order to achieve the goals.

The report has one main goal:

- Explore whether football match outcome predictions made by ANNs can be used for generating a profit from betting.

The two following research questions were formed in order to achieve the goal of the report.

- What a priori information concerning a football match can be used together with an ANN in order to predict the match outcome?

Modern technology allows for recording and storing all kinds of information about football matches, teams, players, etc. The number of goals scored, team and player characteristics, and player movement are just some examples of data available. How can the vast amount of data be utilized in order to predict the outcome of a given match?

- How can the predictions generated by ANNs be used for generating a profit in betting?

  When applying a prediction model in a betting system, there are several different ways of determining whether or not to place a bet, and how much money to place in each bet. The choice of money management strategy can be essential for the system's abilities to make a profit. It is therefore important to explore what money management strategies suit a given prediction model best.

1.2 Report outline

This section presents the outline of the rest of the report.

- **Chapter 2** presents the background research conducted before starting the work in this report. Firstly, a state of the art assessment concerning football match prediction and betting is presented. Then, relevant technologies are presented. Lastly, an overview of the data available at **www.whoscored.com** is presented.

- **Chapter 3** presents the theory needed in order to understand the contents of this report. The theory behind statistical classification and ANNs is presented.

- **Chapter 4** presents the architecture of the software system built for the experiments in this report. Firstly, an overview of the database used is presented. Then, the scraper used when scraping **www.whoscored.com** for data is presented. Lastly, the betting simulator constructed in order to test the prediction models in a betting setting is presented.

- **Chapter 5** presents the setup for the experiments conducted in this report. Firstly, a summary of the different prediction models is presented. Then, the betting simulation procedure is presented.

- **Chapter 6** presents the experiments conducted and results achieved in this report.

- **Chapter 7** presents a conclusion and discussion of the results achieved during the experiments, along with the author's thoughts on where to go from now. The author first reflects on how accepting different kinds of bets will affect the profitability of the networks. Then, thoughts on including different variables are presented. Lastly, a potential improvement to the data collection system is presented.

Chapter 2

Background

This chapter presents the background research conducted before starting the work in this report. Section 2.1, Section 2.2, and Section 2.4 were part of a specialization project done at NTNU (Borøy-Johnsen, 2016).

2.1 Models used for predicting outcomes of football matches

Already in 1982, Maher presented a model for predicting the outcomes of football matches. Earlier studies in the field of football match result predictions used the negative binomial distribution to model the number of goals a given team will score during a given match. The earlier studies had rejected the Poisson model for predicting football results, stating that ”chance does dominate the game” (Maher, 1982). This assumption was shown wrong by Hill (1974). Experts are able to accurately predict the final league standings before the league even start. According to Maher (1982), this indicates that chance may play a considerable part in the results of a match, but differences in teams strengths and skills dominate in the long run.

A team’s strength is usually divided into two separate strengths: attacking and defensive strengths. The attacking strength of a team represents the team’s ability to score goals. The defensive strength of a team represents the team’s ability to avoid conceding goals.

2.1.1 The Poisson distribution

Maher (1982) laid the foundation for several later prediction models when he presented his Poisson distribution model for predicting the number of goals scored by the the competing teams during a match. His model is based on the following assumption: each time a team has possession of the ball, there is an opportunity to attack, with the probability p of scoring a goal. There are n attacks during a match. If p is constant, and the attacks are independent, the number of goals can be approximated using the Poisson distribution. Maher (1982) assumed that if team i is playing at home against team j, and the final scoreline is (x_{ij}, y_{ij}), then the final scoreline can be modelled using Equation (2.1).

$$\begin{aligned} X_{ij} &\sim P(\alpha_i \beta_j) \\ Y_{ij} &\sim P(\gamma_i \delta_j) \end{aligned} \tag{2.1}$$

Maher (1982) described that the variables represent different aspects of the match. A summary is given in Table 2.1.1. The variables are based on the number of goals scored and conceded in the teams' earlier matches.

Variable	**What it represents**
α_i	The strength of team *i*'s attack when playing at home
β_j	The weakness of team *j*'s defence when playing away
γ_i	The weakness of team *i*'s defence when playing at home
δ_j	The strength of team *j*'s attack when playing away

Table 2.1.1: The variables in the model of Maher (1982).

Double Poisson distribution

In the initial model of Maher (1982), X_{ij} and Y_{ij} are assumed to be independent, "representing separate "games" at the two ends of the pitch". This is known as the double Poisson distribution.

Because of the independence, α and β can be estimated for x alone, whilst γ and δ can be estimated from y alone (Maher, 1982). The log likelihood function for the home team's score can therefore be expressed as in Equation (2.2).

$$log\ L(\alpha, \beta) = \sum_{i} \sum_{j \neq i} (-\alpha_i \beta_j + x_{ij}\ log(\alpha_i \beta_j) - log(x_{ij}!)) \tag{2.2}$$

Further, the maximum likelihood estimates, $\hat{\alpha}, \hat{\beta}$ can be shown to satisfy Equation (2.3) (Maher, 1982).

$$\begin{aligned} \hat{\alpha_i} &= \frac{\sum_{j \neq i} x_{ij}}{\sum_{j \neq i} \hat{\beta_j}} \\ \hat{\beta_j} &= \frac{\sum_{i \neq j} x_{ij}}{\sum_{i \neq j} \hat{\alpha_i}} \end{aligned} \tag{2.3}$$

To determine the maximum likelihood estimates, Maher (1982) used the Newton-Raphson method. The similar was done for $\hat{\gamma}$ and $\hat{\delta}$. The determined estimates show the effects of the home ground advantage, as each team's attacking strength is significantly reduced when playing away (Maher, 1982).

When evaluating his model, Maher (1982) used data from the four English Football League divisions for three consecutive years (1973-1975). Figure A.1.1 shows the calculated team strengths for all teams in the English Division 1 1971-1972. Maher (1982) raised the question of whether using four different parameters for each team is really necessary. Using maximum likelihood estimates, Maher (1982) showed that the relative strengths of teams' attack and defense are the same whether playing at home or away, and that only α and β are needed in order to make reasonable predictions. Maher (1982) then looked at the frequencies of goal scores scored in the English Division 1 over the three seasons. The results are shown in Figure A.1.2. The model seems to underestimate the number of occasions where one and two goals are scored, while overestimating the number of times no goals or ≥ 4 goals are scored (Maher, 1982).

Even though the model has shown promising results, the assumptions behind it are not realistic (Maher, 1982). M. Dixon and Robinson (1998) showed how p changes throughout the lifespan of a match. The scoring rate seems to increase throughout the game. M. Dixon and Robinson (1998) pondered that the teams fatigue near the end of the match, making defensive mistakes, which in turn lead to goals. They also showed how the scoring rates are dependent on the current score. A lead to the home team tends to decrease their scoring rate, whilst increasing the scoring rate of the opposing team. A lead for the away team tends to increase the scoring rate of both teams.

Bivariate Poisson distribution

To accommodate the over-simplification of his model, Maher (1982) used the available data to create an improved, bivariate Poisson distribution model. In the new model, the marginal distributions are still Poisson with the same means as before, but a correlation factor, ϱ, is added between the scores. The new model can be thought of as considering the difference in the number of goals scored, $Z_{ij} = X_{ij} - Y_{ij}$, resulting in a model with two dependent parts (Maher, 1982). One way to think of the bivariate Poisson distribution, according to Maher (1982), is that

$$X_{ij} = U_{ij} + W_{ij} \quad \text{and} \quad Y_{ij} = V_{ij} + W_{ij},$$

where U_{ij}, V_{ij} and W_{ij} are independent Poisson with means $(\mu_{ij} - \eta_{ij})$, $(\lambda_{ij} - \eta_{ij})$ and η_{ij}, respectively. η_{ij} being the co-variance between X_{ij} and Y_{ij}. Maher (1982) experimented with different values for ϱ, and a value of 0.2 yielded the best results. The bivariate version of his model improved the results considerably, compared to the initial model. Figure A.1.3 shows the different frequencies for Z_{ij} when applying the model to the English Division 1 1971-1972. One issue present in both the initial and improved models is the tendency to underestimate the number of drawn matches.

It is hard to say whether there is any correlation between the number of goals scored by each competing team. The question has been brought up in several studies. The assumption is, according to Maher (1982), too simple to model reality. D. Karlis and I. Ntzoufras (2003) argued that since two teams interact with each other during a match, the number of goals scored by each team are correlated. A change of style in play from one team will in turn change the probabilities of scoring a goal for both teams (D. Karlis and I. Ntzoufras, 2003). They supported their statements by analyzing data from the Champions League 2000-2001. McHale and Scarf (2007), on the other hand, found little to no evidence of any correlation between the number of goals scored by the opposing teams. They used data from the English Premier League 2003-2006.

D. Karlis and I. Ntzoufras (2003) built upon the model of Maher (1982). They also used the bivariate Poisson distribution in their model. D. Karlis and I. Ntzoufras (2003) modelled the number of goals scored in a match somewhat different from Maher (1982). Instead of adding the correlation factor separately from the distribution, D. Karlis and I. Ntzoufras (2003) added it to the distribution itself, resulting in a model of the form $(X_{ij}, Y_{ij}) \sim BP(\lambda_i, \lambda_j, \varrho)$, where

$$log(\lambda_i) = \mu + H + \alpha_i + \beta_j \quad \text{and} \quad log(\lambda_j) = \mu + \alpha_j + \beta_i. \tag{2.4}$$

λ_i and λ_j represent the expected number of goals scored for the home and away teams, respectively. ϱ is the correlation factor, μ is a constant parameter, and H is the home team effect parameter. α_k and β_i represent the attacking and defensive abilities of team k, like in the model of Maher (1982). μ represents the average number of goals scored per team when two teams of similar strengths play against each other.

Increasing the correlation between the number of goals scored improved the accuracy in prediction of draw games. D. Karlis and I. Ntzoufras (2003) further improved their model by inflating the probability of drawn games. They stated that inflating the probability corrected a possible overdispersion of results. Figure A.2.1 shows the estimates for different versions of their model. Model 8, the diagonal inflated bivariate Poisson distributed model, shows the most promising results.

The model of Koopman and Lit (2015) is also based on the bivariate Poisson distribution of Maher (1982). Their model is similar to that of D. Karlis and I. Ntzoufras (2003):

$$(X_{ij}, Y_{ij}) \sim BP(\lambda_i, \lambda_j, \varrho),$$

where λ_i and λ_j are the intensity coefficients for X and Y, and γ is the coefficient that measures the dependence between X_{ij} and Y_{ij}. λ_i and λ_j are allowed to vary over time. The intensities are then specified as

$$\begin{aligned} \lambda_{i,ijt} &= exp(H + \alpha_{it} + \beta_{jt}) \\ \lambda_{j,ijt} &= exp(\alpha_{jt} + \beta_{it}), \end{aligned}$$

where H is the home ground advantage coefficient and α_{kt} and β_{kt} the attacking and defensive strengths of team k in game week t. The attacking and defensive strengths are specified in Equation (2.5),

$$\begin{aligned} \alpha_{it} &= \mu_{\alpha,i} + \phi_{\alpha,i}\ \alpha_{i,t-1} + \eta_{\alpha,it} \\ \beta_{it} &= \mu_{\beta,i} + \phi_{\beta,i}\ \beta_{i,t-1} + \eta_{\beta,it} \end{aligned} \tag{2.5}$$

where $\mu_{\alpha,i}$ and $\mu_{\beta,i}$ are unknown constants, $\phi_{\alpha,i}$ and $\phi_{\beta,i}$ are auto-regressive coefficients, and $\eta_{\alpha,it}$ and $\eta_{\beta,it}$ are normally distributed independent error terms. α_{it} and β_{it} are determined using the maximum likelihood estimator.

Koopman and Lit (2015) applied their model in a betting setting. They used their predictions in combination with odds published at Football-Data (2016) for the English Premier League 2010-2012. During the evaluation, they used the Fixed bet strategy explained in Section 2.4.2, with a variable threshold τ. Figure A.3.1 shows the effect τ has on the profitability of their system. With $\tau > 0.12$, the system is able to systematically gain a profit.

Altered Poisson distribution models

M. J. Dixon and Coles (1997) used the initial model of Maher (1982) as basis for their model, with a small modification. The modified model included the home ground parameter, and can be seen in Equation (2.6).

$$\begin{aligned} X_{ij} &\sim Poisson(\alpha_i \beta_j H) \\ Y_{ij} &\sim Poisson(\alpha_j \beta_i), \end{aligned} \tag{2.6}$$

where α_k and β_k are the attacking and defensive strengths of team k, and H the home ground advantage parameter. To improve the accuracy for low-scoring matches, M. J. Dixon and Coles (1997) modified their model further, adding a dependence parameter. The new model can be seen in Equation (2.7).

$$Pr(X_{ij} = x, Y_{ij} = y) = \tau_{\lambda,\mu}(x,y) \frac{\lambda^x exp(-\lambda)}{x!} \frac{\mu^y exp(-\mu)}{y!}, \tag{2.7}$$

where

$$\begin{aligned} \lambda &= \alpha_i \beta_j H \\ \mu &= \alpha_j \beta_i \end{aligned} \tag{2.8}$$

and

$$\tau_{\lambda,\mu}(x,y) = \begin{cases} 1 - \lambda\mu\varrho & \text{if } x = y = 0 \\ 1 + \lambda\varrho & \text{if } x = 0, y = 1 \\ 1 + \mu\varrho & \text{if } x = 1, y = 0 \; . \\ 1 - \varrho & \text{if } x = y = 1 \\ 1, & \text{otherwise} \end{cases} \tag{2.9}$$

ϱ is used as an dependence parameter. $\varrho = 0$ corresponds to independence. For $x \leq 1$ and $y \leq 1$, the independence distribution is altered.

Whilst the model of Maher (1982) incorporates static team strengths, M. J. Dixon and Coles (1997) stated that recent results are more important than old ones to describe a team's current form. To incorporate this into their model, M. J. Dixon and Coles (1997) scaled the contributions of older data, making recent data more significant. They also modified their model to enable inclusion of incomplete data sets and data from different leagues.

Rue and Salvesen (2000) also used the initial model of Maher as base for their work. They represent the attacking and defensive strengths of team i as random variables α_i and β_i respectively. Higher values imply greater strengths. They also represent $\mu_{\alpha,i}$ and $\sigma^2_{\alpha,i}$ as the prior mean and variance of α_i, and similar for defence. $e_i = (\alpha, \beta)_i$ represents the properties of team i. Rue and Salvesen (2000) also added a psychological effect, modelling the underestimation when a superior team meets a team that is of supposed inferior quality. The psychological effect is given as

$$\Delta_{ij} = (\alpha_i + \beta_i - \alpha_j - \beta_j)/2,$$

and replaces the home ground advantage. However, Rue and Salvesen (2000) propose the home ground advantage to be part of an extended version of their model. Another change from the model of Maher (1982) is the addition of a time factor, used to weigh the recent results above more distant results. In addition to the added variables, Rue and Salvesen (2000) did some extra modifications to the model of Maher (1982). They assumed that some of the information from the final scoreline comes from the league itself, and not the actual result. In a league where each team on average scores 1.23 goals per match, scoring 1 goal in a match is actually below the expected number of goals. To model this, Rue and Salvesen (2000) used a variable, called ϵ, which determines how much the league average should contribute to the predicted number of goals. This can be interpreted in the sense that only $(1 - \epsilon) * 100\%$ of the "information" in a match result is informative on e_i and e_j (Rue and Salvesen, 2000). Through experiments, they found that $\epsilon = 0.2$ yielded the best predictive results. They also observed that as one team scored a lot of goals, the probabilities diverted from the Poisson distribution. This was "solved" by clipping the number of goals scored at 5. Any number of goals scored above 5 did not count. That is, 8-2 is treated as 5-2, and 6-5 as 5-5. Rue and Salvesen (2000) are not sure whether this is the best approach to solving the problem with diverting probabilities. Another mentioned solution is to reduce the scoreline until one get a more common one, i.e. from 7-4 to 5-2. By doing this, the information of the final result is kept, whilst removing the extra goals that do not provide important information (Rue and Salvesen, 2000). The system of Rue and Salvesen (2000) is modelled as a Bayesian network. They added the use of Bayesian methods to update the estimates after each new match was played, and used Markov chain Monte Carlo techniques to draw inference from the network.

Rue and Salvesen (2000) applied their model in a betting setting, using their strategy presented in Section 2.4.2. The cumulative profits for the English Premier League and Division 1 1997-1998 are shown in Figure A.4.1. As can be seen from the figure, the system was able to gain significant profits in both leagues, with a maximum profit of 250% during the simulation. According to Rue and Salvesen (2000), the big spike in profits in the end of January was due to a single match, Manchester United versus Leicester City. The odds for an away-win was given at 13.8, while the model predicted a probability of 0.184 for the same outcome. The match ended $0 - 1$, resulting in a significant pay-off. In the end, the final profits were 39.6% and 54.0% respectively. Their system was able to win 42 of the 112 bets placed.

Poisson difference distribution models

In a later paper, Dimitris Karlis and Ioannis Ntzoufras (2009) changed from using the bivariate Poisson distribution. They instead used the Poisson difference distribution. By doing this, they removed the effect of the correlation between the performance of the competing teams, and instead modelled the goal difference directly (Dimitris Karlis and Ioannis Ntzoufras, 2009). To model the goal difference, Dimitris Karlis and Ioannis Ntzoufras (2009) calculated

$$Z_{ij} = X_{ij} - Y_{ij} \sim PD(\lambda_i, \lambda_j),$$

where X_{ij} and Y_{ij} are the Poisson distributions for the number of goals scored by the home team and away team, respectively. Z_{ij} is then the distribution of the goal difference, and is called the Skellam's distribution (or Poisson difference distribution (PDD)) (Dimitris Karlis and Ioannis Ntzoufras, 2009). λ_i and λ_j are the model parameters, modelling the expected number of goals scored by each team, just like in D. Karlis and I. Ntzoufras (2003). The parameters are the same as defined in Equation (2.4). Dimitris Karlis and Ioannis Ntzoufras (2009) imposed two constraints on the attacking and defensive strengths of the teams, as given in Equation (2.10),

$$\sum_{k=1}^{K} \alpha_k = 0 \quad \text{and} \quad \sum_{k=1}^{K} \beta_k = 0, \tag{2.10}$$

where K is the number of competing teams. The attacking and defensive parameters can the be interpreted as the deviations from a team of moderate performance. Further, H can then be interpreted as the expected goal difference in a match between two teams of the same attacking and defensive skills (Dimitris Karlis and Ioannis Ntzoufras, 2009). Dimitris Karlis and Ioannis Ntzoufras (2009) authors show that the marginal distributions of X and Y are only Poisson distributed in special cases, and are in general defined as the convolution of a Poisson random variable with another discrete random variable, thus removing a large portion of the distributional assumptions concerning the number of goals scored by each team. This is one of the reasons they proposed the use of the Poisson difference distribution (Dimitris Karlis and Ioannis Ntzoufras, 2009). Dimitris Karlis and Ioannis Ntzoufras (2009) then proceed to use Bayesian methods in order to incorporate any available information about each game via the prior distributions. Examples of mentioned relevant information are injuries, weather conditions and the fitness of the teams. Where no information is available, Dimitris Karlis and Ioannis Ntzoufras (2009) proposed to use normal prior distributions for the parameters of the model, with mean equal to zero and a large variance (for example 10^4). This is done to express prior ignorance (Dimitris Karlis and Ioannis Ntzoufras, 2009). The posterior predictive distributions are calculated using the Markov chain Monte Carlo algorithm.

According to Shahtahmassebi and Moyeed (2016), there are numerous limitations to the before-mentioned models utilizing the Poisson distribution. They cite Dimitris Karlis and Ioannis Ntzoufras (2009), supporting the use of goal difference instead of the number of goals scored. However, they address the drawback of overestimating the number of draws in the model of Dimitris Karlis and Ioannis Ntzoufras (2009). Another issue with the PDD is an issue with the double round-robin structure of most leagues. The home ground advantage results in distributions with one or both tails being too short or too long for the distribution (Shahtahmassebi and Moyeed, 2016). To overcome the limitations of the other Poisson-based models, Shahtahmassebi and Moyeed (2016) used the generalized Poisson difference distribution (GPDD). The GPDD function is given in Equation (2.11).

$$f_{GPDD}(Z = X - Y = z|\lambda_1, \lambda_2, \theta_1, \theta_2) = e^{-\lambda_1 - \lambda_2 - \theta_1 z} \sum_{y=0}^{\infty} (\lambda_1, \theta_1)_{z+y} \, (\lambda_2, \theta_2)_y \, e^{-(\theta_1 + \theta_2)y} \tag{2.11}$$

for any $z \in \mathbb{Z}$, where

$$(\lambda, \theta)_x = \frac{\lambda(\lambda + x\theta)^{x-1}}{x!}.$$

λ_1 and θ_1 refer to the positive half of the distribution, while λ_2 and θ_2 refer to the negative half. The GPDD model can, just like the PDD model, be used for predicting the outcome of a match, but not the scoreline itself. The GPDD model, however, introduces more flexibility in its tails than the PDD model (Shahtahmassebi and Moyeed, 2016). To model the goal difference of a match, Shahtahmassebi and Moyeed (2016) used the GPDD model as follows:

$$E(Z_i) = \mu_i = H + a_{h_i} - a_{v_i}$$
$$Var(Z_i) = \sigma_i^2 = \gamma_1 + |a_{h_i} - a_{v_i}|,$$

where a_{h_i} and a_{v_i} are the abilities of the home and visiting team in match i, H is the home ground effect parameters (equal for all teams), and γ_1 is a positive constant for the variance. The variance is defined to increase with the difference in team abilities. Shahtahmassebi and Moyeed (2016) imposed the constraint that all abilities sum to zero $(\sum_{k=1}^{K} a_k = 0)$. Furthermore, the values of θ_1 and θ_2 are assumed constant with respect to the team abilities. The values for λ_1 and λ_2 can be obtained using Equation (2.12).

$$\begin{aligned} \lambda_{1,i} &= \frac{[(1-\theta_2)^2\, \sigma_i^2 + \mu_i](1-\theta_1)^3}{(1-\theta_1)^2 + (1-\theta_2)^2} \\ \lambda_{2,i} &= \frac{[(1-\theta_1)^2\, \sigma_i^2 + \mu_i](1-\theta_2)^3}{(1-\theta_1)^2 + (1-\theta_2)^2} \end{aligned} \tag{2.12}$$

Shahtahmassebi and Moyeed (2016) fitted the model in a Bayesian framework in order to incorporate any available information about each game via the prior distribution. The Bayesian approach allows for predicting match outcomes via the posterior predictive distribution, as well as for producing quantitative measures relating each team's performance. To generate samples for the posterior distribution, Shahtahmassebi and Moyeed (2016) used the Markov chain Monte Carlo random walk Metropolis-Hastings algorithm. The model presented use the previous season as a baseline for the following season's results. Shahtahmassebi and Moyeed (2016) state that teams generally at least intend to keep their position in the table from one season to the next. It is thus, according to the authors, realistic to use information from the previous season as prior information. For teams that are promoted and playing in the league for the first time, a non-informative normal prior distribution is assigned, like the one of Dimitris Karlis and Ioannis Ntzoufras (2009). The authors explain that this is done due to the nature of a football league: each team's abilities are measured relative to other teams in the same league. Shahtahmassebi and Moyeed (2016) mention some issues with their final model. Firstly, using the previous season as baseline for the prior distributions may not be the optimal solution, as this does not allow for time-varying team abilities (Shahtahmassebi and Moyeed, 2016). They suggest an extension of the model would be to consider a dynamic model that supports varying team abilities, as well as a varying home ground advantage effect.

2.1.2 Elo rating

The ELO rating system is a rating system developed by for calculating the relative skills of players or teams in competitor-vs-competitor games, initially developed by Appard Elo. The system was initially developed for assessing the strengths of chess players, but have since been widely adopted for use in other sports (Ross, 2007). The central assumption of the ELO rating system is that the performance of a competitor (either a person or a team) is a normally distributed random variable. Elo assumed the true skill level of a competitor to be the mean of this variable, and that the true skill level changes slowly over time. The skill level of a competitor serves as the basis for the competitor's ELO rating (Ross, 2007).

Hvattum and Arntzen (2010) used the ELO rating system in their model. Based on the results of previous matches, each team is assigned an ELO rating. This rating serves as a measure of the team's current strength. It should be noted that the computations of the ELO ratings need some initial ratings to be provided for each team, and that the ratings therefore can not be expected to be reliable before a sufficient number of matches have been taken into account (Hvattum and Arntzen, 2010).

Let ℓ_i^H and ℓ_i^A be the ratings of the home and away teams before a match at time i. A score system is defined, where points are rewarded to the teams after each match. The points awarded for a single match sum to 1. The expected score by the home and away teams are given by γ^H and γ^A respectively, where

$$\gamma^H = \frac{1}{1 + c^{(\ell_i^A - \ell_i^H)/d}} \quad \text{and} \quad \gamma^A = 1 - \gamma^H = \frac{1}{1 + c^{(\ell_i^H - \ell_i^A)/d}}.$$

c and d can be interpreted as setting the scaling of the rating (Hvattum and Arntzen, 2010). To calculate the ratings, the expected scores are compared to the observed scores, α^H and α^A respectively, given by

$$\alpha^H = \begin{cases} 1.0 & \text{if the home team wins} \\ 0.5 & \text{if the match is drawn} \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad \alpha^A = 1 - \alpha^H.$$

The rating of the home team is updated after each match according to Equation (2.13). The same is done for the away team rating.

$$\ell_i^H = \ell_{i-1}^H + k(\alpha^H - \gamma^H) \tag{2.13}$$

Hvattum and Arntzen (2010) present two different versions of the ELO rating system. The first, referred to as the *basic* ELO rating (ELO_b), set k to be a constant parameter. The second, referred to as the *goalbased* ELO rating (ELO_g), replace k with the expression $k = k_0(1+\delta)^\lambda$, where k_0 and λ are constant parameters, and δ the absolute goal difference. The *goal based* rating takes the goal difference in a match into account, rewarding a 3-0 win more strongly than a 2-1 win (Hvattum and Arntzen, 2010). According to Hvattum and Arntzen (2010), the values of c and d are not that important, as they only serve to set a scale of the ratings. Alternative values for c and d produce identical rating systems, but one have to set suitable values for k, k_0 and λ (Hvattum and Arntzen, 2010). In their experiments, Hvattum and Arntzen (2010) set $c = 10$ and $d = 400$, and calibrated the values of k, k_0 and λ by minimizing the quadratic loss between the predicted values and the observed outcomes.

In order to use the ELO ratings for match prediction, Hvattum and Arntzen (2010) make use of an ordered logit regression model (R. Williams et al., 2006). An initial set of matches is used to compute initial ratings for all the teams in the league. A second set of matches is used to estimate the parameters of the model. The rating difference,

$$x = \ell_i^H - \ell_i^A,$$

prior to the match serves as the covariate in the regression model. This system allows for updating both the ratings and the regression parameters, ensuring the most recent data is always utilized (Hvattum and Arntzen, 2010).

The regression model serves as the basis for calculating the match predictions. The predictions are obtained by assigning the corresponding probability for each outcome of the match, resulting in a probability distribution for the three outcomes. Figure 2.1.1 shows the outcome probabilities as a function of the rating difference, given by the regression model at the end of the English Premier League 2006-2007. The home win and away win probabilities are equal when the rating difference is about -80. This shows how the model captures the home ground advantage (Hvattum and Arntzen, 2010).

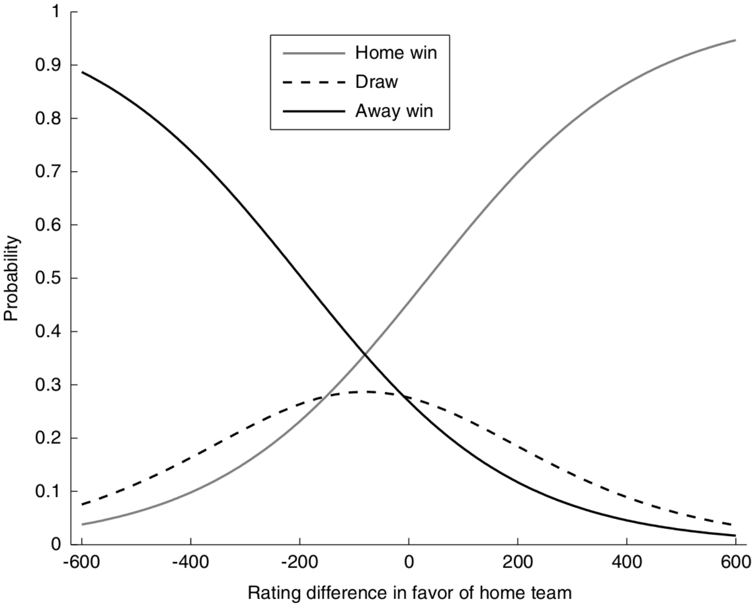


Figure 2.1.1: Outcome probabilities as a function of rating difference (in favor of home team), given by the regression model at the end of the English Premier League season 2006/2007. Taken from Hvattum and Arntzen (2010).

Hvattum and Arntzen (2010) evaluated their models in a betting setting. During the evaluation, they compared their models to other prediction models. The other prediction models include naive methods like assuming uniform probability of all outcomes (UNI), and basing the probabilities of observed past frequencies (FRQ). In addition, two versions of the model presented in John Goddard (2005) were used (GOD_b, GOD_g). To compare the different models, Hvattum and Arntzen (2010) used their predictions in combination with odds collected from various bookmakers. Match data from the English Premier League seasons 1993-2008 were used. The first two seasons were used for initial calculations of the ELO ratings. The five next seasons were used for estimating the parameters in the different prediction models. Finally, the eight remaining seasons were used for actual testing. Figure A.5.1 shows the total return for the different models. As can be seen, none of the presented models were able to gain a profit.

2.1.3 Bradley–Terry modelling

Cattelan, Varin, and Firth (2013) built a prediction system using a dynamic Bradley-Terry model. The Bradley-Terry model is a probability model for predicting the outcome of a comparison. Given two individuals, a and b, the model estimates the probability $P(a > b)$ using

$$P(a > b) = \frac{p_a}{p_a + p_b}, \tag{2.14}$$

where p_a and p_b are the scores assigned to a and b respectively. Here, $a > b$ can be read as "a is preferred to b".

Cattelan, Varin, and Firth (2013) use descriptors of the competing team's strengths as the basis for the scores. Two separate team strengths are calculated, one for matches played at home, one for matches played away. The strengths are calculated using earlier strengths and the number of points achieved in recent matches. In their model, Cattelan, Varin, and Firth (2013) specify an evolution of a team's strengths depending only on past matches played of the same type. For home matches, the strength is estimated using Equation (2.15).

$$\alpha_{h_i}(t_i) = \lambda_1 \mu_{h_i}(t_i) + (1 - \lambda_1)\alpha_{h_i}(t_{i-1}), \tag{2.15}$$

where $\alpha_{h_i}(t_i)$ is the home strength of team i at time t_i and t_{i-1} the time of the previous match played at home by team h_i. The term $\mu_{h_i}(t_i)$ denotes the recent home strength of team i, based on the number of points earned in the last home match. $\mu_{h_i}(t_i)$ is defined as

$$\mu_{h_i}(t_i) = \beta_1 r_{h_i}(t_{i-1}),$$

where β is a home-specific parameter, and $r_{h_i}(t_{i-1})$ the number of points earned in the last home match (3 for victory, 1 for draw, 0 for loss). $\lambda_1 \in [0, 1]$ is used for determining how last result is weighted when estimating the team's home strength. The away strengths are estimated similarly, using away-specific parameters (λ_2 and β_2) instead of λ_1 and β_1.

To estimate the initial strengths, Cattelan, Varin, and Firth (2013) assume that all teams start with equal home strength, $\beta_1 \bar{r}_h$, where $\bar{r}_h$ is the average number of points gained at home in the previous season. $\alpha_{h_i}(t_i)$ is then estimated using iterated back-substitution, thus incorporating the whole past of home matches (Cattelan, Varin, and Firth, 2013). The same goes for the away strength. The values of λ_1 and λ_2 are estimated using maximum profile likelihood estimation (Cattelan, Varin, and Firth, 2013).

To estimate the probabilities of each outcome, Cattelan, Varin, and Firth (2013) use Equation (2.16).

$$P(Y_i \leq y_i | Y_{i-1} = y_{i-1}, ..., Y_1 = y_1) = \frac{exp(\delta_{y_i} + \alpha_{h_i}(t_i) - \alpha_{v_i}(t_i))}{1 + exp(\delta_{y_i} + \alpha_{h_i}(t_i) - \alpha_{v_i}(t_i))}, \tag{2.16}$$

where $y_1 \in \{0, 1, 2\}$ denotes the outcome of the match (2 for home team victory, 1 for draw and 0 for away team victory). δ_{y_i} are cut-point parameters, where $\delta_0 < \delta_1 < \delta_2$. The cut-point parameters are needed for the Bradley-Terry model to support three outcomes. By setting $\delta_0 = -\delta$ and $\delta_1 = \delta$, with $\delta \geq 0$, one can ensure that two teams of the same strength playing at a neutral ground have the same probabilities of winning the match (Cattelan, Varin, and Firth, 2013).

When applying their model to the Italian Serie A 2008-2009, Cattelan, Varin, and Firth (2013) concluded that their model seems to capture the relevant aspects of the evolution of team strengths. However, they find it reasonable to assume that using more information about the previous matches may result in improved predictions and more accurate forecasts.

2.1.4 pi-football

The pi-football (probabilistic intelligence football) model (A. C. Constantinou, N. E. Fenton, and Neil, 2012) is a Bayesian network model for predicting the outcomes of football matches, in the form of probabilities for each possible outcome (home win, draw, away win). The model is built up from mixing historical data with subjective expert knowledge. When building the model, the authors collected historical data from more than 6000 matches in the English Premier league from 1993 to 2010. The system was then used to predict the outcomes of all matches in the English Premier League 2010-2012, all of which are available online. The historical data is used to generate the model priors. A special feature of the pi-football model is the use of ”anonymous” priors. That is, priors are predetermined by team-strength, not by distinct team names. Team strength is supplied as a ranked number representing the strength of a team for a particular season. The strength is based on a team’s table position (using the number of accumulated points), separating the space of the league table into 14 levels. For example, the Manchester City match at home against Aston Villa the season of 2006-2007 is classified as a ranked 10 team at home against a ranked 8 team (with Manchester City totalling 42 points and Aston Villa totalling 50 points (A. C. Constantinou, N. E. Fenton, and Neil, 2012)). The anonymous approach has several advantages (A. C. Constantinou, N. E. Fenton, and Neil, 2012):

- It allows for making maximum use of limited data, as when predicting matches including newly-promoted teams.
- There is no need to ignore or weigh historical observations, as the system use the current strength of teams, and not their historical strengths.
- Historical observations do not need to be updated frequently, as there is a lot of historical data available.
- Data from one league can easily be adapted to work for another league, as the specific teams are not part of the model.

The model make use of four different factors to determine the abilities of a team:

1. **Team strength**
2. **Team form**
3. **Team psychology**
4. **Team fatigue**

Factor 1 is the only objective factor in the model. It makes use of recent historical data to estimate of a team's current strength. The other three factors are used to revise the predictions made from factor 1. All factors are modelled as their own Bayesian network. The outcome of the three subjective factors are summarized in a single parameter, with a value from 0 to 1. A value of 0.5 signals no advantage to either team. A value of less than 0.5 signals an advantage to the home team. A value greater than 0.5 signals an advantage to the away team.

The network corresponding to factor 1 has three main components:

- **Previous information:** Five parameters, each holding the number of pints accumulated the last five seasons. There is an increasing degree of uncertainty for the older seasons.
- **Current information:** A single parameter, holding a rough estimate of the team's current strength. Measured according to the number of points accumulated the current season and the expected number of points from the remaining matches. There is an increasing degree of uncertainty for the number of remaining matches.
- **Subjective information (optional):** Represented by a single parameter, holding an expert's subjective believed about the strength of a team. This is used to capture important events not captured by the historical data, such as the vast money usage by Manchester City the seasons from 2009 to 2012 (they spent £160m, £77m, £75m before the start of each seasons, respectively), improving their squad significantly.

The form of a team indicates a team's recent performance against its expectations. This is measured by comparing the team's expected performance against its observed performance during the five last game-weeks. The network of factor 2 determines whether one of the teams has better current form than the other, and has two main components:

- **Current form:** Measured by a scale from 0 to 1. Scaled similarly to the subjective factors, indicating whether the team has over- or under performed according to its strength. Incorporates the home ground advantage; weights home form and general form ($\frac{2}{3}$ and $\frac{1}{3}$, respectively).
- **Availability of players resulted in current form:** The form is revised according to subjective factors including the availability of certain players, and the effect of returning first team players.

The network of factor 3 determines the difference in the psychological impact between the two teams, and has three main components:

- **Head-to-head biases:** Models the psychological effect of head-to-head biases, such as local derbies.
- **Managerial impact:** Models any impact managerial issues might have on the team, such as recent change of manager.
- **Team spirit and motivation:** Models the current team spirit and motivation of the team.

Fatigue is determined by the toughness of the previous match, the number of days since that match, the number of first team players rested, and the participation of first team players in national team matches. The network of factor 4 has three main components:

- **Restness:** The number of days since last match, along with information about resting first team players during that match. Gives an indication on how rested the team is.

- **Toughness of previous match:** The toughness of the previous match is also important in modelling a team's fatigue.

- **National team participation:** Can increase the fatigue by up to 50%, depending on the level of participation of first team players in national team matches.

By combining the objective historical data with the subjective factors, the forecast prediction accuracy increased significantly, according to A. C. Constantinou, N. E. Fenton, and Neil (2012). They emphasise the importance of the quality of the expert's knowledge, claiming "...a perfect BN model would still fail to beat the bookmakers at their own game if the subjective expert inputs are inaccurate". With the weekly pressure to post their predictions online, the authors often had to get their subjective inputs from a team member, who "is certainly not an expert on the English premier League", resulting in inconsistent prediction accuracy (A. C. Constantinou, N. E. Fenton, and Neil, 2012). The authors also emphasise the importance of Bayesian networks, in which the subjective information easily can be represented and displayed.

A. C. Constantinou, N. E. Fenton, and Neil (2012) applied their model in a betting setting. They used their predictions in combination with odds published at Football-Data (2016) for the English Premier League 2010-2011. In the evaluation, they used the Fixed bet strategy explained in Section 2.4.2, with a fixed threshold $\tau = 5\%$.

The works of A. C. Constantinou and N. E. Fenton (2013) show that the odds of a single bookmaker are not representative of the overall betting market. Therefore A. C. Constantinou, N. E. Fenton, and Neil (2012) considered three different sets of odds when evaluating their system: the maximum odds available for each match outcome, the mean odds available for each outcome, and the odds specified by the most used bookmaker in the UK (representing 25% of the total UK and Irish betting market (A. C. Constantinou, N. E. Fenton, and Neil, 2012)), William Hill.

Figure A.6.1 shows the cumulative profits gained by the model using the different odds sets. As can be seen, the model was able to gain a profit for each of the different odds sets. Approximately 35% of all placed bets were won. Figure A.6.2 show more detailed statistics for the three simulations. For $\tau = 5\%$, the model was able to generate a total profit of 14.19% for the max odds set by the end of the season. By adjusting the value of τ to 11%, A. C. Constantinou, N. E. Fenton, and Neil (2012) were able to increase the total profits to 35.63%.

A. C. Constantinou, N. E. Fenton, and Neil (2012) suggest some extensions to the pi-football model. First, they mention a planned extension, exploring the effectiveness of the individual components used in their model. They hope this will help them understand how the specific components help in matching the bookmakers' odds. Another extension can explore whether revising the team strength itself (given subjective information), rather than the probability distribution, will improve the model's accuracy. Lastly, A. C. Constantinou, N. E. Fenton, and Neil (2012) discuss exploring the impact of the time-dependent uncertainty when weighing the recent information.

2.2 Variables to consider when predicting match outcomes

When predicting the outcome of football matches, there are several variables that can impact the prediction accuracy of a model.

2.2.1 Home ground advantage

Multiple studies have covered the home ground advantage, a phenomenon where several aspects of a match favor the team playing at its home ground.

Courneya and Carron (1992) did a state-of-the-art review concerning the home ground advantage, which was reviewed a decade later by Carron, T. M. Loughead, and Bray (2005). Courneya and Carron (1992) presented a framework for game location research. Carron, T. M. Loughead, and Bray (2005) presented a revised version of the framework, which is shown in Figure 2.2.1. The framework incorporates five major components, where the factors influence each other from left to right. The components are as follows:

- **Game location**: Simply represents the game site; home versus away. Courneya and Carron (1992) suggested that the framework would not work for matches played at neutral grounds, even though one of the teams might be designated as the ”home team”.

- **Game location factors**: Represent four major factors that differently impact the teams (players and coaches) playing at their own ground versus playing away:

    - The *crowd factor* is an acknowledgment that the home team has more support from their spectators than the away team has.

      Studies have demonstrated how the crowd behavior affects the competing teams, showing that the home team seem to commit more violations when the crowd is showing antisocial behavior (like swearing and throwing objects onto the pitch) (Carron, T. M. Loughead, and Bray, 2005).

      Other studies have shown the effect of the crowd size. The works of Nevill, Newell, and Gale (1996) indicate that absolute crowd size is positively related to the home ground advantage in English and Scottish football. The home teams had an increased home ground advantage in matches where the crowd size was large, while the home ground advantage was nearly absent in two leagues (GM Vauxhall League and Scottish Second Division) where crowd sizes were small (Nevill, Newell, and Gale, 1996).

    - The *learning factor* is an acknowledgment that the players playing at home are more familiar with the grounds, and that the club has the ability to temporarily capitalize on their strengths (for example by softening the pitch through extensive watering).

According to Carron, T. M. Loughead, and Bray (2005), studies indicate that teams playing on smaller or larger playing surfaces may have a higher home team advantage than average. Studies also show that teams playing on pitches of artificial grass have a significantly higher home team advantage than average.

The effect of home ground familiarity on the home ground advantage has also been studied. Todd M Loughead et al. (2003) collected match information from the English and Scottish Professional Football Associations 1988-2000. They classified matches into three blocks: a) 10 games immediately before relocating to a new venue b) 10 games immediately after relocating to a new venue c) 10 games when the teams had become familiar to the new venue. Before relocating, the teams won 55.2% of their home games. Immediately after relocating, the percentage was reduced to 53.9%. After becoming familiar with the new venue, the percentage was virtually unchanged at 53.1% (Todd M Loughead et al., 2003). Later on, Todd M Loughead et al. (2003) conducted an post-hoc analysis of the results to examine the relationship between team quality and venue familiarity. The findings showed how teams with high home ground advantage suffered significant reductions immediately after moving (i.e. 70.6% to 59.2%), while teams with low advantage had the opposite effect (i.e. 34.1% to 46.8%).

- The *travel factor* is an acknowledgment that the away team has to undergo the inconvenience of travelling.

  Studies involving the relationship between home ground advantage and travel distance for the away team have shown that travel distance contribute to the home ground advantage, but that its impact is relatively small (Carron, T. M. Loughead, and Bray, 2005).

- The *rule factor* is an acknowledgment that in some sports the rules may favor the home team. This factor does not affect football (Carron, T. M. Loughead, and Bray, 2005).

- **Critical psychological states and Critical behavioural states**: Represent how the psychological and behavioral states of the teams are influenced by game location factors. Focus on the impacts of playing at home ground versus playing away.

  According to Carron, T. M. Loughead, and Bray (2005), there are only a few studies concerning the effect of football players' and coaches' psychological states when playing away vs playing at home. One study showed that for male university rugby players, playing at home reduced the level of anxiety, tension, depression, anger, fatigue, etc. The effect of behavioural state has also received little attention (Carron, T. M. Loughead, and Bray, 2005).

  In the initial framework, match officials were also part of the psychological and behavioural states. But seeing as officials do not have home or visitor status, Carron, T. M. Loughead, and Bray (2005) removed match officials from the revised framework, and instead looked at them separately.

- **Performance outcomes**: Represent how the performance of the teams is influenced by game location, game location factors and the psychological and behavioral states of the teams.

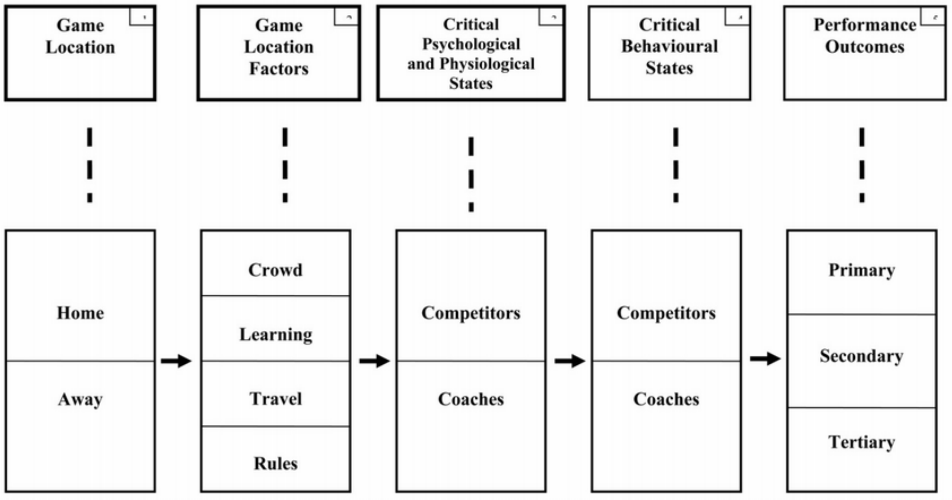


Figure 2.2.1: Framework for game location research. Taken from Carron, T. M. Loughead, and Bray (2005).

The framework has been proven useful for providing guidelines on what factors to examine when researching the home ground advantage.

To quantify the home ground advantage, J. Goddard (2006) collected match results for 35 consecutive seasons in English league football. He looked at the different outcomes of the matches, and how victories are affected by playing at home versus playing away. He also recorded the number of goals scored by the teams over the same period. A summary of his findings is shown in Table 2.2.1. The findings clearly show that, even though it has declined the later years, there is still a significant difference between the average performance of the home team and the away team. Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) studied teams' style of play. They found that the home team tend to play more in the attacking third of the pitch, which may help explain the difference in number of goals scored between the home and away teams.

There are several other factors influencing the home ground advantage. Pollard (2008) also did a state-of-the-art review on the home ground advantage. He mention how increased travelling distance might increase the home ground advantage, but that the research has shown inconclusive results. The advantage is, however, reduced in local derbies, such as when Arsenal play against Tottenham, two teams with home grounds only 6.6 km. apart. Pollard (2008) also mention a referee bias. There is evidence that the referee decisions favor the home team. One example is the number of bookings, where the home team is given less bookings that the away team. The bias has been demonstrated in a laboratory setting, where the committed fouls are considered and compared (Pollard, 2008). The findings are supported by Nevill, Balmer, and A. M. Williams (2002), who analyzed 40 referees assessments of an English Premier League match between Liverpool and Leicester City. The referees were exposed to either an audible crowd noise group or no sound at all. The officials in the audible noise group called significantly fewer fouls against the home team than the referees in the silent group. Lastly, Pollard (2008) present research supporting the factors presented in the framework of Carron, T. M. Loughead, and Bray (2005), as well as the special tactics discovered by Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014)

	Period			
Match results (%)	**1970–1980**	**1980–1990**	**1990–2000**	**2000–2005**
Home win	50.3	48.8	46.4	45.4
Draw	28.6	26.7	27.6	27.5
Away win	21.0	24.5	25.9	27.2
Goals per match (avg.)				
Home team	1.58	1.60	1.51	1.50
Away team	0.97	1.06	1.08	1.10

Table 2.2.1: Trends in home ground advantage in the period 1970-2005. Taken from J. Goddard (2006).

2.2.2 Attacking and defensive strengths

Several studies mention the importance of modelling a team's playing abilities, usually in the form of attacking and defensive strengths. However, the way team strengths are modelled have changed over the years.

In the early studies, like that of Maher (1982), only the number of goals scored and conceded were used for calculating the team strengths. Usually, a team's attacking strength represents their ability to score goals, when their defensive strength represents their ability to avoid conceding goals. Maher (1982) modelled four different strengths for each team: attacking strength when playing at home, defensive strength when playing at home, attacking strength when playing away, and defensive strength when playing away. The distinction between home and away strengths is found in almost every reviewed model, either in the form of separate strengths, or by adding some sort of home ground advantage parameter to the model itself. While the strengths in the model of Maher (1982) do not change over the course of a season, later goal-based studies, like M. J. Dixon and Coles (1997) and Rue and Salvesen (2000), agree that the strengths of a team vary over time, and that recent results are more important than older results when modelling the current strengths. Some later models, like that of Cattelan, Varin, and Firth (2013), focus match outcomes, rather than the number of goals scored. They model a team's strengths at time-step i by the strength at time-step $i - 1$ and the number of points achieved in the most recent match.

When adjusting the teams' strengths, the models above do not differentiate ”expected” results from those that do not fit the models. For example how lower ranked teams are ”expected” to lose against higher ranked teams. The results of a match where the 1st ranked team loses $0 - 2$ at home against the 17th ranked team are treated equally to the results where the 17$^{\text{th}}$ ranked team loses $0 - 2$ at home against the 1$^{\text{st}}$ ranked team. The probabilities of the two outcomes are quite different, but this difference is not used when adjusting the strengths of the two teams. In the model of Hvattum and Arntzen (2010), the current rank of the two competing teams are taken into account. The difference in rank forms the basis for the ELO rating system.

The system of A. C. Constantinou, N. E. Fenton, and Neil (2012) takes it a step further. In their system, A. C. Constantinou, N. E. Fenton, and Neil (2012) do not care about what teams are playing. They consider the teams' current strengths only. To model a team's current strength, A. C. Constantinou, N. E. Fenton, and Neil (2012) make use of the total number of points accumulated the last three seasons, and the number of points accumulated so far the current season, supplemented by the expected number of points for the remaining matches. Team strength is further adjusted using subjective information not captured by previous results (see Section 2.1.4). Another quite unique feature in the model of A. C. Constantinou, N. E. Fenton, and Neil (2012) is the use of a team's current form. Other models also make use of the recent results of a team, but the model of A. C. Constantinou, N. E. Fenton, and Neil (2012) measure the recent results of a team according to the expected results, indicating whether a team currently has a good run. In addition, the form is adjusted using the availability of important players.

2.2.3 Team characteristics

Several models try to incorporate the characteristics of football teams in some way. This is usually incorporated in the form of attacking and defensive strengths, number of goals scored per match etc. One shortcoming in these models is the failure to capture how the teams actually play. How do the teams build their attacks? What tactical decisions are made by the teams? These, and similar question, are important to ask, as they are definitive of how the teams play, and thus how the match is played (Pollard and Reep (1997), Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014), Hirotsu and Wright (2003)).

Style of play

Pollard and Reep (1997) created a model to capture a team's characteristics. They used a so-called "team possession" as the basic unit of their model. A team possession starts when a player gains possession of the ball, except for when the ball is received from a team member. The receiving player must have good enough control over the ball to be able to deliberately influence the direction of the ball. A team possession ends when one of the following events occur: a) the ball goes out of play b) the ball touches a player of the opposing team (a momentary touch not significantly altering the ball direction is excluded) c) an infringement of the rules takes place.

A team possession consists of several components, like passes, throw-ins etc. To assess the effectiveness of the components, the outcome of a team possession need to be quantified. Several outcomes were considered, such as whether the possession ended in a goal, a shot etc. The authors finally composed a outcome measure called "yield". Each outcome is classified by two variables: type of possession and zone of origin. The type of a possession is either free play or set play (like free kicks). The pitch is divided into six different zones, slicing the pitch in equal parts (Pollard and Reep, 1997).

$$p_{ij} = \sum_{k=1}^{n_{ij}} \frac{p_{ijk}}{n_{ij}} \tag{2.17}$$

p_{ij} denotes the probability of scoring after a team possession of type j (1 for open play, 2 for set play) originating in zone $i = 1, ..., 6$. The probability is calculated using Equation (2.17), where p_{ijk} is the k^{th} team possession of type j originating in zone i and n_{ij} is the total number of team possessions of type j originating in zone i. Each recorded team possession is assigned a value p_{ij}, defined by the outcome of the possession and the zone of origin of the next team possession. For example, if a team possession ends with the other team regaining the ball in zone 3, then the first team possession will be assigned a value of $-p_{31}$, indicating that the initial team possession ended in favor of the opposing team (Pollard and Reep, 1997). Using the values of p_{ij}, the average outcome value for team possessions can be calculated. The average outcome value of a team possession of type j originating in zone i is called the yield y_{ij}. The process of setting the values for all y_{ij} is done iteratively.

The yield value can be used to quantify the actual outcome of a team possession. It can also be used to assess the effectiveness of a particular game strategy, by for example taking the mean yield of all team possessions in which the strategy was used. Different strategies can then be compared. Pollard and Reep (1997) supplies an example of such a comparison, based on statistics from the World Cup of 1986 in Mexico. The example situation is a throw-in in zone 6. Per 1000 team possessions recorded in the cup, throwing the ball towards the goalmouth had a yield of 21.7, compared to 3.50 for a short throw to a nearby team member (Pollard and Reep, 1997).

Pollard and Reep (1997) conclude in their article that fans, media, coaches and players are all sceptical about the suggestion that a statistician might have something useful to provide for a team's tactical analysis, and that a coach's subjective opinions on how to run the game triumphs any number the statistician provide. They suggest using the recorded yields as guidelines on what to base a team's strategy upon. They show examples of actions that provide different yields, such as a zone five free kick; direct shot vs pass to team member, and open field play; running with the ball vs long passes forward vs short passes.

Team formation

Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) take a closer look on what defines a team's formation, and how it can be identified. Their research question is as following: "Given all the player and ball tracking data of a team in a season, what team-based features can adequately discriminate a team's behavior?". They answer this question using an in-depth model of a football match, focusing on team formations.

During a match, each player is assigned a role. A given role can only be assigned to one player at any given time, but players may change roles during the match. A role is described by its position relative to the other roles (the left back plays to the left of the central defenders, etc.). A formation assigns a space on the pitch to each player at every time-frame (capturing 10 frames per second). Identifying a team's formation based on player tracking data can be framed as a minimum entropy data partitioning problem (Bialkowski, Lucey, Carr, Yue, Sridharan, et al., 2014) for each time-frame. An example of such a problem is shown in Figure 2.2.2. This problem can be modelled as a linear assignment problem, which the authors solve by using the Hungarian algorithm (Kuhn, 1955).

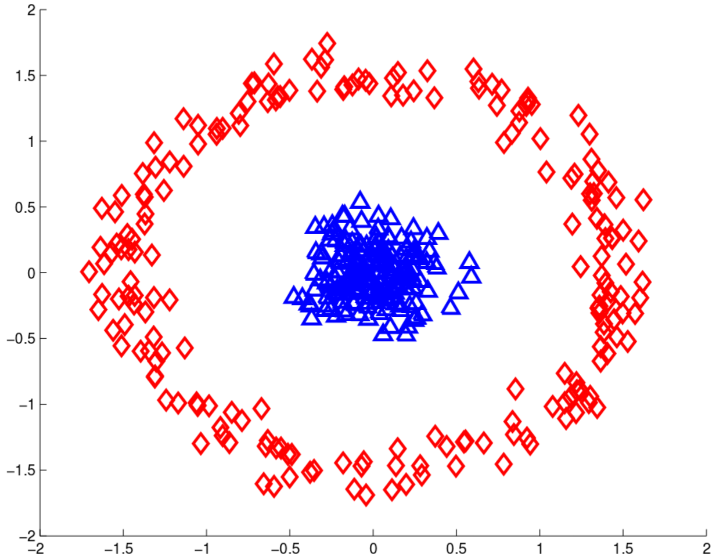


Figure 2.2.2: An example of clustering a ring. Taken from Lee and Choi (2004).

Using only player tracking data and ball events for a given team, Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) created a model for identifying different teams. The model is based on three match descriptors:

- **Match statistics:** Various statistics registered during a match. The statistics capture team and individual behavior, and include variables like corners, catches, goals, bookings, chances, shots, etc. Each match statistics event is associated with a timestamp and a location on the pitch.

- **Ball occupancy:** The pitch is divided into a $10x8$ cell big spatial grid. The ball occupancy is calculated for each cell in the grid, and gives a quantitative description of how often the team was in possession of the ball at each cell during a match. This descriptor captures where the different teams like to put pressure during a match. An example of a ball occupancy map is given in Figure 2.2.3. The map shows how the team tend to attack on the left side of the pitch (Bialkowski, Lucey, Carr, Yue, Sridharan, et al., 2014).

- **Formation descriptor:** The formation of a team is described as above. The formation description is defined as a MxN matrix, where M is the number of cells in the field and N the number of roles (set to 10, excluding keeper). The descriptor describes where the players of different roles tend to move on the pitch. Ten examples of team formation descriptor depictions are shown in Figure 2.2.4.

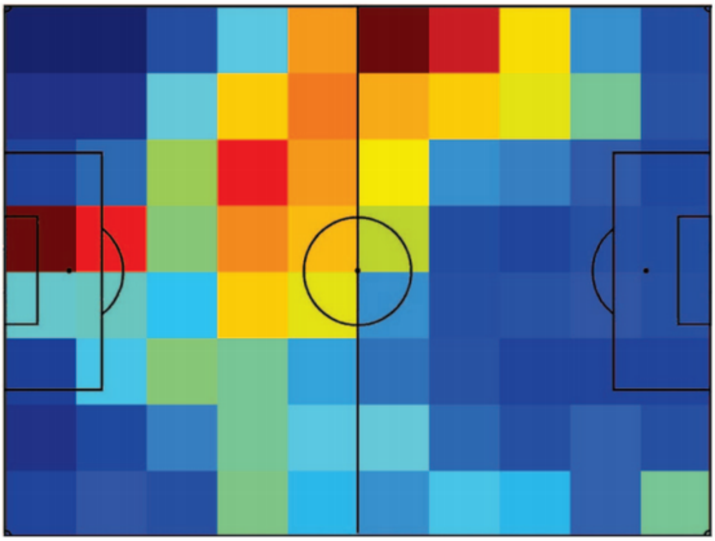


Figure 2.2.3: An example ball occupancy map (over a match half) for a team attacking from left to right. Taken from Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014).

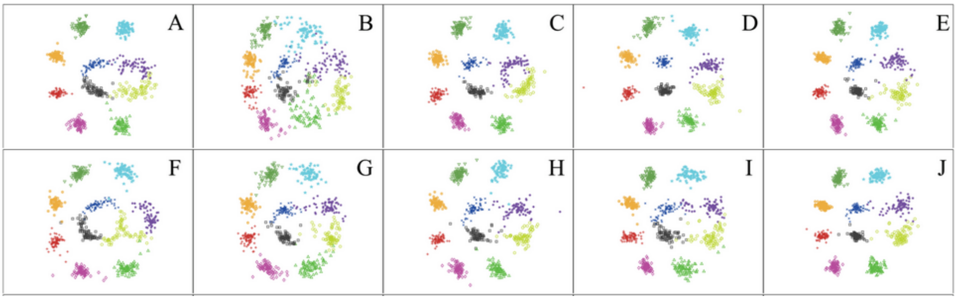


Figure 2.2.4: Depictions of team formation descriptors for a team attacking from left to right. The colors represent the different roles. Only the centroid for each role for each match is depicted. Taken from Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014).

Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) set up an experiment to test the accuracy of their model. The experiment was conducted using ”leave-one-out” cross-validation, training their model on all but one matches for each team. Using data from a top-tier professional soccer league, the model correctly predicted the team in over 70% of the cases. These results clearly show that ”teams have a true underlying signal which can be encapsulated in the way the team moves in formation over time” (Bialkowski, Lucey, Carr, Yue, Sridharan, et al., 2014). In addition, there is also additional information to gain from where different teams put pressure during matches, and how much they interact with the ball throughout a match. This information, combined with the attacking and defensive strengths of a team, might be useful in prediction of match outcomes (Bialkowski, Lucey, Carr, Yue, and Matthews, 2014). For example, knowing that a team plays a lot on their wings, crossing the ball into the goalmouth, whilst the opposing team has good, strong central defenders might tell something about how the match will progress.

The experiments of Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) also showed that teams are rather rigid in the way the play across a season, which they suggest could be used as a powerful prior for preparing for upcoming matches.

Team style is a very subjective and high-level attribute, especially in football, and is therefore hard to segment into discrete parts. Team style covers all aspect of play (Bialkowski, Lucey, Carr, Yue, Sridharan, et al., 2014). The way they quantified team style was by computing a linear combination of prior behavior styles. Given a set of team behavior descriptors, they discovered a discrete set of play styles using k-means clustering. Using different values for k, completely different patterns were discovered. Every team's style was classified uniquely, with each style modelled as a weighted combination of different styles. This makes sense, as a team might play a pressing game one match, and defending the next (Bialkowski, Lucey, Carr, Yue, Sridharan, et al., 2014). Figure 2.2.5 shows an example of how different values for k affect the style clustering descriptors. According to Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014), these descriptors can be used for predicting the results of future matches.

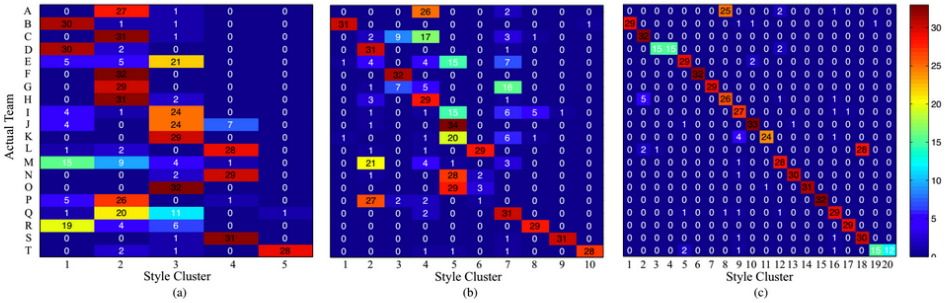


Figure 2.2.5: Clustering descriptor of each match half for different values for k. (a) 5, (b) 10, and (b) 20. Taken from Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014).

In addition to identifying teams based on match data, Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) also used the system for predicting team behavior and how matches are played. Given the identities of two opposing teams, they were able to precisely predict the locations of the players in the different roles in most matches. To do this, they used k-NN regression using the learnt team style priors as input. Their predictions estimated with an average of 2 meters error per role for most matches.

In the future, Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) plan to use their system for both short-term (who will pass to to etc.) and long-term (match outcome) predictions.

2.2.4 Team psychology

In some cases, the psychology of the competing teams can have an impact on the final results.

J. Goddard and Asimakopoulos (2004) mention the importance of ”significant” matches. A match is significant for a team if it is possible (before the match is played) for the team to either win the championship, or to be promoted or relegated, assuming all other teams in contention for the same outcome score one point each on average. According to J. Goddard and Asimakopoulos (2004), teams are more likely to over-perform in significant matches. As a result, if a match is significant for one team, but not for the other, the incentive difference is likely to influence the final result.

Early elimination from knockout tournaments may also influence a team’s results in subsequent league matches. On the one side, a team recently eliminated from a tournament may able to concentrate more on the league, consequently improving their league results. On the other side, elimination might reduce the overall team spirit and confidence of a team, deteriorating the league results. Statistics suggest the latter effect dominates the former (J. Goddard and Asimakopoulos, 2004).

Rue and Salvesen (2000) mention the effect of superiority. If team A is superior to team B, in terms of team strength and historical results, A tend to underestimate B, changing the outcome probabilities in favor of B. The effect is reversed if A is far too superior, so that B develop an inferiority complex facing A (Rue and Salvesen, 2000).

2.3 Measuring the accuracy of a prediction model

Measuring the accuracy of a prediction model is a crucial part of its validation (A. C. Constantinou, N. E. Fenton, and Neil, 2012). There are several ways of evaluating the accuracy of a prediction model, with different degrees of quality.

Anthony C Constantinou, Norman E Fenton, et al. (2012) present five scoring rules, and show why they are not able to correctly evaluate the accuracy of two hypothetical prediction models. They then present the Rank Probability Score (RPS), an alternative scoring rule. The study presents two prediction models, α and β, and their predicted probabilities for the outcomes of five hypothetical matches, numbered $1-5$. Table 2.3.1 shows the five matches together with the predicted probability distributions for the two models. As can be seen from the figure, model α produces the best prediction for all five matches.

Anthony C Constantinou, Norman E Fenton, et al. give the following reasons as to why model α is the best model:

- **Match 1:** Model α predicts the correct outcome with total certainty, and must therefore score higher than model β.

- **Match 2:** Both models assign the highest probability to the correct outcome, with the two other outcomes evenly distributed. Since model α assigns the correct outcome a higher probability than model β, model α must score higher.

- **Match 3:** Both models assign the same probability to the correct outcome. Still,

Match	Model	p(H)	p(D)	p(A)	Result	'Best model'
1	α	1	0	0	H	α
	β	0.9	0.1	0		
2	α	0.8	0.1	0.1	H	α
	β	0.5	0.25	0.25		
3	α	0.35	0.3	0.35	D	α
	β	0.6	0.3	0.1		
4	α	0.6	0.3	0.1	H	α
	β	0.6	0.1	0.3		
5	α	0.5	0.45	0.05	H	α
	β	0.55	0.10	0.35		

Table 2.3.1: Predicted probabilities by the two hypothetical prediction models, α and β, for five hypothetical matches. Taken from Anthony C Constantinou, Norman E Fenton, et al. (2012).

model α is more accurate, as its overall distribution of probabilities is more indicative of a draw than that of model β.

- **Match 4:** Both models assign the same probability to the correct outcome. Still, model α is more accurate, as its overall distribution of probabilities is more indicative of a home win than that of model β.

- **Match 5:** Even though model α predicts the correct outcome with a lower probability than β, the distribution of model α is more indicative of a home win than that of model β. According to Anthony C Constantinou, Norman E Fenton, et al., this is easily explained by considering a gambler who is confident that the home team will not lose, and seeks to place a *lay* bet (a bet that is successful if the home team wins, or the match ends with a draw). If α and β are forecasts by two different bookmakers, bookmaker α will pay less for the winning bet.

Table 2.3.2 shows how five different scoring rules score models α and β. A check mark indicates that the scoring rule correctly considers model α more accurate than model β. A single cross indicates that the scoring rule incorrectly considers the models equally accurate. Two crosses indicate that the scoring rule incorrectly considers model β more accurate than model α.

Equation (2.18) presents the RPS, introduced by Epstein (1969).

$$RPS = \frac{1}{r-1}\sum_{i=0}^{r-1}\left(\sum_{j=0}^{i}(p_j - e_j)\right)^2, \tag{2.18}$$

where r is the number of outcomes ($r = 3$ for football matches), p_j the predicted probability for outcome j, and e_j the observed value for outcome j (1 if j is the observed outcome, 0 otherwise).

The RPS calculates the difference between the cumulative distributions of the predicted and observed probabilities. Lower scores indicate better predictions. Table 2.3.3 shows the

Match (model)	**Binary Decision Score**	**Brier Score**	**Geometric Mean Score**	**Information Loss Score**	**MLLE Score**
1	✓	✓	✓	✓	✓
(α)	1	0	1	0	0
(β)	0	0.02	0.9	0.152	-0.1054
2	✗	✓	✓	✓	✓
(α)	1	0.06	0.8	0.3219	-0.2231
(β)	1	0.375	0.5	1	-0.6931
3	✗	✓	✗	✗	✗
(α)	0	0.735	0.3	1.7369	-1.2039
(β)	0	0.86	0.3	1.7369	-1.2039
4	✗	✗	✗	✗	✗
(α)	1	0.26	0.6	0.7369	-0.5108
(β)	1	0.26	0.6	0.7369	-0.5108
5	✗	✗✗	✗✗	✗✗	✗✗
(α)	1	0.455	0.5	1	-0.6931
(β)	1	0.335	0.55	0.8625	-0.5978

Table 2.3.2: Comparison of different scoring rules. Taken from Anthony C Constantinou, Norman E Fenton, et al. (2012).

calculated RPS values for the predictions of model α and model β. As can be seen from the table, RPS correctly considers model α more accurate than model β for all matches.

When using RPS to evaluate a prediction model over several matches, Anthony C Constantinou, Norman E Fenton, et al. (2012) suggest using either the arithmetic mean over the individual scores, or the total of the individual scores.

Match	**Model**	$\sum_{j=0}^{i=0,1,2} p_j$	$\sum_{j=0}^{i=0,1,2} e_j$	**RPS**
1	α	1, 1, 1	1, 1, 1	(0.0000)
	β	0.90, 1, 1	1, 1, 1	0.0050
2	α	0.80, 0.90, 1	1, 1, 1	(0.0250)
	β	0.50, 0.75, 1	1, 1, 1	0.1562
3	α	0.35, 0.65, 1	0, 1, 1	(0.1225)
	β	0.60, 0.90, 1	0, 1, 1	0.1850
4	α	0.60, 0.90, 1	1, 1, 1	(0.0850)
	β	0.60, 0.70, 1	1, 1, 1	0.1250
5	α	0.50, 0.95, 1	1, 1, 1	(0.1262)
	β	0.55, 0.65, 1	1, 1, 1	0.1625

Table 2.3.3: RPS values for the predictions of model α and model β. Taken from Anthony C Constantinou, Norman E Fenton, et al. (2012).

2.4 Applying predictions to betting

When applying prediction models in a betting setting, the goal of a model changes from predicting the correct result to make a profit from the predictions.

2.4.1 Background

In order to understand how to beat the bookmakers, one must first understand how their odds are formed. The expected margin (gain) of a bookmaker on a football match is represented as.

$$E(M) = 1 - \sum_{i=0}^{2} P_i * w_i * d_i, \tag{2.19}$$

where i correspond to the different outcomes (0 for home victory, 1 for draw, 2 for away victory). The expected margin M of a match depends both on the probability P_i of each outcome, the percentage w_i of bets placed on each outcome, and the given odds d_i for each outcome (Vlastakis, Dotsis, and Markellos, 2009).

Bookmakers are interested in keeping a stable profit, and set their prices accordingly (Vlastakis, Dotsis, and Markellos, 2009). Equation (2.19) implies there are different ways for bookmakers to set their prices. One example is to accurately forecast the game outcomes so that the odds reflects the expectations. Another way is to forecast the distribution of bets placed on each outcome. Finally, it is possible to combine the two (Vlastakis, Dotsis, and Markellos, 2009).

As the true outcome probabilities are not known, the bookmakers can only control the values of d_i. The values of w_i change (usually according to the corresponding d_i), and are controlled by the bettors. Therefore, to ensure a profit, the bookmakers must ensure that

$$\forall i \in \{0,1,2\} : w_i * d_i < 1 \quad \text{or, more generally,} \quad \sum_{i=0}^{2}(1 - w_i * d_i) > 0.$$

To calculate the actual margins for a single match, one need to know both the odds on every outcome and how the bets are distributed. The odds are publicly available, but the bet distributions are not. One can therefore not calculate the actual margin, but rather an *implied* margin. According to Vlastakis, Dotsis, and Markellos (2009), the standard way of doing this is by assuming the bets are evenly distributed across all outcomes. That is,

$$\forall i \in \{0, 1, 2\} : w_i = \frac{1}{3}.$$

It is further assumed that the odds are set according to the true probabilities of each outcome. The true probabilities are usually estimated by the bookmaker's own odds compilers (Vlastakis, Dotsis, and Markellos, 2009). Assuming the odds are set according to the true probabilities, the fair odds for outcome i is then $\frac{1}{P_i}$. However, if the odds are based on the true probabilities, then the expected bookmaker margin would be zero (this can be seen by replacing d_i with P_i^{-1} in Equation (2.19)). Therefore, the actual odds correspond to *implied* probabilities (P_i') somewhat larger than the true probabilities. The expected margin of a bookmaker can then be seen as

$$E(M') = \left(\sum_{i=0}^{2} P_i' \right) - 1 = \left(\sum_{i=0}^{2} \frac{1}{d_i} \right) - 1. \tag{2.20}$$

When Liverpool played at home against West Ham on December 11, 2016, the average odds were 1.29, 6.22, and 10.23 for home, draw, and away, respectively (Odds-Portal, 2016). By Equation (2.20), the expected margins of the combined betting marked was $\frac{1}{1.29} + \frac{1}{6.22} + \frac{1}{10.23} - 1 = 0.034$. The bookmakers were expected to gain a profit of 3.4%. For a prediction model to make a profit against the bookmakers, it must therefore not just beat the predictions of the bookmakers. It must also beat the built-in margins.

2.4.2 Money management strategies

When deciding whether to place a bet, one must compare the probabilities of the outcomes calculated by the prediction model with the odds offered by the bookmakers. The expected gain for a given bet can be calculated as $P_i * d_i - 1$, and bets are usually placed on outcomes where the gain is expected to be positive. It is normal to set a threshold, τ on the minimum discrepancy level allowed. This is done to take the built-in profit margins of the bookmakers into account, increasing the level of confidence needed in order to place a bet.

When an expected profitable match is found, one must be able to decide how much money to place on the bet. Consider the following scenarios:

- **Match 1**: $P_2 = 0.20, d_2 = 6.0$.
- **Match 2**: $P_0 = 0.60, d_0 = 2.0$.

The expected gain for each of the bets is $P_i * d_i - 1 = 0.2$. However, the amount of money to put on each of the bets is not clear. While the odds of Match 2 is clearly lower than that of Match 1, one has to wager more money on Match 2 in order to win the same amount of money. Langseth (2013) presents five different strategies deciding on how much money to place on each bet, based on the outcome probability P_i and odds d_i, and the bankroll C of the bettor. Each strategy output the amount c_i to place on a given bet, where 1 is the unit size. The strategies are as follows:

- **Fixed bet:** The simplest strategy. For each feasible bet, place the same amount of money. $c_i \propto 1$.
- **Fixed return:** Each bet is assured to produce the same profit. $c_i \propto \frac{1}{d_i}$. This results in lower amounts placed on high-odds bets.
- **Kelly ratio:** Kelly (1956) proposed a strategy based on a decision-theoretic approach. In his setup, the utility of having a bankroll of C after a bet is set to $\ln(C)$. The utility for going broke therefore approaches $-\infty$. The expected utility of a bet c_i is $P_i * \ln(C + d_i c_i) + (1 - P_i) * \ln(C - c_i)$. The utility is then maximized by choosing $c_i = C * \frac{P_i d_i - 1}{d_i - 1}$. To reduce the potential of emptying the bankroll in the early stages, Langseth (2013) proposed a modification of the strategy, where the size of c_i is limited not to exceed a predefined value, C_0, chosen to be ”much smaller than the bankroll”.
- **Markowitz portfolio management:** In the Markowitz portfolio management strategy, one look at a collection of bets over a game-week. The goal is to find an allocation of bets that maximizes $\sum_{i=1}^{n}(\mathbb{E}[\Delta_i] - \nu Var[\Delta_i])$ under the constraint that the bets during the game-week sum to a predefined value C_0. ν represents the accepted level of risk. The dual representation of the optimization problem is then to minimize $\sum_{i=1}^{n} Var[\Delta_i]$ under the constraints $\sum_{i=1}^{n} c_i = C_0$ and $\sum_{i=1}^{n} \mathbb{E}[\Delta_i] = \mu$. μ is then the representation of the accepted risk level.

  Langseth (2013) used three different values for μ in his experiments:

  - **Risk-averse:** $\mu = \mu_{\downarrow} = (\sum_{i=1}^{n} P_i d_i)/n - 1$
  - **Risk-seeking:** $\mu = \mu^{\uparrow} = \max_i P_i d_i - 1$
  - **Intermediate:** $\mu = (\mu_{\downarrow} + \mu^{\uparrow})/2$
- **Variance-adjusted:** Rue and Salvesen (2000) proposed a strategy for minimizing the difference between the expected profit and the variance of the profit. According to Langseth (2013), the variance-adjusted strategy can be seen as a simplification of the Markowitz portfolio management strategy. After placing a bet, the difference is $P_i d_i c_i - P_i(1 - P_i)(d_i c_i)^2$, which is minimized by choosing $c_i = (2d_i(1 - P_i))^{-1}$.

In his paper, Langseth (2013) compared the profitability of three prediction models using the presented strategies. The experiments were conducted over the course of two consecutive seasons of the English Premier League. In the first season, each of the different prediction models were able to make a profit. However, the three prediction models had their own respective most profitable betting strategy. For the second season, only a single combination of prediction model and betting strategy was able to make a profit. The profit, however, was of only 0.4%. These results highlight the importance of choosing a betting strategy that suits the prediction model. It should be noted that the threshold τ was 0 during these experiments, and that loosing less than the built-in bookmaker margin therefore can be considered ”decent” (Langseth, 2013).

In addition to the strategies presented by Langseth (2013), Rue and Salvesen (2000) presented another strategy, based on the mean and variance of a bet’s profit. From outcome i of a match, let c_i be the corresponding bet. The mean and variance of a bet’s profit is found using outcome probability P_i and outcome odds d_i. The optimal bet for a match is then found as

$$\arg\max_{c_i>0} U(\{c_i\}), \quad \text{where} \quad U(\{c_i\}) = \text{E(profit)} - \text{Var(profit)} = c_i(\mu_i - c_i\sigma_i^2)$$

The solution is $c_i = \max\{0, \mu_i/(2\sigma_i^2)\}$. In order to place only one bet per match, i is chosen to maximize $c_i\mu_i$.

2.4.3 Biases

In addition to the bookmakers’ built-in margins, there has been observed biases for different kind of bets. These biases are important to consider when placing bets, as they can significantly impact the profitability of a strategy. A. C. Constantinou and N. E. Fenton (2013) present the three most important odds biases:

- **Favorite-longshot bias:** Low-odds bets tend to generate a higher return than high-odds bets. According to A. C. Constantinou and N. E. Fenton (2013), the strongest hypothesis behind this phenomenon is the bettors’ preference in backing risky outcomes. For example, in a match where the probability of a home win is 0.9, even if the bookmaker sets a fair odds of 1.11, a typical bettor is reluctant to place $\pounds 100$ bet only to win $\pounds 11$. If the probability of an away win is 0.05, a typical bettor would be prepared to bet at less than fair odds, for example 15, because a small bet of $\pounds 10$ can potentially return $\pounds 150$.

- **Home-away bias:** A. C. Constantinou and N. E. Fenton (2013) mention how some researchers consider all home wins and away wins to serve as favorite and longshot outcomes, respectively. On average, away wins can be seen as long shots (as indicated by the home ground advantage), but for a significant portion of matches, this does not hold. A. C. Constantinou and N. E. Fenton (2013) demonstrated the home-away bias by comparing the cumulative returns generated by simulating $\pounds 1$ bets on all home wins, draws, and away wins during seven seasons in four different major leagues.

- **Most-likely/least-likely bias:** A. C. Constantinou and N. E. Fenton (2013) also demonstrated how the odds are tailored in favor of the most-likely outcome of a match. By performing a similar simulation to that above, the cumulative loss when betting on all least-likely outcomes was significantly higher than when betting on all most-likely outcomes. These results are in agreement with the favorite-longshot bias, whereby low-odds bets generate higher returns than high-odds ones.

2.5 www.whoscored.com

www.whoscored.com is a football statistics web site, offering free statistics for almost 500 different leagues and tournaments all over the world (as of June 2017). Their database is vast, covering matches all the way back to 1999 for some competitions.

2.5.1 Match statistics

www.whoscored.com offers match statistics on three levels of details: minimum, intermediate, and full.

Minimum details

For matches with minimum available details, only the basic information is available: what teams are playing, time elapsed, final result, and kickoff time. The majority of all competitions fall into this category.

Intermediate details

For matches with intermediate details, some additional information is included in addition to the basic information: lineups, substitutes, and the most important match events (goals, penalty misses, substitutions, and red and yellow cards).

This was the most detailed level of details from August 2002 throughout the season 2008/2009.

Full details

Only a handful of competitions are covered with full match details. In addition to the most basic information, the full details include generic information about the match, such as venue, attendance number, referee, weather conditions, and team managers.

The full details also include a detailed overview of almost every event taking place during the match. Free kicks, tackles, saves, and passes are examples of such events. Each event has several properties, such as location on the pitch, at what time it happened, and players involved. A thorough description of the detailed match data is included in Appendix B.

In addition to event data, the full details contain information about the teams' formations: each formation the team used during the match, with player positions, who was captain, and when the formation started and ended.

The full details also include statistics for each participating player. The statistics consist of several metrics, such as the number of tackles, saves, or passes, and how they developed over time. A list of all available metrics is included in Table B.2.1.

Lastly, the full details include ratings of all involved players, along with a rating for the team as a whole. The ratings are valued on a scale from 0 through 10, where 10 is the top rating. The ratings are calculated based on the player's contribution to the team.

2.5.2 Team and player statistics

www.whoscored.com also offers team and player statistics. Descriptions of the statistics are shown in Table B.3.2 and Table B.3.3.

2.6 Technologies

The following technologies have been utilized in this report.

2.6.1 Regular expressions

A regular expression is a string describing a set of strings, a pattern, following given syntax rules. Regular expressions are widely used in text editors, when searching and replacing text based on patterns. Regular expressions are supported by several programming languages, such as JavaScript and Python.

An important part of regular expressions are meta-characters. Meta-characters describe a set of characters, and allow evaluating logical expressions on the string. Table 2.6.1 shows some of the most central meta-characters.

Character	**Meaning**
ˆ, $	Start and end of string
.	Any character except newline
\\.	Escaping of a meta-character (in this case .)
$[abc]$	Group of characters (in this case, *a*, *b*, or *c*)
$a \mid b$	Logical or of characters (in this case *a* or *b*)
\s, \d, \w	Whitespace character, digit, and word character
{m, n}	Between *m* and *n* inclusive occurrences of the previous meta-character
?	Zero or one occurrences of the previous meta-character
+	One or more occurrences of the previous meta-character
*	Zero or more occurrences of the previous meta-character

Table 2.6.1: List of central meta-characters used in regular expressions.

2.6.2 HTML

HyperText Markup Language (HTML) is the de facto standard markup language for creating web pages. An HTML document is built up by tags describing how the web page should look and behave. Web browsers read HTML documents and present the content accordingly.

2.6.3 JavaScript

JavaScript is a high-level programming language. Together with HTML, JavaScript is a powerful tool supported by just about every modern web browser. JavaScript is versatile, supporting both object oriented-programming and functional programming.

JavaScript is natively supported in HTML, and is extremely useful for creating dynamic web pages. A popular use case for JavaScript is dynamic loading of data without reloading the web page.

2.6.4 Python

Python is a high-level programming language. It supports several programming paradigms, such as imperative, functional, object-oriented, and procedural programming. Python was originally developed as a scripting language, but is today used in several other contexts, such as web servers and desktop applications.

There is a vast community around Python, and countless libraries for everything from web scraping to 3D animation.

2.6.5 Tensorflow

TensorFlow is an open source library for machine learning. TensorFlow performs numerical computations using data flow graphs. The nodes in the graphs represent mathematical operations, while the edges represent the data arrays communicated between the nodes. TensorFlow has a flexible architecture, allowing for parallel computations over several CPUs or GPUs on servers, desktop computers, or mobile devices (TensorFlow, 2017).

2.6.6 Keras

Keras is a high-level ANN API, written in Python. Keras does not implement any form for ANN functionality itself, but runs on top other libraries, like TensorFlow. Keras was developed with a focus on user friendliness, modularity, and extensibility (Keras, 2017).

2.6.7 MySQL

MySQL is a database management system based on Structured Query Language (SQL). SQL is a programming language designed for use with databases. The language is based on relational algebra and calculations, and is mostly used for interacting with relational databases (Codd, 1982).

Chapter 3

Theory

In this chapter, the general theory used in the rest of the report is presented. This chapter serves as a reference for the later chapters where the theory is applied.

3.1 Statistical classification

Statistical classification is the problem of identifying to what set of categories an observation belongs, given a set of observations with known categories (or classes). A far too well-known classification problem is whether or not email should be considered ”spam”. Another well-known example is assigning a diagnosis to a patient, given a description of his/her symptoms and personal characteristics.

When collecting and analyzing large amounts of data, it is often necessary to separate the different data points into classes. As there are few limitations on the dimension of the input data, the classification task can easily become too comprehensive for any human to perform. A digital classifier can then be used instead.

A digital classifier works similarly to how humans preform classification. Just like humans, a digital classifier increases its knowledge using a set of known observations and their corresponding classes (the training set). The classifier then applies its obtained knowledge to determine the classes of new observations. This training method is called ”supervised learning”.

There are several digital classifier algorithms and systems, such as the k-Nearest Neighbor (kNN) algorithm (Peterson, 2009), Support Vector Machines (SVMs) algorithm (Noble, 2006), and ANNs (Hopfield, 1988).

3.1.1 Overfitting

For a digital classifier to perform well, it must accurately assign correct classes to different observations. However, it must not blindly learn the mappings for the observations in the training data set. It is important that a classifier generalizes, so that it can accurately map new observations. If a classifier can accurately assign classes to known observations, without accurately classifying new observations, the classifier ”overfits” on the training data (Hawkins, 2004).

Overfitting usually occurs when a classifier is too complex. When overfitting, the classifier memorizes known observations rather than learning the underlying target function. Unknown observations then result in random error or noise.

3.2 Artificial neural networks

An ANN is a computational model used in computer science, machine learning, and other research areas. ANNs are loosely modeled after the human brain. The human brain contains vast amounts of nerve cells (neurons), which are highly interconnected, creating a huge network of signal transmission. Each neuron receives an electric signal from all the cells it is connected to (its neighbours). If the signal reaches a certain threshold, the neuron sends a signal on to all its neighbours. The procedure repeats itself, propagating electric signals throughout the brain.

Similarly to the human brain, ANNs are built up by connecting a set of simple units called *perceptrons*. [1] A perceptron takes in up to several weighted input values. The input values are summarized, usually together with a weighted constant, called the *bias*. The summed inputs are fed into a function, called the *activation function*. The result of the activation function is called the perceptron *output*. The equation for a perceptron can be written as

$$y = K\left(\sum_{i=1}^{n} w_i x_i + w_0 b\right),$$

where y is the perceptron output, K the activation function, n the number of incoming connections, x_i and w_i the value and weight of the i^{th} incoming connection, and b and w_0 the perceptron’s bias and bias weight. The value of b is usually -1 or 1.

Perceptrons are usually connected in several *layers* with various topologies (see Section 3.2.1). In ANNs, the electric signals are replaced with sets of real valued numbers, called the network *input*. The input travel from the first (input) layer, possibly via intermediate (hidden) layers to the last (output) layer. An example of a simple two-layer ANN is shown in Figure 3.2.1.

Changing the number of layers, number of perceptrons in each layer, or the layer topologies can have huge impacts on how well the network perform, and what it can learn. Another important parameter of an ANN is the perceptrons’ activation functions. The activation function determines what signal the perceptron outputs. There are countless available functions to use. The Hyperbolic Tangent (tanh) function, sigmoid function, softplus function (Glorot, Bordes, and Bengio, 2011), and the Rectified Linear Unit (ReLU) (Nair and G. E. Hinton, 2010) are popular choices.

[1] Perceptrons are often referred to as neurons or nodes. In the context of ANNs, the three are interchangeable.

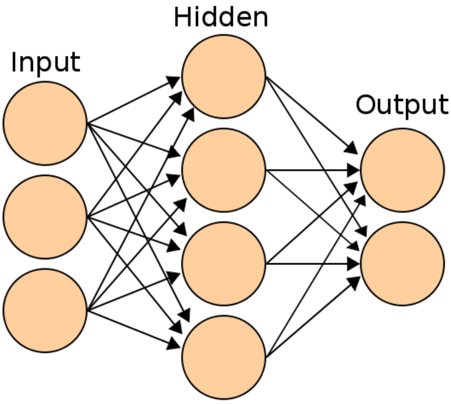


Figure 3.2.1: Example of a simple two-layer feedforward artificial neural network. By Colin M. L. Burnett, distributed under a CC BY-SA 3.0 license.

3.2.1 Network structures

There are several ways to organize the layers of an ANN, depending on what the network is trying to learn. Below is a description of two commonly used network structures that are used later in this report.

Feedforward Neural Networks (FNNs)

An FNN is the simplest type of ANN. In an FNN, the signals move forwards from the input through the output. Each perceptron in one layer is fully connected to all perceptrons in the subsequent layer. An example of an FNN is shown in Figure 3.2.1.

Recurrent Neural Networks (RNNs)

Unlike the connections in an FNN, the connections in an RNN can form cyclic graphs. An RNN contains at least one feed-back connection, i.e. a connection that forms a loop. This creates an internal state of the network, allowing the network to incorporate a form of memory.

The internal memory of RNNs make them extra useful in processing sequences of inputs, such as speech recognition.

3.2.2 The training procedure

There are several ways of training an ANN. One of the most popular training methods is called back-propagation. The method consists of two main phases: forward-propagation of input signals, and backward-propagation of error gradients.

In the forward-propagation phase, input values are fed through the input nodes and propagated through the network, generating an output. The generated output is then compared to the target output, using the *loss function*. A loss function calculates the difference between the generated output and the desired output in some way. The Mean Squared Error (MSE) and Mean Absolute Error (MAE) are examples of such functions.

In the backward-propagation phase, the output errors propagate backwards through the network, calculating an error gradient value for each neuron in the hidden layers. The neuron gradients are fed into the *optimization function*. An optimization function adjusts the neuron weights in order to minimize the loss function. A common way of minimizing the cost function is through methods using *gradient descent*. Figure 3.2.2 shows an example of the progression of the gradient descent method applied to a 2D function. Methods using gradient descent adjust the weights in order to ”descend” the loss function towards a minimum. Popular methods utilizing gradient decent include *stochastic gradient descent* (Bottou, 2010), *RMSProp* (G. Hinton, Srivastava, and Swersky, n.d.), and *Adam* (Kingma and Ba, 2014).

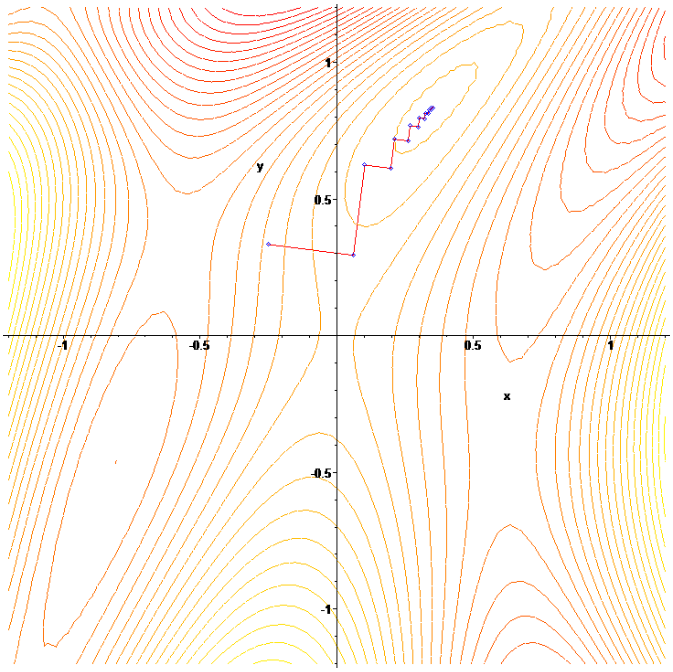


Figure 3.2.2: Example of the gradient descent method applied to a 2D function. By Joris Gillis.

By iterating the back-propagation cycle, the weights eventually converge towards the target function. That is, if the input values are sufficient in order to describe the target function. The neurons in the hidden layers organize themselves so that they learn to recognize the patterns of the input space. Then, if a noisy, unknown observation appears, the network can respond properly if the observation contains the same underlying patterns as a training example.

Dropout

ANNs are, as other classification methods, prone to overfitting. Dropout is a widely used method for avoiding overfitting (Srivastava et al., 2014). Using dropout, for each node, there is a probability p that the node will be ”deactivated”, set to zero and not evaluated during training. Figure 3.2.3 shows an ANN where dropout is applied.

The back-propagation method builds up co-adaptations for the training data. These adaptations do not generalize to unobserved data. Dropout breaks up these adaptations by making the presence of a specific hidden unit unreliable (Srivastava et al., 2014).

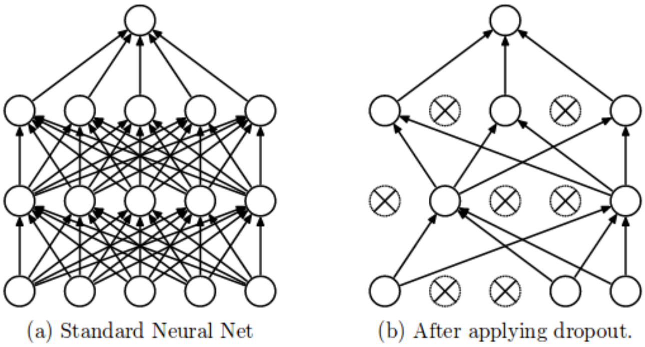


Figure 3.2.3: Graphical illustration of the effects of dropout. Taken from Srivastava et al. (2014).

Chapter 4

System architecture

This chapter presents a description of the architecture of the system used for gathering and evaluating the data used in this report.

4.1 Database overview

To be able to use the data available at **www.whoscored.com** for training an ANN, it should be stored on a computer data storage device in some way. As there are vast amounts of available data, and the data is highly connected, a good option is to store the data in a relational database. Relational databases are designed with relational models in mind, and is therefore a natural choice of storage.

This section describes the database used for storing the data used in this report. The database is set up using MySQL version 5.7.

4.1.1 Central data

Figure 4.1.1 shows the five most central database tables.

A *region* is either a nation, continent, or *International*. International covers tournaments spanning several continents, such as the FIFA World Cup. Continents cover tournaments spanning several countries within the same continent, such as the UEFA European Championship. Nations cover national tournaments, such as the English Premier League. Nations are also used for tracking the nationality of club teams and players.

A *league* is a tournament, either a league or a knockout tournament. A tournament spans over several *seasons*, given by the year the season starts. This makes it possible to differentiate two seasons from each other, such as the 2015-2016 and 2016-2017 seasons of the English Premier League.

A *team* is either a club or national team. Teams are not directly connected to any leagues, as teams can be promoted or relegated.

A *player* is a former or present football player. Players are not directly connected to any teams, as players may change team.

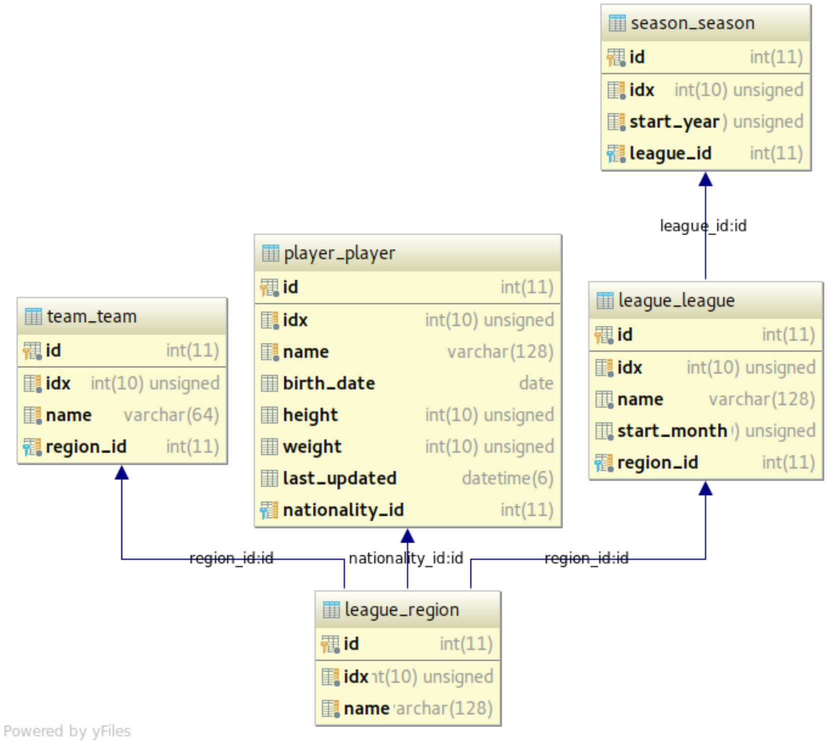


Figure 4.1.1: Most central database tables.

4.1.2 Matches

Figure 4.1.2 shows the database tables containing information concerning matches.

A *match* is a single match between two teams, the *home team* and the *away team*. Matches are played at a specific *venue*, and is conducted by a specific *referee*. All matches are part of a *season*. Matches can be *sparse*, meaning that they are covered by a minimum or intermediate level of details. Complete matches are marked with the Boolean attribute *complete*. Matches that are postponed are marked with the *postponed* attribute. For speeding up data fetching, two attributes, *last_matches_fetched* and *previous_matches_fetched*, marks whether the *last matches* for the two teams, and the *previous matches* between the teams have been fetched.

A *previous meetings* instance lists the previous meetings between the two competing teams of a match. Previous meetings instances list a summary of the number of victories, goals scored, and cards issued for the two teams. For each match, there are three sets of previous meetings: one for the most recent matches, one for the most recent matches played at the home team venue, and one for the most recent matches played at the away team venue. These three sets are stored in the *head to head* table.

It is possible to store a set of *match odds* for a match. A match odds instance lists the odds for the three match outcomes (home victory, draw, or away victory) offered by a given *bookmaker*.

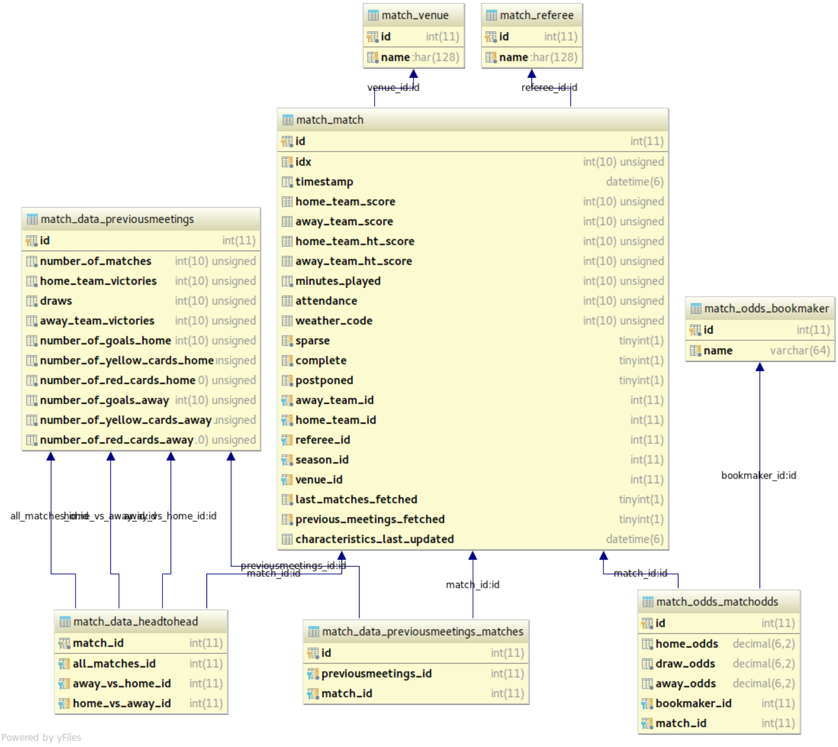

match_venue
id int(11)
name :har(128)
match_referee
id int(11)
name :har(128)
venue_id:id
referee_id:id
match_match
id int(11)
idx int(10) unsigned
timestamp datetime(6)
home_team_score int(10) unsigned
away_team_score int(10) unsigned
home_team_ht_score int(10) unsigned
away_team_ht_score int(10) unsigned
minutes_played int(10) unsigned
attendance int(10) unsigned
weather_code int(10) unsigned
sparse tinyint(1)
complete tinyint(1)
postponed tinyint(1)
away_team_id int(11)
home_team_id int(11)
referee_id int(11)
season_id int(11)
venue_id int(11)
last_matches_fetched tinyint(1)
previous_meetings_fetched tinyint(1)
characteristics_last_updated datetime(6)
match_data_previousmeetings
id int(11)
number_of_matches int(10) unsigned
home_team_victories int(10) unsigned
draws int(10) unsigned
away_team_victories int(10) unsigned
number_of_goals_home int(10) unsigned
number_of_yellow_cards_home ınsigned
number_of_red_cards_home 0) unsigned
number_of_goals_away int(10) unsigned
number_of_yellow_cards_away ınsigned
number_of_red_cards_away .0) unsigned
match_odds_bookmaker
id int(11)
name varchar(64)
bookmaker_id:id
all_matcheshidmie_vs_awaywayidvs_home_id:id
previousmeetings_id:id
match_id:id
match_id:id
match_id:id
match_data_headtohead
match_id int(11)
all_matches_id int(11)
away_vs_home_id int(11)
home_vs_away_id int(11)
match_data_previousmeetings_matches
id int(11)
previousmeetings_id int(11)
match_id int(11)
match_odds_matchodds
id int(11)
home_odds decimal(6,2)
draw_odds decimal(6,2)
away_odds decimal(6,2)
bookmaker_id int(11)
match_id int(11)
Powered by yFiles

Figure 4.1.2: Database tables containing match related information.

4.1.3 Team stats

Figure 4.1.3 shows the database tables containing information concerning team stats.

A *team stats* instance contains information concerning a team's participation in a match. For each match, there are two team stats, one for the home team, and one for the away team. A fully detailed team stats instance contains sets of *ratings* and *statistics*, developed over time, in addition to a final rating. Team ratings are sampled almost every minute and stored in the ratings set. The statistics contains a set for every metric in Table B.2.1.

Each team stats instance has a set of associated *team characteristics*. Characteristics are given by a type (offensive or defensive) and a name. A characteristic is either a strength, a weakness, or a style. Strengths and weaknesses have associated levels, ranging from 15 (very weak) to 55 (very strong).

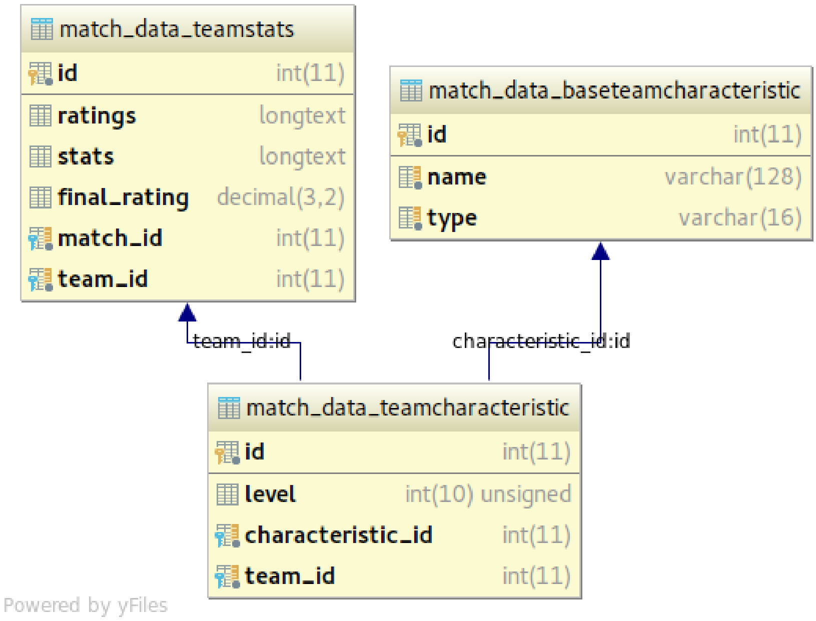


Figure 4.1.3: Database tables containing team stats related information.

4.1.4 Player stats

Figure 4.1.4 shows the database tables containing information concerning player stats.

A *player stats* instance contains information concerning a player's participation in a match. For each match, there are at least 22 player stats, one for each player. For matches with intermediate or full level of details, substitutes are also included. Player stats instances are marked with *minute started* and *minute ended*, marking what parts of the game the player participated in. A fully detailed player stats instance contains *ratings* and *statistics*, similar to those for team stats. In addition, player stats instances contain the final count for every metric in Table B.2.1. Player stats are marked with two Boolean values, *is_first_eleven* and *is_man_of_the_match*, signaling whether the player started the match or was rated man of the match, respectively. A player stats instance is also associated with a *player position*, such as *goal keeper*, *left back*, etc.

Player characteristics, are not associated with a specific match, but rather with the player object itself.

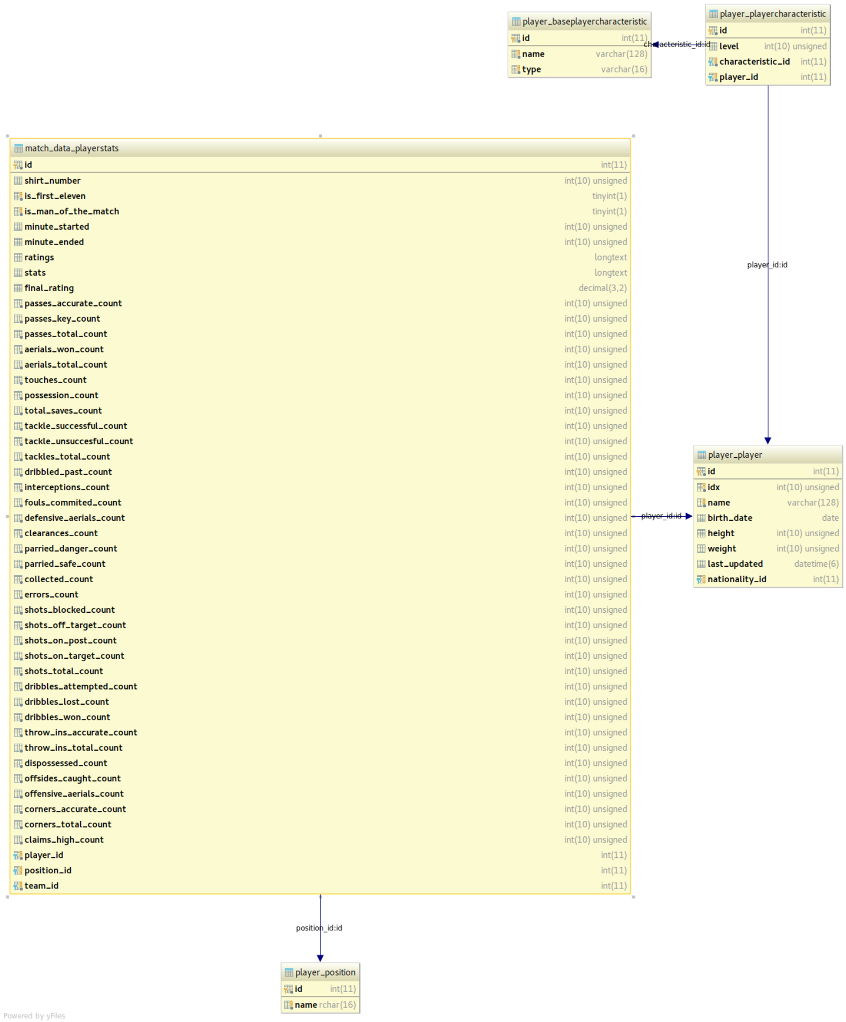

player_baseplayercharacteristic
id int(11)
name varchar(128)
type varchar(16)
characteristic_id:id
player_playercharacteristic
id int(11)
level int(10) unsigned
characteristic_id int(11)
player_id int(11)
player_id:id
match_data_playerstats
id int(11)
shirt_number int(10) unsigned
is_first_eleven tinyint(1)
is_man_of_the_match tinyint(1)
minute_started int(10) unsigned
minute_ended int(10) unsigned
ratings longtext
stats longtext
final_rating decimal(3,2)
passes_accurate_count int(10) unsigned
passes_key_count int(10) unsigned
passes_total_count int(10) unsigned
aerials_won_count int(10) unsigned
aerials_total_count int(10) unsigned
touches_count int(10) unsigned
possession_count int(10) unsigned
total_saves_count int(10) unsigned
tackle_successful_count int(10) unsigned
tackle_unsuccesful_count int(10) unsigned
tackles_total_count int(10) unsigned
dribbled_past_count int(10) unsigned
interceptions_count int(10) unsigned
fouls_commited_count int(10) unsigned
defensive_aerials_count int(10) unsigned
clearances_count int(10) unsigned
parried_danger_count int(10) unsigned
parried_safe_count int(10) unsigned
collected_count int(10) unsigned
errors_count int(10) unsigned
shots_blocked_count int(10) unsigned
shots_off_target_count int(10) unsigned
shots_on_post_count int(10) unsigned
shots_on_target_count int(10) unsigned
shots_total_count int(10) unsigned
dribbles_attempted_count int(10) unsigned
dribbles_lost_count int(10) unsigned
dribbles_won_count int(10) unsigned
throw_ins_accurate_count int(10) unsigned
throw_ins_total_count int(10) unsigned
dispossessed_count int(10) unsigned
offsides_caught_count int(10) unsigned
offensive_aerials_count int(10) unsigned
corners_accurate_count int(10) unsigned
corners_total_count int(10) unsigned
claims_high_count int(10) unsigned
player_id int(11)
position_id int(11)
team_id int(11)
player_id:id
player_player
id int(11)
idx int(10) unsigned
name varchar(128)
birth_date date
height int(10) unsigned
weight int(10) unsigned
last_updated datetime(6)
nationality_id int(11)
position_id:id
player_position
id int(11)
name rchar(16)
Powered by yFiles

Figure 4.1.4: Database tables containing player related information.

4.1.5 Events

Figure 4.1.5 shows the database tables containing information concerning match events.

The *event* table covers almost everything that takes place on the pitch during a match. An event is of a specific *event type*, as listed in Table B.1.1. An event takes place at specific *position* on the pitch, at a given *time* during the match. Most events are executed by a *player* for one of the competing *teams*. Some events, such as *goals* and *missed shots*, have related events. Events are marked with four Boolean attributes, *is_touch*, *is_goal*, *is_own_goal*, *is_shot*. Events that span an area (such as shots, passes, etc.) are marked with an *end position*. Events that enter the goal are marked with a *goal position*, marking where it passed the goal mouth.

An event has a set of *qualifiers*. Qualifiers describe the event in details. A qualifier is of a specific *type*. *Corner taken*, *key pass*, *angle*, *length*, *zone*, *goal kick*, *parried danger*, and *hands* are some examples of qualifier types. Qualifiers such as *angle*, *length*, and *zone* have associated *values*.

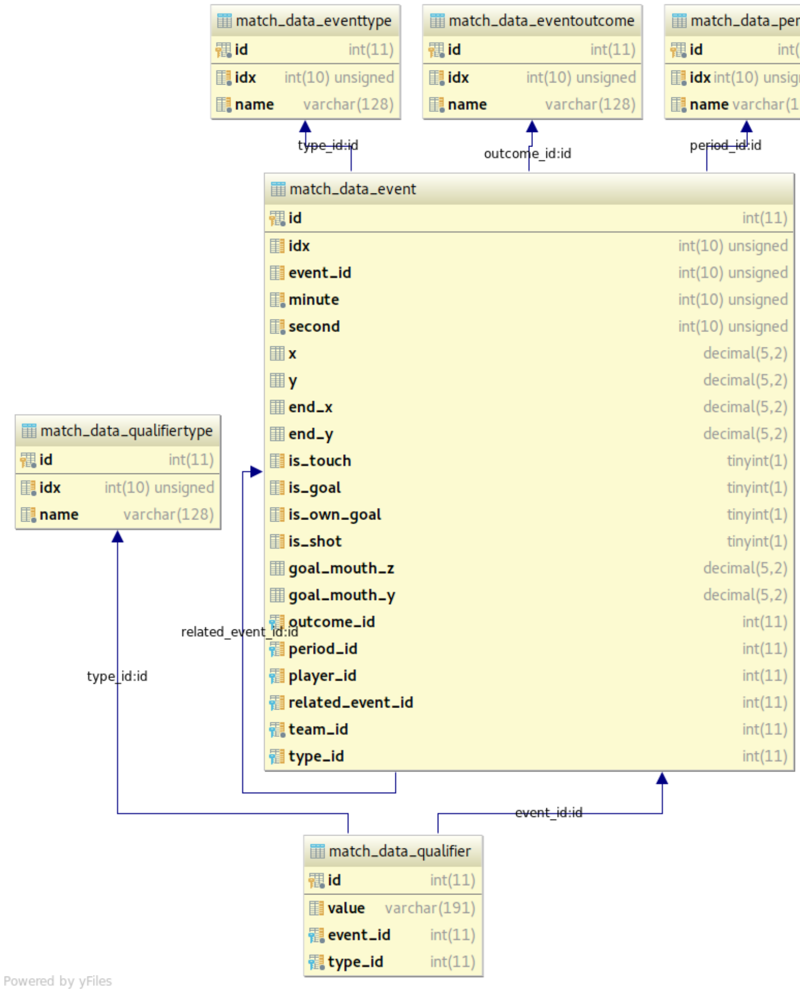


Figure 4.1.5: Database tables containing event related information.

4.1.6 Formations

Figure 4.1.6 shows the database tables containing information concerning team formations.

Team formations are included in matches with full level of details. A team formation concerns a specific *team's* setup during a specific *interval* of a match. Team formations have associated *team captains*. A new team formation is created every time a team makes a substitution or rotates *player positions*. A player position concerns a single player, and where on the pitch he plays at the given team formation.

Each team formation has an associated *formation*. A formation signals how the players are arranged (4-4-2, 4-3-1, etc.).

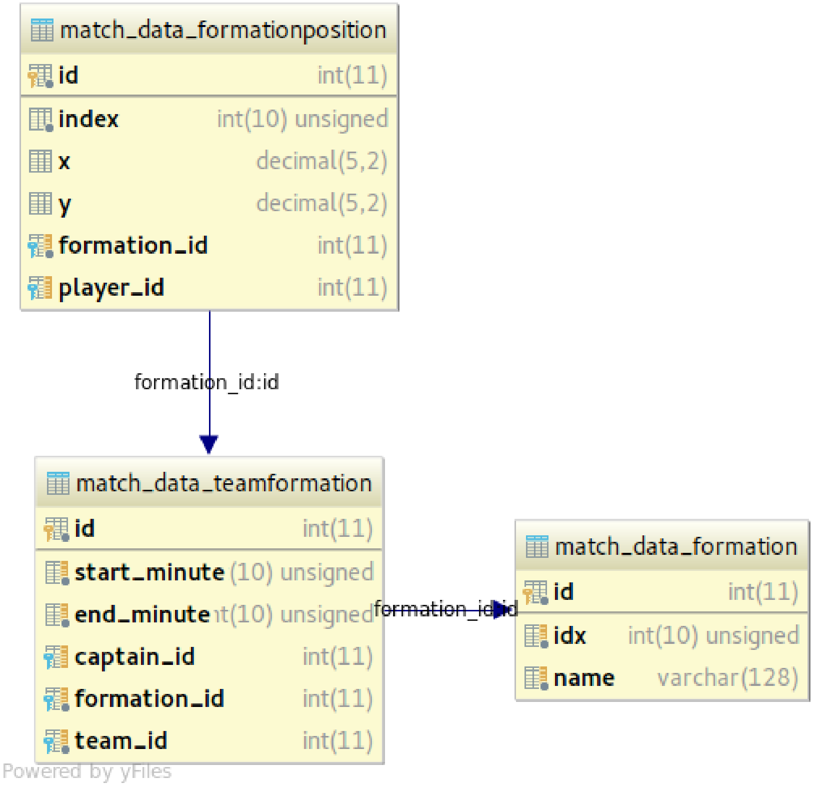
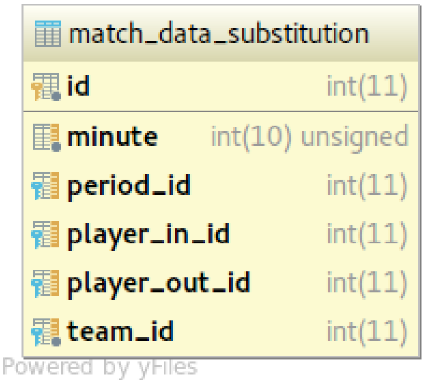


Figure 4.1.6: Database tables containing formation related information.

4.1.7 Substitutions

Figure 4.1.7 shows the database tables containing information concerning substitutions.

Every time a player is substituted, a *substitution* instance is created. A substitution contains information about what players were substituted at what time and period of the match.



match_data_substitution	
id	int(11)
minute	int(10) unsigned
period_id	int(11)
player_in_id	int(11)
player_out_id	int(11)
team_id	int(11)

Figure 4.1.7: Database table containing substitution related information.

4.2 www.whoscored.com crawler

There are several steps one must go through in order to populate the database in Section 4.1. For each season, one must first gather the IDs for all the matches. When the IDs are gathered, one can start traversing all matches, gathering scorelines, event information, player statistics, etc. Below comes a description of how to locate the different data sources, and how to extract the data.

4.2.1 URL structure

When crawling **www.whoscored.com** for data, one must know where the web pages containing the wanted data are located. Every match, team, and player listed on **www.whoscored.com** has its own unique ID, given as an integer number. Regions, leagues, seasons, and season stages also has their own unique IDs. Different entity types have their own URL structure. The only difference between the URLs of two entities of the same kind are their IDs.

4.2.2 Match IDs

Season fixtures are available at the following URLs:

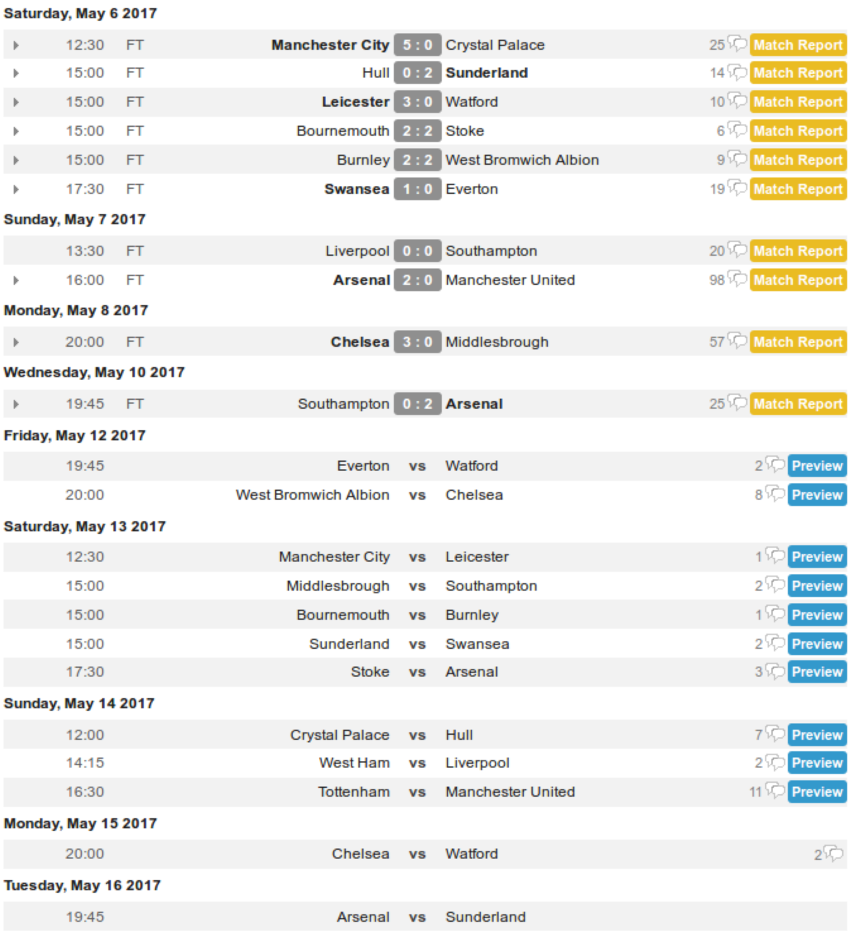
`www.whoscored.com/Regions/REGION/Tournaments/TOURNAMENT/Seasons/SEASON/Stages/STAGE/Fixtures`,

where **REGION** is the region of the tournament, **TOURNAMENT** the tournament in question, **SEASON** the season to fetch fixtures for, and **STAGE** the stage of the season. For league tournaments, the stage ID stays the same. For knockout tournaments, there are different stages per season (qualification, group stage, and knockout stage). The example below points to the fixture list for the English Premier League 2016-2017:

`www.whoscored.com/Regions/252/Tournaments/2/Seasons/6335/Stages/13796/Fixtures`,

where 252 is the region ID for England, 2 the tournament ID for the English Premier League, 6335 the season ID for the English Premier League 2016-2017, and 13796 the stage ID for the season.

Fixtures are grouped by year and month. Finished matches are marked with the final score, while coming matches are marked with ”**vs**”. The matches are listed in a large table, with one row for each match, as shown in Figure 4.2.1.



Saturday, May 6 2017						
12:30	FT	**Manchester City**	5 : 0	Crystal Palace	25	Match Report
15:00	FT	Hull	0 : 2	**Sunderland**	14	Match Report
15:00	FT	**Leicester**	3 : 0	Watford	10	Match Report
15:00	FT	Bournemouth	2 : 2	Stoke	6	Match Report
15:00	FT	Burnley	2 : 2	West Bromwich Albion	9	Match Report
17:30	FT	**Swansea**	1 : 0	Everton	19	Match Report
Sunday, May 7 2017						
13:30	FT	Liverpool	0 : 0	Southampton	20	Match Report
16:00	FT	**Arsenal**	2 : 0	Manchester United	98	Match Report
Monday, May 8 2017						
20:00	FT	**Chelsea**	3 : 0	Middlesbrough	57	Match Report
Wednesday, May 10 2017						
19:45	FT	Southampton	0 : 2	**Arsenal**	25	Match Report
Friday, May 12 2017						
19:45		Everton	vs	Watford	2	Preview
20:00		West Bromwich Albion	vs	Chelsea	8	Preview
Saturday, May 13 2017						
12:30		Manchester City	vs	Leicester	1	Preview
15:00		Middlesbrough	vs	Southampton	2	Preview
15:00		Bournemouth	vs	Burnley	1	Preview
15:00		Sunderland	vs	Swansea	2	Preview
17:30		Stoke	vs	Arsenal	3	Preview
Sunday, May 14 2017						
12:00		Crystal Palace	vs	Hull	7	Preview
14:15		West Ham	vs	Liverpool	2	Preview
16:30		Tottenham	vs	Manchester United	11	Preview
Monday, May 15 2017						
20:00		Chelsea	vs	Watford	2	
Tuesday, May 16 2017						
19:45		Arsenal	vs	Sunderland		

Figure 4.2.1: Small portion of the match fixtures for English Premier League, May 2017.

The row elements contain the ID for each match. Extracting the match ID is as simple as fetching the `data-id` attribute from the row element. The example below shows the row element for Arsenal versus Manchester United May 7, 2017.

```
<tr class="item" data-id="1080862">...</td>
```

Listing 4.1: Hypertext to Arsenal versus Manchester United May 7, 2017.

4.2.3 Match data

Match URLs are of the following form:

`www.whoscored.com/Matches/ID/VIEW`,

where **ID** is the match ID, and **VIEW** is the view. A match has several views, depending on whether it is detailed or not, and whether it is finished, ongoing or not started. The different views are listed in Table B.3.1. The example below points to the `Live` view showing detailed match event information for Arsenal versus Manchester United May 7, 2017.

`www.whoscored.com/Matches/1080862/Live`

Match events

Match event data is located at the `Live` match view. Event data is stored in three different locations, depending on detail level. All match event data is located in JavaScript variables in the source code of the respective web page. Fetching the data can easily be done using simple regular expressions.

The minimum match details are located in a variable called *matchHeader*. A description of the data in the match header is shown in Table B.4.1. The example below is from Arsenal versus Manchester United May 7, 2017.

```
matchHeader.load([13,32,'Arsenal','Manchester United','07/05/2017 16:00:00
    ','07/05/2017 00:00:00',6,'FT','0 : 0','2 : 0',,,'2 : 0','England','
    England']);
```

Listing 4.2: Match header from Arsenal versus Manchester United May 7, 2017.

13 and 32 are the team IDs of Arsenal and Manchester United, respectively. The match started at ”07/05/2017 16:00:00”. The status code 6 indicates that the match is complete. FT indicates that the match was ended at full time. No goals were scored during the first half. Arsenal won the match 2-0. Both teams are located in England.

Extracting the match header from the web page source code can be done using the following regular expression:

```
    /matchHeader\.load\((.+?)\);/
```

Listing 4.3: Regular expression used to extract match header data.

The intermediate match details are located in a variable called *initialMatchDataForScrappers [sic]*. The variable contains the match header, lineups, substitutions and most important match events. Extracting the intermediate match details from the web page source code can be done using the following regular expression:

```
    /var initialMatchDataForScrappers = (.+?);/
```

Listing 4.4: Regular expression used to extract initialMatchDataForScrappers.

The full match details are located in a variable called *matchCentreData*. The variable is divided into three main parts: general information, home team information, and away team information. The included information is described in Section 2.5.1. Extracting the full match details from the web page source code can be done using the following regular expression:

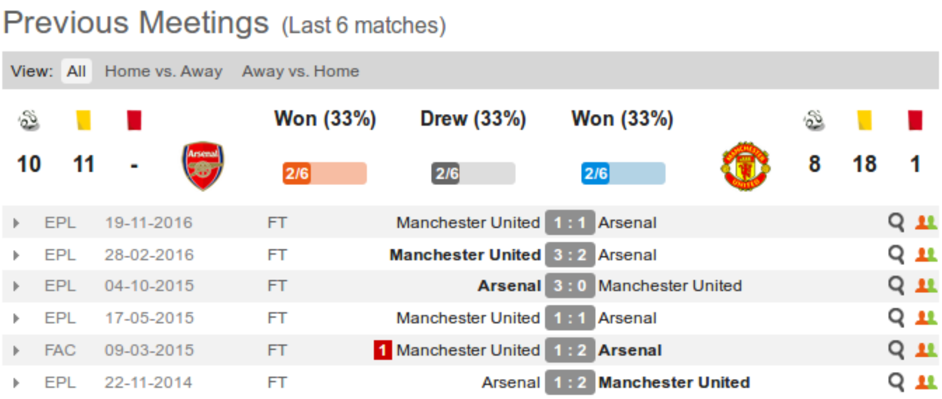
```
/var matchCentreData = (.+?);/
```

Listing 4.5: Regular expression used to extract matchCentreData.

Head to head information

Head to head information is located at the `Show` match view.

Previous meetings (up to six matches) between the two teams are listed in a table, with one row for each match, as shown in Figure 4.2.2. For each match, there is a hypertext to the match web page.



Previous Meetings (Last 6 matches)

View: All Home vs. Away Away vs. Home

⚽	🟨	🟥		Won (33%)	Drew (33%)	Won (33%)		⚽	🟨	🟥
10	11	-		2/6	2/6	2/6		8	18	1

EPL	19-11-2016	FT	Manchester United	1 : 1	Arsenal
EPL	28-02-2016	FT	**Manchester United**	3 : 2	Arsenal
EPL	04-10-2015	FT	**Arsenal**	3 : 0	Manchester United
EPL	17-05-2015	FT	Manchester United	1 : 1	Arsenal
FAC	09-03-2015	FT	1 Manchester United	1 : 2	**Arsenal**
EPL	22-11-2014	FT	Arsenal	1 : 2	**Manchester United**

Figure 4.2.2: Previous meetings between Arsenal and Manchester United before the match May 7, 2017.

The example below is from Arsenal versus Manchester United May 7, 2017, showing a hypertext to their last meeting.

```
<a class="..." href="/Matches/1080633/Live/England-Premier-League
    -2016-2017-Manchester-United-Arsenal">1 : 1</a>
```

Listing 4.6: Hypertext to the last meeting before the match between Arsenal and Manchester United May 7, 2017.

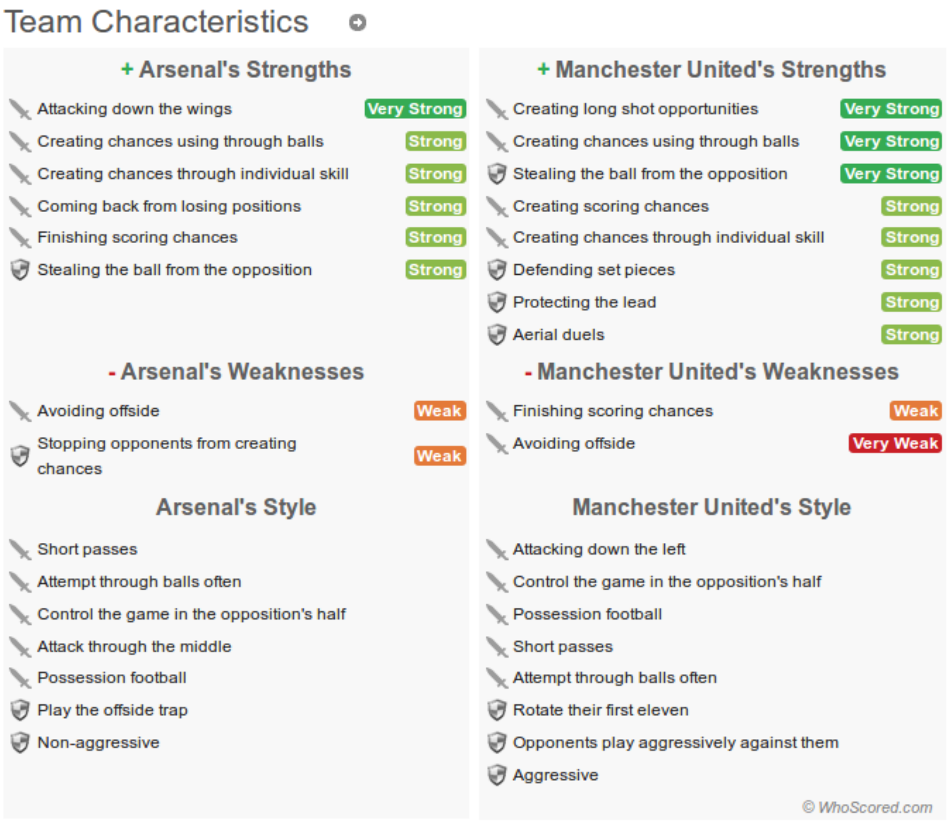
One can extract the match ID from the hypertext using the following regular expression on the hypertext URL:

```
/Matches/(\d+)/
```

Listing 4.7: Regular expression used to extract match ID from hypertext.

Previous matches (also up to six matches) for the two teams are listed in the same way.

Team characteristics for the two teams are listed as three different sets of characteristics: strengths, weaknesses, and styles. All characteristics are given a textual description. Strengths and weaknesses are also given indications of their levels (very weak, weak, strong, very strong).



Team Characteristics

+ Arsenal's Strengths

⚔	Attacking down the wings	Very Strong
⚔	Creating chances using through balls	Strong
⚔	Creating chances through individual skill	Strong
⚔	Coming back from losing positions	Strong
⚔	Finishing scoring chances	Strong
🛡	Stealing the ball from the opposition	Strong

- Arsenal's Weaknesses

⚔	Avoiding offside	Weak
🛡	Stopping opponents from creating chances	Weak

Arsenal's Style

⚔	Short passes
⚔	Attempt through balls often
⚔	Control the game in the opposition's half
⚔	Attack through the middle
⚔	Possession football
🛡	Play the offside trap
🛡	Non-aggressive

+ Manchester United's Strengths

⚔	Creating long shot opportunities	Very Strong
⚔	Creating chances using through balls	Very Strong
🛡	Stealing the ball from the opposition	Very Strong
⚔	Creating scoring chances	Strong
⚔	Creating chances through individual skill	Strong
🛡	Defending set pieces	Strong
🛡	Protecting the lead	Strong
🛡	Aerial duels	Strong

- Manchester United's Weaknesses

⚔	Finishing scoring chances	Weak
⚔	Avoiding offside	Very Weak

Manchester United's Style

⚔	Attacking down the left
⚔	Control the game in the opposition's half
⚔	Possession football
⚔	Short passes
⚔	Attempt through balls often
🛡	Rotate their first eleven
🛡	Opponents play aggressively against them
🛡	Aggressive

© WhoScored.com

Figure 4.2.3: Team characteristics for Arsenal and Manchester United before the match May 7, 2017.

4.3 Neural networks

During the experiments conducted in this report, several ANNs have been explored. As almost all the different networks had the same basic structure, with identical loss functions, optimizer functions, accuracy measures, data sources, etc., the need to re-use the source code soon emerged. To solve this, an abstract network class supporting all the basic functionality was constructed. The abstract network class was implemented with modularity, extensibility, customizability, and ease of use in mind.

4.3.1 The abstract network class

The abstract network class contains all the basic functionality needed in order to train and evaluate an ANN, along with procedures to fetch pre-trained models and use them for prediction of new observations. The abstract network class takes care of everything not problem specific, such as generating training, validation, evaluating, and prediction data sets, storing and handling trained models, storing data sets, etc.

The abstract network class uses Keras to handle its network functionality. Keras supplies functionality for constructing, training, and evaluating ANNs. Keras also supplies functionality for storing and accessing trained models. These models (called model checkpoints) can easily be used for prediction the labels of new observations.

All the different properties of the network class are implemented as functions. By extending the abstract network class, one can easily override the functions to customize the network.

Network name

As each network generates its own data set and checkpoints, there is a need to distinguish one network from another. Each network implemented is therefore given an unique name, describing its function. An example is the network *head-to-head-home-to-result*, that predicts match result given the head to head information for the match.

When each network has its own unique identifier, it is easy to store data sets and checkpoints in separate folders. In addition, Keras supports naming the network layers. Each layer in the network needs an unique name. For supporting networks as combination of other networks, all layer names are prefixed with the network name.

Loss function

Most of the ANNs used in the experiments map some input data to a probability distribution for the three possible match outcomes (home victory, draw, away victory). The loss function is therefore set to *sparse categorical crossentropy* as default. Sparse categorical crossentropy is used for categorical classifying, i.e. for assigning a class to an observation, and not for approximating a real numbered value. When using the sparse categorical crossentropy, labels are given as integers. It is then an easy task assigning a label to a match: 0 for home victory, 1 for draw, 2 for away victory.

Optimizer

The optimizer is set to Adam (Kingma and Ba, 2014) as default. Adam is an adaptive optimizer function that offers some extensions to another popular optimizer: RMSProp (G. Hinton, Srivastava, and Swersky, n.d.). Both Adam and RMSProp compute adaptive learning rates, but Adam also utilizes an exponentially decaying average of gradients, similar to a momentum.

Network layers

One thing that differs between the networks is the network topology. Most networks have different hidden layer topologies. Therefore, the function that defines the hidden layers must be implemented for each network.

Input data shapes also differ from network to network. The function for defining the network's input layer is implemented, but an own function defining the input data shape is not. Each network defines its own input data shape, and the input layer is generated based on the shape.

The output layer is also defined in an own function. Almost all networks have the same output layer: a fully connected layer of three nodes, activated by the softmax function. The softmax function squeezes a K-dimensional vector of real values into K-dimensional vector of real values in the range $[0, 1]$. The values in the output vector sum to 1. This makes the softmax function exceptional for probability distributions.

Setting up the network model

When the above properties are implemented, the network is ready for model training and outcome prediction.

The input layer is set up as model input. The hidden layers are then iterated and added to the model. Then, the output layer is added. Lastly, the model is set up to use the loss function and optimizer specified.

Generating input data

Each network has an unique mapping from match instance to model input data. The function for generating input data must therefore be implemented for each network. However, procedures for storing input data are included in the abstract network class. Storing the input data instead of generating it each time training the network saves quite a lot of time.

Storing and handling trained models

Every time a network is trained, the results are stored by the abstract network class. Each training run is assigned its own folder on the file system.

Keras has implemented a callback function called *ModelCheckpoint*. This callback function allows for storing model checkpoints during training. The callback function allows for storing only the best model checkpoint (every time the loss value of the validation data set reaches a new minimum). Resorting the checkpoint file is then easily done using Keras' API.

Storing and handling data sets

Every time the abstract network class encounters a new match instance, it generates the corresponding model input data. If the match is complete, the input data is added to a dictionary, with the match ID as key. The dictionary is then stored in a file for future use.

If the network encounters a known match, it can simply fetch the model input from the dictionary instead of generating it again.

Training

The abstract network class has an own procedure for training the model. The training procedure sets up the network model using the specified topology, loss function, and optimizer. Training data and validation data are then generated using the model input data generation function. The model is then fitted to the training data, and cross validated using the validation set. The training procedure returns the model checkpoint with lowest validation loss.

The training procedure itself is customizable. The number of training epochs and batch size can be changed using run-time arguments. What matches to include in the training data set can also be changed using run-time arguments.

Evaluating

The abstract network class also has a procedure for evaluating the model. What season to evaluate is set using run-time arguments. The matches for the given season are fed to the model, and the resulting prediction accuracy is shown. The minimum, maximum, and mean RPS values are also shown.

Predicting

Lastly, the abstract network class has a procedure for predicting match instances. Specific matches to predict can be set using run-time arguments. It is also possible to predict all matches in an entire season. For each match, the probability distribution for the three outcomes are shown.

4.3.2 Extending the abstract network class

To construct a valid ANN using the abstract network class, one must create a class that extends the abstract network class. Four properties must be implemented: network name, model input data shape, hidden network layers, and model input data generation function.

When these four properties are implemented, training and evaluating the model, and predicting the label of new observations can easily be done by instantiating the network class and calling the respective functions on the instance.

Example using the head-to-head-home-to-result network

Listing 4.8 shows the complete class for the head-to-head-home-to-result network. The network takes takes a summary of the recent matches between the teams as model input.

The network is further explained in Section 5.1.2.

The *__init__* function takes in the run-time arguments. These arguments are used for generating training, evaluation, and prediction data, and when training the model.

The static function *name* simply returns the name of the network.

The function *input_shape* defines the shape of the model data. Each match is mapped to a list of five floating point numbers.

The function *default_layers* defines the hidden layers of the model. The model consists of a single hidden layer, with 32 nodes. The hidden layer is activated using the sigmoid function.

The function *input_function* maps a match to model input data. Input model data consist of a set of features and a label for each match. A summary of the most recent matches between the teams at the home team venue is fetched. If no such summary exists, the match is skipped, as it does not have the sufficient data for training or predicting. The distributions of match outcomes and goals scored are added as the match's features. The final result is the label of the match.

```python
class HeadToHeadHomeToResultNetwork(AbstractNetwork):
    def __init__(self, args, layers=None):
        super().__init__(args, layers)

    @staticmethod
    def name():
        return 'head-to-head-home-to-result'

    @property
    def input_shape(self):
        return 5,

    @property
    def filters(self):
        super_filters = super().filters

        super_filters.update({
            'previousmeetings__isnull': False,
        })

        return super_filters

    @property
    def default_layers(self):
        return [
            Dense(32, activation='sigmoid'),
        ]

    def input_function(self, match, training_data=True):
        summary = match.previous_meetings.home_vs_away

        if summary is not None:
            return [*summary.victory_distribution, *summary.
                goal_distribution], match.final_result
```

Listing 4.8: The head-to-head-home-to-result network class.

4.4 Betting simulator

To test the profitability of the prediction models evaluated in this report, a betting simulator was created.

The betting simulator simulates one season of a given tournament. The matches for the given season are grouped into ordered game weeks, like in real life. Odds from several bookmakers are considered when deeming bets feasible, and when placing bets.

Each simulation starts with a given bankroll. Algorithm 1 shows the steps taken during the betting simulation.

Algorithm 1: Betting simulation procedure

```
set bankroll, $C$, to initial bankroll size;
for game week $\in$ season do
    for match $\in$ game week do
        calculate match outcome probability distribution;
        for outcome $i \in \{0, 1, 2\}$ do
            fetch all available odds for outcome $i$, $d_{ij}$, from bookmakers
              $j \in \{1, ..., N\}$;
            find the highest odds, $max\{d_{ij}\}$, and the bookmaker $j$ offering it;
            calculate the mean odds, $\overline{d_{ij}}$;
            if bet is feasible, that is $\overline{d_i} * P_i > 1 + \tau$ then
                place bet of size $c_i$ at bookmaker $j$ offering the highest odds;
                update bankroll, removing bet size, $C = C - c_i$;
            end
        end
    end
    for placed bets do
        if bet is successful then
            update bank roll, adding gain from bet, $C = C + d_{ij} * c_i$
        end
    end
end
```

According to the findings of Vlastakis, Dotsis, and Markellos (2009), the European betting marked is inefficient. That is, some bookmakers offer unreasonable high odds, not representative of the overall betting market. This is the reason why mean odds are considered when deeming bets feasible.

Chapter 5

Experimental setup

This chapter presents the setup for the experiments conducted in this report. The chapter first presents a description the prediction models. A presentation of the betting simulation procedure is then presented.

5.1 Prediction models

This section presents the different networks constructed for training prediction models. For each network, the model input is presented, along with a rationale behind the network.

When measuring the performance of a network, different network configurations are evaluated. For each configuration, ten different prediction models are trained. At each training round, the initial weights and biases of the network are randomized, making sure no two models are the same. In addition, the training data set is scrambled randomly before each training round. By doing this, the results present the overall accuracy of the network, and not just the effects of a potentially lucky combination of initial weights and training data.

For each configuration, the minimum, maximum, and mean RPS values are presented. In addition, the prediction accuracy of the model is presented.

Prediction accuracy is evaluated for the 2015-2016 season of the English Premier League. The most promising configuration for each network is then used for evaluating the profitability of the prediction models over the span of the seasons 2015-2017 of the English Premier League. The 2016-2017 season is not included when measuring prediction accuracy, to give a more realistic betting simulation. The results are not that credible if one choose network configuration based on the same data for which the model will be evaluated.

When training the prediction models, matches all the way back to the 2009-2010 season of the English Premier League are used. This is the first season where **www.whoscored.com** offered fully detailed match information. As the data available at **www.whoscored.com** is not perfect, matches with incomplete data are excluded from training.

5.1.1 Player ratings

The player ratings provided by **www.whoscored.com** are calculated using over 200 statistics, and provide an accurate measure on a player's contributions to the team (WhoScored.com, 2017). The player ratings are adjusted throughout the whole match. Every event of importance is taken into account when calculating the rating.

Before using player ratings for predicting match outcome, one should confirm that the ratings actually have predictive properties. To confirm this, a simple FNN with one hidden layer of 64 nodes, activated using ReLU was set up. The network takes in the final rating of the 22 starting players for each match. Figure 5.1.1 shows the network structure.

The input values $I_1, ..., I_{22}$ are the final ratings for the 22 starting players, divided by 10 to map them to the range $[0, 10]$. The players are first ordered by team. The first 11 input values are players from the home team, and the next 11 are from the away team. The players in each team are also ordered, first by x position on the field, then by y position on the field. The position $(0, 0)$ is at each team's right corner flag. This makes the keeper the first player, and the forwards the last.

For the match between Arsenal and Manchester United May 7, 2017, shown in Figure 5.1.2, the input values would be as follows:

```
[0.76, 0.70, 0.74, 0.68, 0.81, 0.72, 0.75, 0.73, 0.72, 0.70, 0.77,
    0.60, 0.67, 0.62, 0.65, 0.66, 0.65, 0.67, 0.61, 0.69, 0.66, 0.65]
```

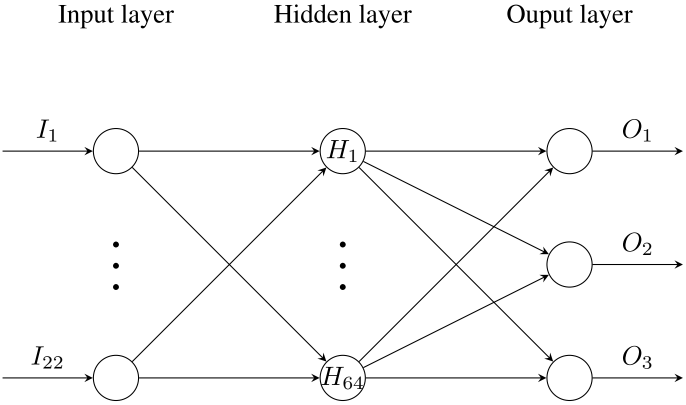


Figure 5.1.1: Player ratings network structure.

The network can with 90% certainty predict the match outcome (averaged over ten network instances), yielding an average RPS of 0.0394. This indicates the player ratings say a lot about the final rating.

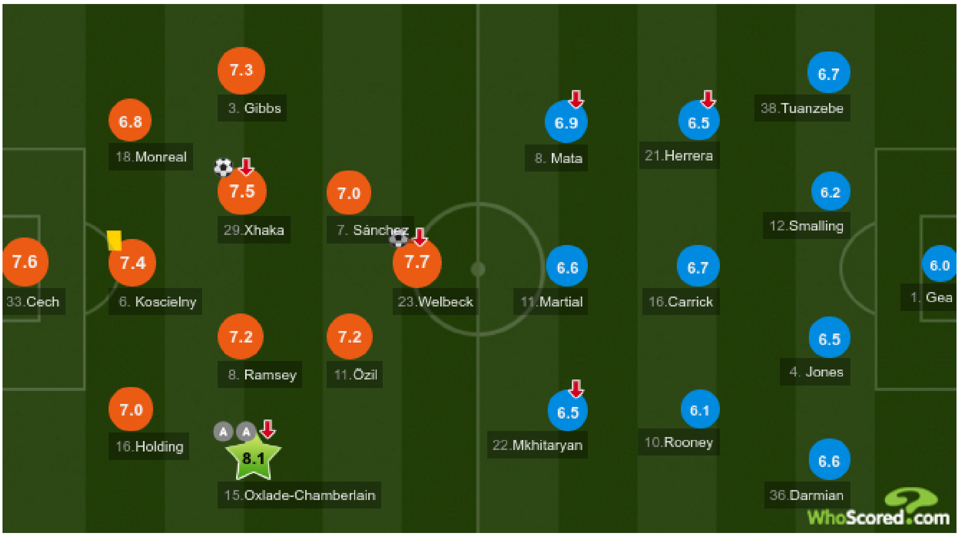


Figure 5.1.2: The final ratings for the players starting the match between Arsenal and Manchester United May 7, 2017.

Input

As the final ratings are not known at match start, ratings from previous matches must be used. For each starting player, the three most recent matches are taken into consideration. The following values from the three previous matches of each player are added to the model input:

- Player's final rating.
- Portion of the match played: The number of minutes played divided by 120. This is added in order to increase the impact of players playing the whole match. 120 is used instead of 90 to support matches that might go to extra time.
- Player's team's final rating. This is added to capture cases where the player rating are affected by of the team's collective effort.
- Other team's final rating. This is added to capture cases where the player rating are affected by of the other team's collective effort.
- A Boolean value, indicating whether the player played at home or away. This is added to take the home ground advantage into consideration.
- Days since the match, $exp(-\text{days since match}/7)$. This is added in order to increase the impact of more recent matches.

In addition to the previous match features, the number of matches the last two game weeks are added for each player. That gives $3 * 6 + 1 = 19$ features for each player. With 22 players, there are $22 * 19 = 418$ features per match.

5.1.2 Head to head

The idea behind this network is to capture match history and the home ground advantage. The network is inspired by the earlier works presented in Section 2.1.

Input

The network takes in a *PreviousMeetings* (see: Section 4.1.2) object containing information about the last matches played at the home team ground.

For each match, the following values are added to the model input:

- Distribution of outcomes over the previous registered matches. For each of the three outcomes, the number of occurrences of the outcome is divided by the total number of matches.
- Distribution of goals over the previous registered matches. The number of goals scored by the home team, divided by the total number of goals scored, and the number of goals scored by the away team, divided by the total number of goals scored.

For the match between Arsenal and Manchester United May 7, 2017, the input values would be as follows:

```
    [1/3, 1/3, 1/3, 7/12, 5/12]
```

Over the last six matches, Arsenal has won two, Manchester United has won two, and there has been two draws. Arsenal has scored seven goals, and Manchester United has scored five.

5.1.3 Previous meetings

This model can be seen as an extension to the head to head model. The model incorporates the same information at the head to head model, but for each of the previous matches. In addition to the outcome distribution and the goal distribution, this model adds the final ratings of the teams and the final result for each of the previous matches.

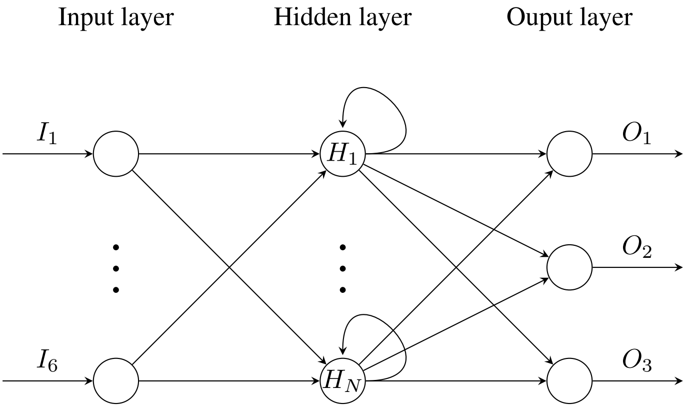


Figure 5.1.3: Previous meetings network structure. N is the number of nodes in the hidden layer.

Input

The following values are added for each of the previous matches:

- Days since the match, $exp(-\text{days since match}/730)$. This is added in order to increase the impact of more recent matches.
- Goal distribution for the match.
- Home team final rating.
- Away team final rating.
- Final result of the match. 0 for home victory, 0.5 for draw, 1 for away team victory.

For each previous match, there are six values. With up to six previous matches, there is a total of $6 * 6 = 36$ features for each match. If two teams have less than six previous matches, a list of six zeros are supplied for each missing match.

For the match between Arsenal and Manchester United May 7, 2017, the input values would be as follows:

```
    [[0.451, 3/3, 0/3, 0.77, 0.63, 0.0]
    [0.293, 1/3, 2/3, 0.65, 0.71, 1.0]
    [0.199, 1/2, 1/2, 0.73, 0.71, 0.5]
    [0.133, 1/2, 1/2, 0.67, 0.67, 0.5]
    [0.071, 1/3, 2/3, 0.67, 0.71, 1.0]
    [0.049, 1/1, 0/1, 0.71, 0.64, 0.0]]
```

The hidden layer is a fully connected simple recurrent layer. Figure 5.1.3 shows the structure of the network. When feeding a match through the network, the previous matches are fed through the hidden layer one by one. The hidden layer output from one match is added to the input for the next.

5.1.4 Team characteristics

This model uses the ideas from Section 2.2.3. The model also incorporates some of the ideas from A. C. Constantinou, N. E. Fenton, and Neil (2012), whereas teams are evaluated anonymously. Teams are represented as the set of their characteristics at match start.

The rationale behind this model is to capture what makes teams win against some opponents, but lose against others. Aside from team strengths and player abilities, what decides a football match? The idea is that the model will capture cases where one team has characteristics that give an advantage over the other team's characteristics. For example if a strength of the home team is "Creating long shot opportunities", whilst a weakness of the away team is "Defending against long shots".

Input

Each team is represented as 43 values in the range $[0, 1]$. Each value corresponds to a team characteristic (see: Section 4.1.3). Values corresponding to present styles are set to 1. Strengths and weaknesses are divided by 100 to fit them into the range $[0, 0.55]$. With 43 values for each team, that gives a total of $43 * 2 = 86$ features for each match.

For the match between Arsenal and Manchester United May 7, 2017, the input values would be as follows:

```
    [1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.55, 0.55,
        0.45, 0.45, 0.0, 0.0, 0.0, 0.25, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.15, 0.0, 0.0, 0.0, 0.0, 0.45, 0.0, 0.0,
         0.0, 0.45, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0,
         0.0, 1.0, 0.0, 0.25, 0.0, 0.45, 0.0, 0.45, 0.55, 0.45, 0.15, 0.0,
         0.0, 0.45, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.55, 0.0, 0.0, 1.0, 0.55, 0.0, 1.0, 1.0]
```

The first 43 values are Arsenal characteristics. The last 43 are Manchester United characteristics.

5.1.5 Team characteristics and strengths

This model is an extension of the team characteristics model. In addition to the team characteristics, the model incorporates team strengths. The team strengths are given as final ratings of the starting players after the team's last match, as well as the most recent final rating for the team itself.

Input

In addition to the 43 values for the team characteristics, there are eleven values, one for each player. Lastly, there is an additional value, for the team rating. With $43+11+1 = 55$ features for each team, there is a total of $55 * 2 = 110$ features for each match.

For the match between Arsenal and Manchester United May 7, 2017, the input values would be as follows:

```
    [0.77, 0.56, 0.67, 0.61, 0.53, 0.65, 0.60, 0.65, 0.63, 0.72, 0.65,
        1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.55, 0.55,
         0.45, 0.45, 0.0, 0.0, 0.0, 0.25, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.15, 0.0, 0.0, 0.0, 0.0, 0.45, 0.0, 0.0,
         0.0, 0.45, 1.0, 0.0, 1.0, 0.0, 0.63, 0.70, 0.72, 0.71, 0.68,
        0.72, 0.69, 0.75, 0.82, 0.68, 0.78, 0.71, 1.0, 1.0, 1.0, 0.0, 0.0,
         0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.25, 0.0, 0.45, 0.0, 0.45, 0.55,
        0.45, 0.15, 0.0, 0.0, 0.45, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.55, 0.0, 0.0, 1.0, 0.55, 0.0, 1.0,
         1.0, 0.0, 0.70]
```

5.2 Betting simulation

When evaluating a network in a betting situation, the most promising configuration of the network is used. Ten different prediction model instances are trained, as when measuring prediction accuracy.

To evaluate the profitability of the network, for each network instance, four different betting strategies are used:

- Fixed bet, with $c_i = 1$
- Fixed return, with $c_i = 1/d_i$
- Variance adjusted
- Kelly ratio, with $C_0 = 0.05$

Each combination of model instance and strategy is evaluated over the span of two seasons of the English Premier League. The initial bankroll is set to 100.

For each model instance, the results for each strategy are presented. The development of the ROI over the span of the season is plotted. The least profitable and most profitable simulations for each strategy are plotted together with the strategy mean.

Chapter 6

Experiments and results

This chapter presents the experiments conducted and the results achieved in this report.

Layer sizes marked with * and ** suffered from overfitting. To overcome the overfitting, dropout was applied. * indicates dropout with $p = 0.1$. ** indicates dropout with $p = 0.2$.

For each network, the share of matches that ended in a home victory is presented. Any accuracy above that level indicates the model has learned something other than to always predict home victory as the most probable outcome.

For each network, strategy, and season, graphs present the development of the ROI generated by the combination of network and strategy over the given season. The graphs show the least profitable, most profitable, and average ROI achieved.

For each network and season, graphs showing the connection between predicted probabilities and available odds for every feasible bet were plotted. The lower frontiers of the graphs follow the line $1/P_i$, as bets below the line are never deemed feasible. If $d_i < 1/P_i$, then $d_i * P_i < 1$, and the bet will have a negative expected gain.

6.1 Benchmark values

To properly evaluate the networks evaluated in this report, we need some benchmark values. The benchmark values indicate the absolute minimum of what the networks should achieve.

A natural benchmark model is based on the home ground advantage. Of all matches played in the English Premier League August 2009 though May 2015, 46.46% ended with a home victory, 25.45% ended draw, and 28.09% ended with an away victory. Table 6.1.1 shows the RPS values achieved when predicting all matches in the 2015-2016 and 2016-2016 seasons with the probability distribution [0.4646, 0.2545, 0.2809].

Another potential benchmark model is to use the implied probabilities of the bookmakers. The networks are, after all, trying to beat the bookmakers, and not the home ground advantage. Table 6.1.2 shows the RPS values achieved when assigning the av-

	RPS values		
Season	**Min**	**Max**	**Mean**
2015-2016	0.141	0.355	0.227
2016-2017	0.145	0.360	0.227

Table 6.1.1: RPS values achieved when predicting all matches with the same probability distribution, [0.4646, 0.2545, 0.2809].

erage implied probabilities of a match as its probability distribution. Using the implied probabilities achieved the best results, and will therefore be used as benchmark values.

	RPS values		
Season	**Min**	**Max**	**Mean**
2015-2016	0.0333	0.683	0.210
2016-2017	0.0187	0.733	0.192

Table 6.1.2: RPS values achieved when predicting all matches with the probability distribution formed by the bookmaker's implied probabilities.

6.2 Player ratings

6.2.1 Network structure

Table 6.2.1 shows the RPS values and accuracy of the player ratings network. In the evaluation data set, 40.8% of all matches ended in a home victory.

Using a hidden layer with 256 nodes activated by the tanh function yielded the most promising results, and will therefore be used when evaluating the profitability of the network. Table 6.2.2 shows the RPS values and prediction accuracy when evaluating the same configuration over the 2016-2017 season. Unfortunately, the network did not perform better than the benchmark model for any of the seasons.

Hidden layer		**RPS values**			
Activation	**Size**	**Min**	**Max**	**Mean**	**Accuracy**
ReLU	32	0.0622	0.623	0.232	0.397
ReLU	64	0.0349	0.565	0.230	0.416
ReLU	128	0.0567	0.624	0.232	0.418
ReLU	256	0.0529	0.631	0.238	0.391
Sigmoid	32	0.0377	0.640	0.233	0.413
Sigmoid	64	0.0234	0.746	0.239	0.413
Sigmoid	128	0.0501	0.631	0.232	0.424
Sigmoid	256	0.0484	0.637	0.236	0.410
Sigmoid	512*	0.0265	0.711	0.241	0.408
Tanh	32	0.0728	0.569	0.234	0.386
Tanh	64	0.0368	0.732	0.239	0.399
Tanh	128	0.0349	0.647	0.234	0.402
Tanh	256*	0.0674	0.554	0.228	0.426
Tanh	512*	0.0467	0.652	0.233	0.413

Table 6.2.1: Prediction accuracy of the player ratings network, with different hidden layer configurations. The row colored green shows the configuration with most promising results.

RPS values			
Min	**Max**	**Mean**	**Accuracy**
0.0390	0.721	0.223	0.507

Table 6.2.2: Prediction accuracy of the player ratings network for the 2016-2017 season of the English Premier League, using the most promising hidden layer configuration.

6.2.2 Betting results

English Premier League 2015-2016

Figures 6.2.1 to 6.2.4 show the development of the ROI generated by the player ratings network over the English Premier League season 2015-2016.

Table 6.2.3 shows a summary of the ROI values achieved by the different strategies when used by the player ratings network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

Figure 6.2.5 shows the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by a random instance of the player ratings network.

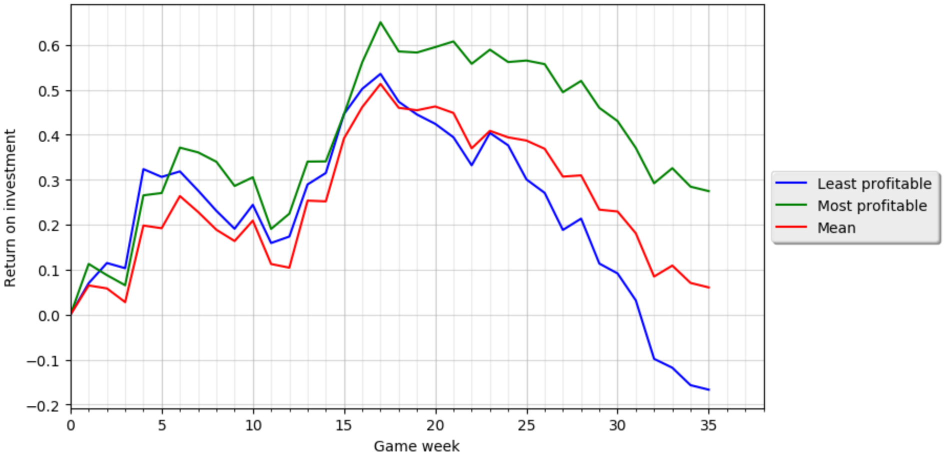


Figure 6.2.1: ROI over the span of the English Premier League season 2015-2016 using the player ratings network and the fixed bet strategy.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.16	0.28	0.068
Fixed return	-0.098	0.022	-0.040
Kelly ratio	-0.99	1.4	0.25
Variance adjusted	-0.14	0.010	-0.073

Table 6.2.3: Final ROI values for the four strategies when using the player ratings network during the 2015-2016 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

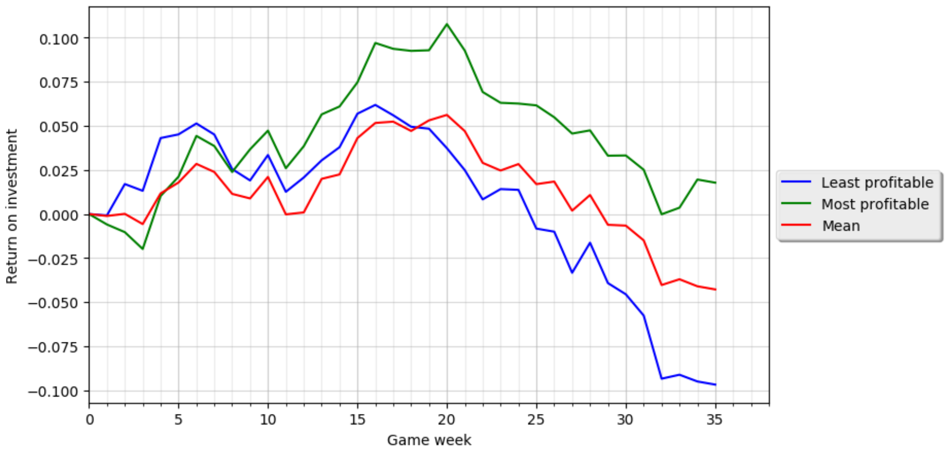


Figure 6.2.2: ROI over the span of the English Premier League season 2015-2016 using the player ratings network and the fixed return strategy.

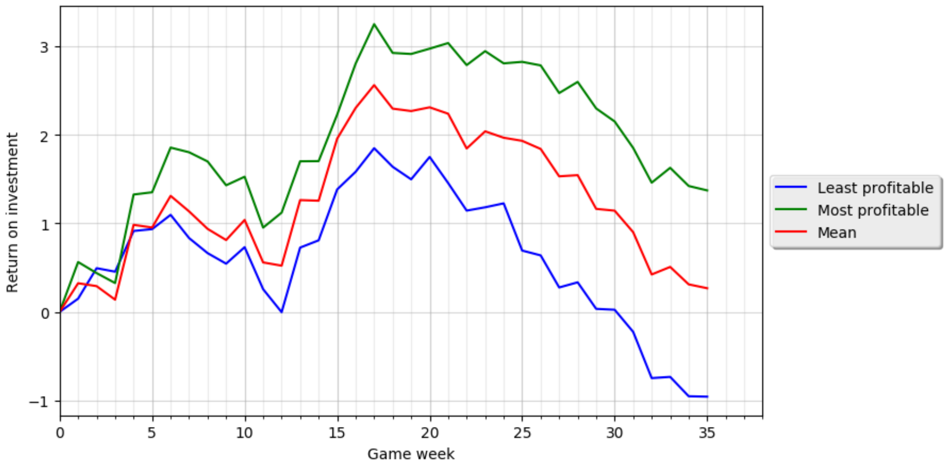


Figure 6.2.3: ROI over the span of the English Premier League season 2015-2016 using the player ratings network and the Kelly ratio strategy.

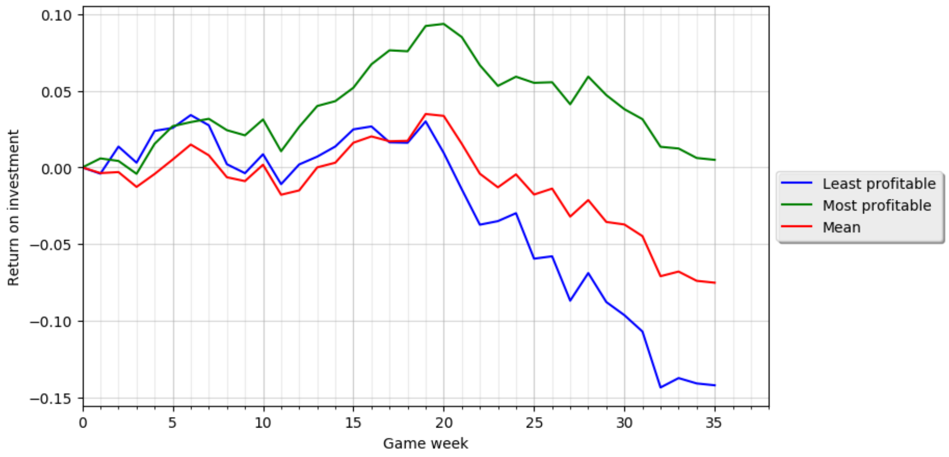


Figure 6.2.4: ROI over the span of the English Premier League season 2015-2016 using the player ratings network and the variance adjusted strategy.

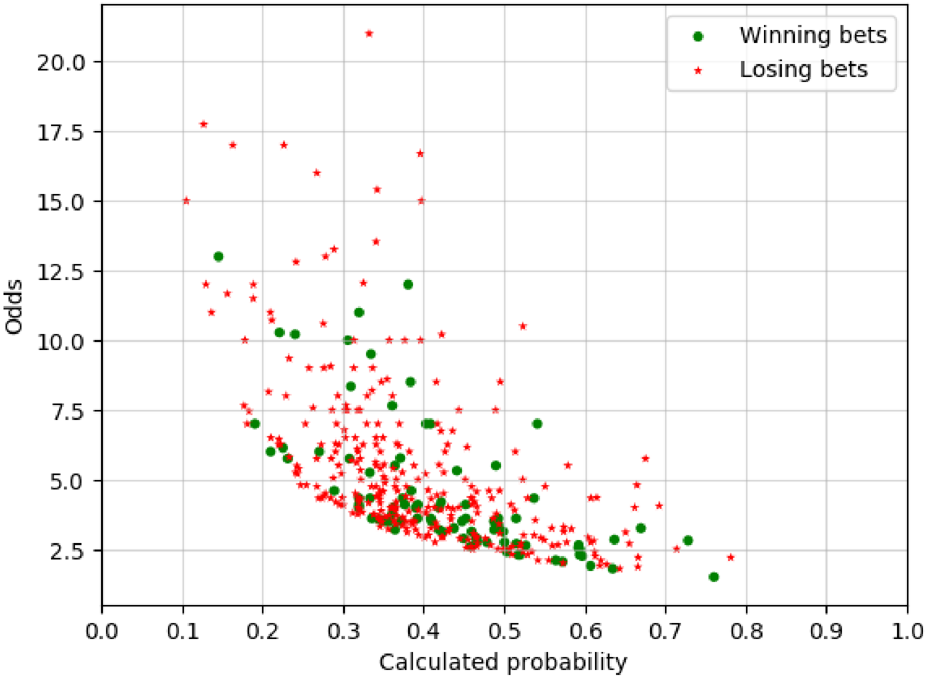


Figure 6.2.5: Offered odds and predicted probabilities for the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by the player ratings network.

English Premier League 2016-2017

Figures 6.2.6 to 6.2.9 show the development of the ROI generated by the player ratings network over the English Premier League season 2016-2017.

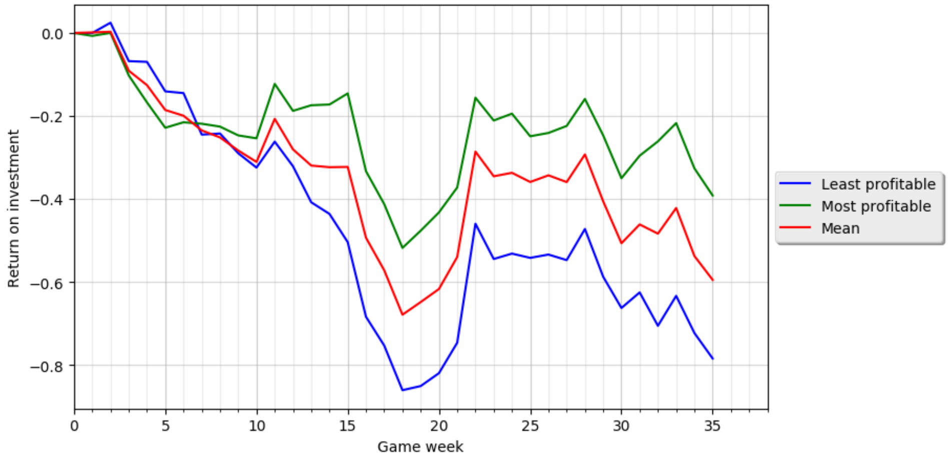


Figure 6.2.6: ROI over the span of the English Premier League season 2016-2017 using the player ratings network and the fixed bet strategy.

Table 6.2.4 shows a summary of the ROI values achieved by the different strategies when used by the player ratings network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.78	-0.38	-0.59
Fixed return	-0.14	-0.050	-0.095
Kelly ratio	-1.0	-1.0	-1.0
Variance adjusted	-0.098	-0.013	-0.053

Table 6.2.4: Final ROI values for the four strategies when using the player ratings network during the 2016-2017 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

Figure 6.2.10 shows the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by a random instance of the player ratings network.

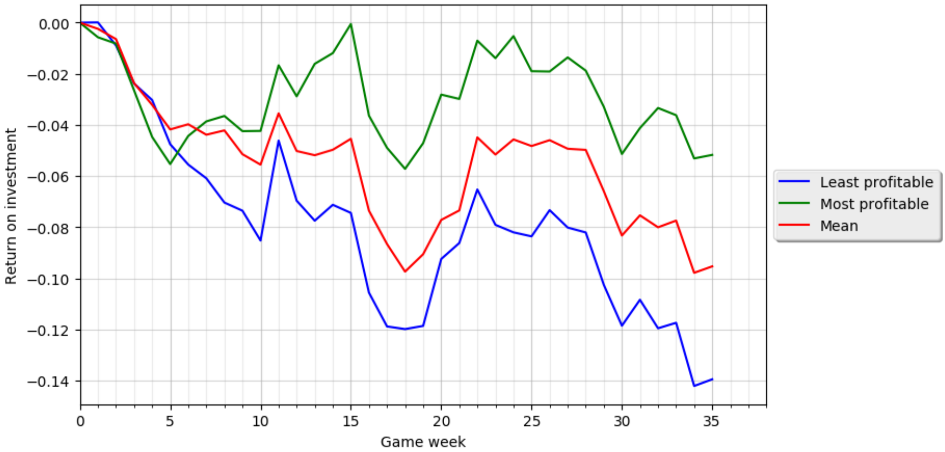


Figure 6.2.7: ROI over the span of the English Premier League season 2016-2017 using the player ratings network and the fixed return strategy.

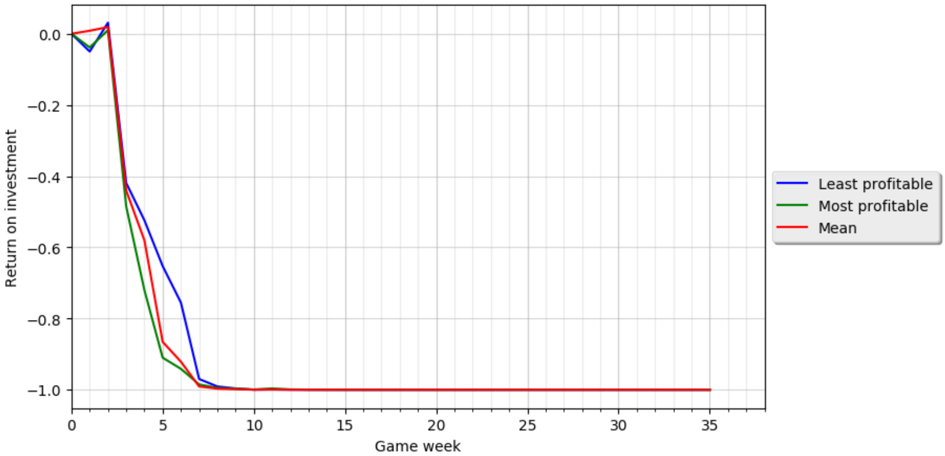


Figure 6.2.8: ROI over the span of the English Premier League season 2016-2017 using the player ratings network and the Kelly ratio strategy.

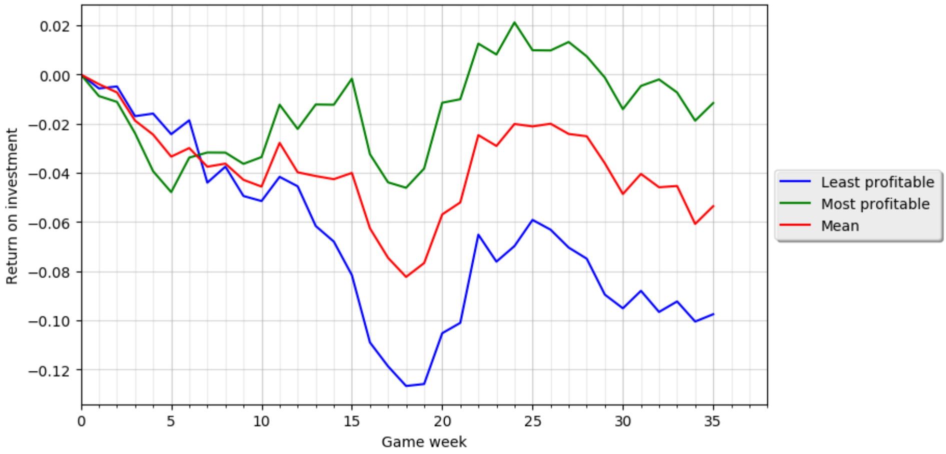


Figure 6.2.9: ROI over the span of the English Premier League season 2016-2017 using the player ratings network and the variance adjusted strategy.

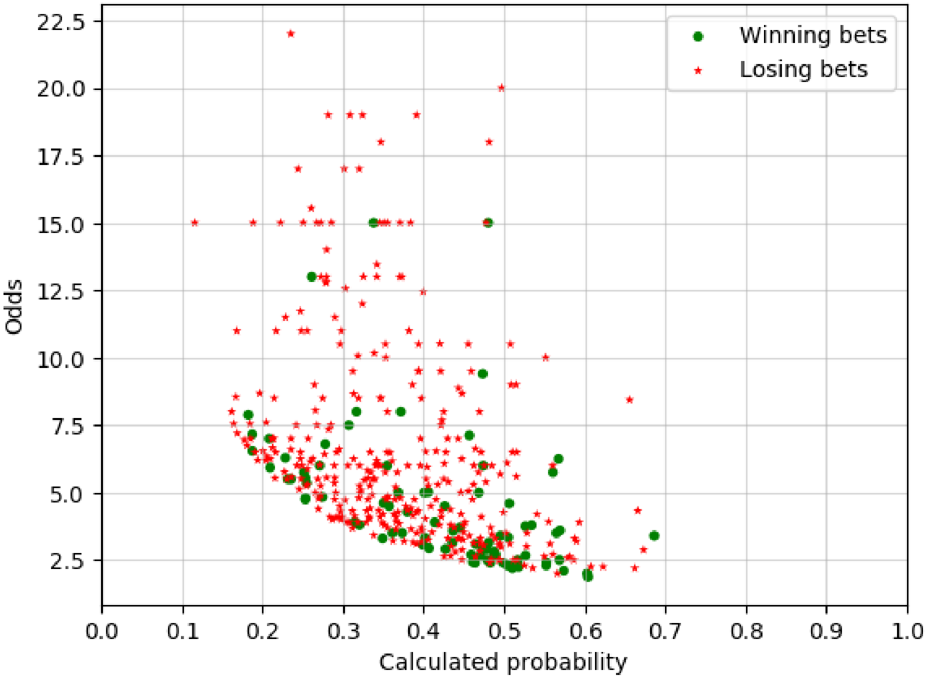


Figure 6.2.10: Offered odds and predicted probabilities for the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by the player ratings network.

Summary

The network did not achieve consistent good results. The first season, the Kelly ratio strategy was the only strategy to generate a profit. The second season, however, the strategy went bankrupt.

Figures 6.2.5 and 6.2.10 show the connection between odds and probabilities predicted by the player ratings network. The player ratings network tend to overestimate the probability of too many high-odds outcomes. A great portion of the bets in the lower half of the horizontal axis are overestimated. The overestimates contribute a lot to the lack of profitability of the network. Over the two seasons, the prediction models won approximately 20.4% of all bets placed, with an average odds of 4.34.

6.3 Head to head

6.3.1 Network structure

Table 6.3.1 shows the RPS values and accuracy of the head to head network. In the evaluation data set, 38.1% of all matches ended in a home victory.

Hidden layer		**RPS values**			
Activation	**Size**	**Min**	**Max**	**Mean**	**Accuracy**
ReLU	16	0.127	0.479	0.300	0.387
ReLU	32	0.0733	0.571	0.234	0.391
ReLU	64	0.0889	0.548	0.232	0.397
ReLU	128	0.0692	0.586	0.238	0.387
Sigmoid	16	0.0823	0.553	0.236	0.394
Sigmoid	32	0.111	0.468	0.228	0.401
Sigmoid	64	0.081	0.557	0.232	0.381
Tanh	16	0.0904	0.537	0.232	0.397
Tanh	32	0.0883	0.537	0.231	0.384
Tanh	64	0.0959	0.523	0.230	0.384

Table 6.3.1: Accuracy of the head to head network, with different hidden layer configurations. The row colored green shows the configuration with most promising results.

Using a hidden layer with 32 nodes activated by the sigmoid function yielded the most promising results, and will therefore be used when evaluating the profitability of the network. Table 6.3.2 shows the RPS values and prediction accuracy when evaluating the same configuration over the 2016-2017 season. Unfortunately, the network did not perform better than the benchmark model for any of the seasons.

RPS values			
Min	**Max**	**Mean**	**Accuracy**
0.0933	0.727	0.220	0.497

Table 6.3.2: Prediction accuracy of the head to head network for the 2016-2017 season of the English Premier League, using the most promising hidden layer configuration.

6.3.2 Betting results

English Premier League 2015-2016

Figures 6.3.1 to 6.3.4 show the development of the ROI generated by the head to head network over the English Premier League season 2015-2016.

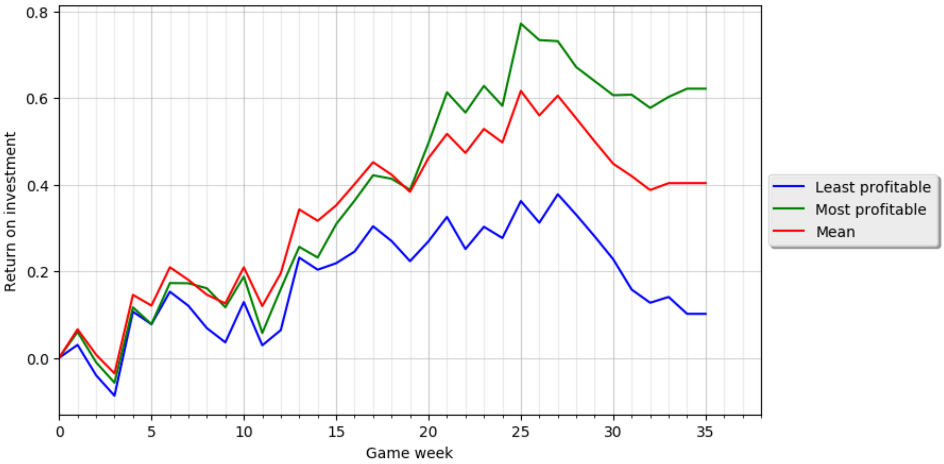


Figure 6.3.1: ROI over the span of the English Premier League season 2015-2016 using the head to head network and the fixed bet strategy.

Table 6.3.3 shows a summary of the ROI values achieved by the different strategies when used by the head to head network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

Figure 6.3.5 shows the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by a random instance of the head to head network.

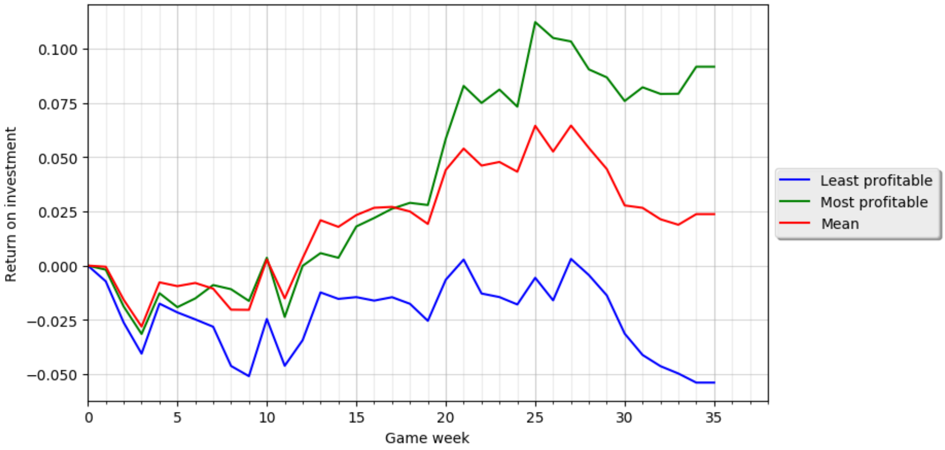


Figure 6.3.2: ROI over the span of the English Premier League season 2015-2016 using the head to head network and the fixed return strategy.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	0.16	0.61	0.40
Fixed return	-0.053	0.092	0.025
Kelly ratio	0.44	3.1	2.0
Variance adjusted	-0.070	0.057	-0.0080

Table 6.3.3: Final ROI values for the four strategies when using the head to head network during the 2015-2016 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

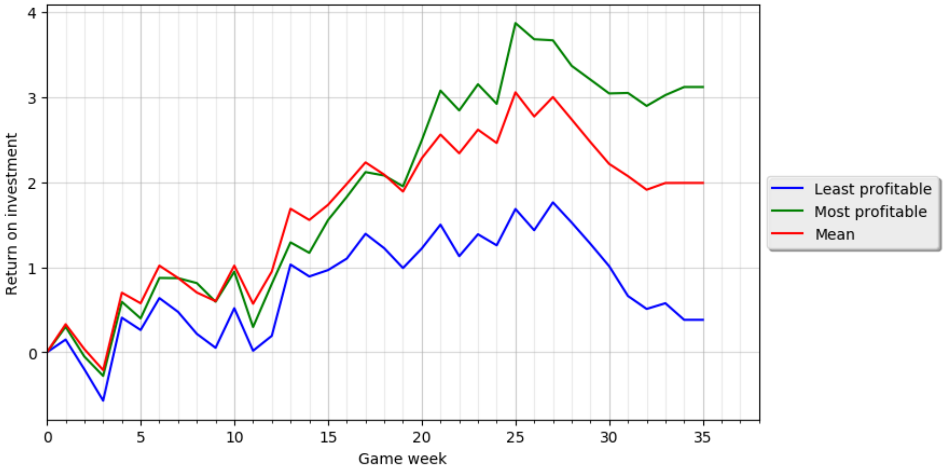


Figure 6.3.3: ROI over the span of the English Premier League season 2015-2016 using the head to head network and the Kelly ratio strategy.

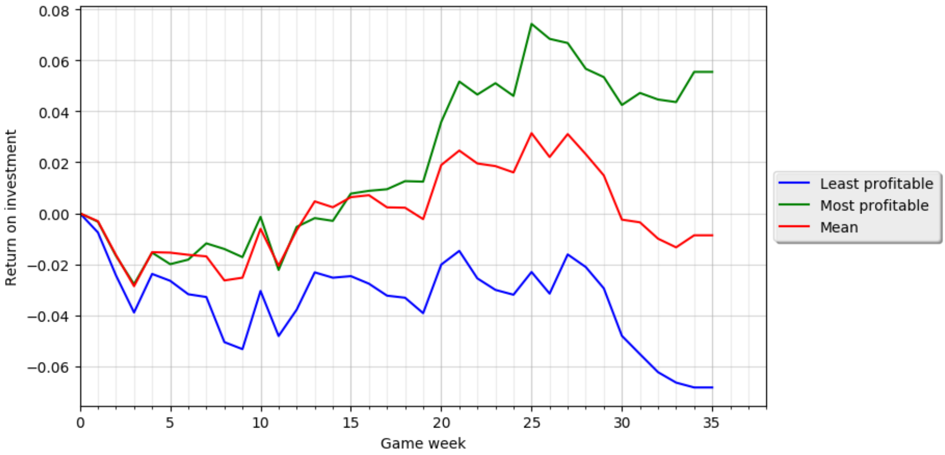


Figure 6.3.4: ROI over the span of the English Premier League season 2015-2016 using the head to head network and the variance adjusted strategy.

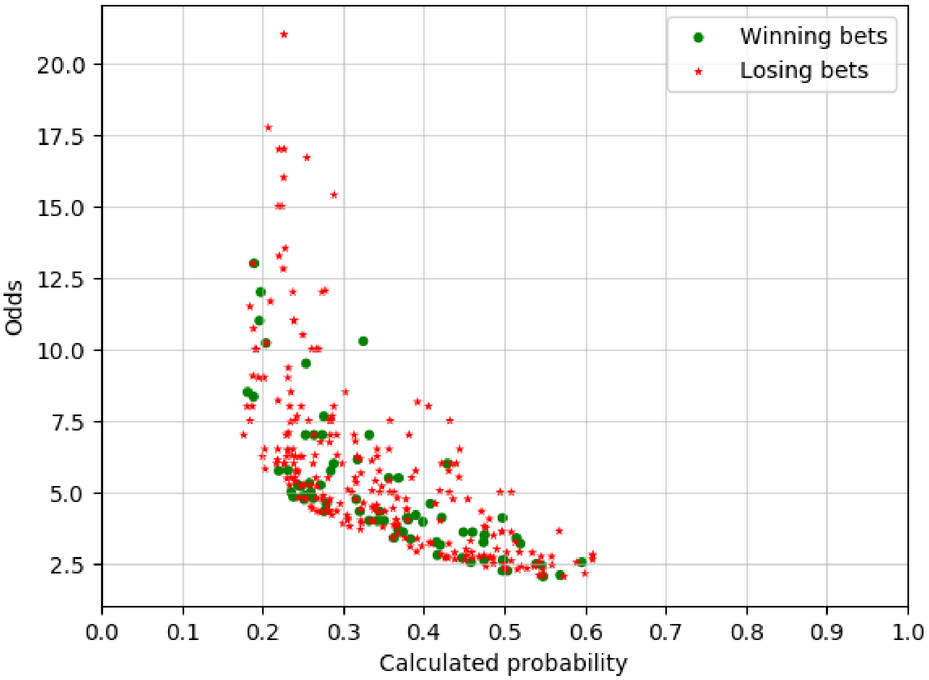


Figure 6.3.5: Offered odds and predicted probabilities for the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by the head to head network.

English Premier League 2016-2017

Figures 6.3.6 to 6.3.9 show the development of the ROI generated by the head to head network over the English Premier League season 2016-2017.

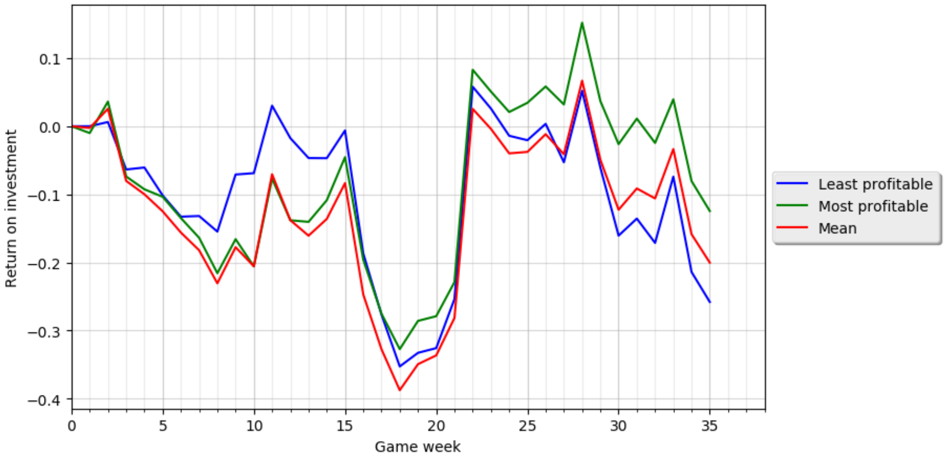


Figure 6.3.6: ROI over the span of the English Premier League season 2016-2017 using the head to head network and the fixed bet strategy.

Table 6.3.4 shows a summary of the ROI values achieved by the different strategies when used by the head to head network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.25	-0.12	-0.20
Fixed return	-0.038	0.07	-0.018
Kelly ratio	-1.0	-1.0	-1.0
Variance adjusted	-0.024	0.014	-0.0060

Table 6.3.4: Final ROI values for the four strategies when using the head to head network during the 2016-2017 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

Figure 6.3.10 shows the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by a random instance of the head to head network.

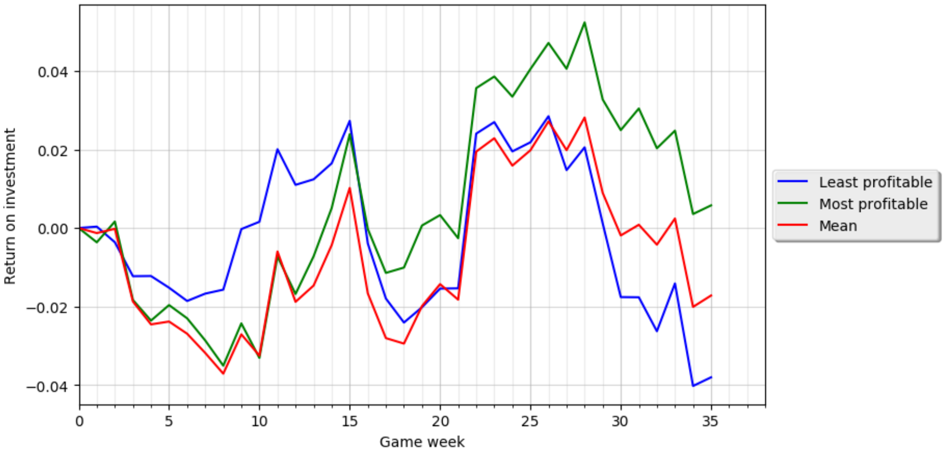


Figure 6.3.7: ROI over the span of the English Premier League season 2016-2017 using the head to head network and the fixed return strategy.

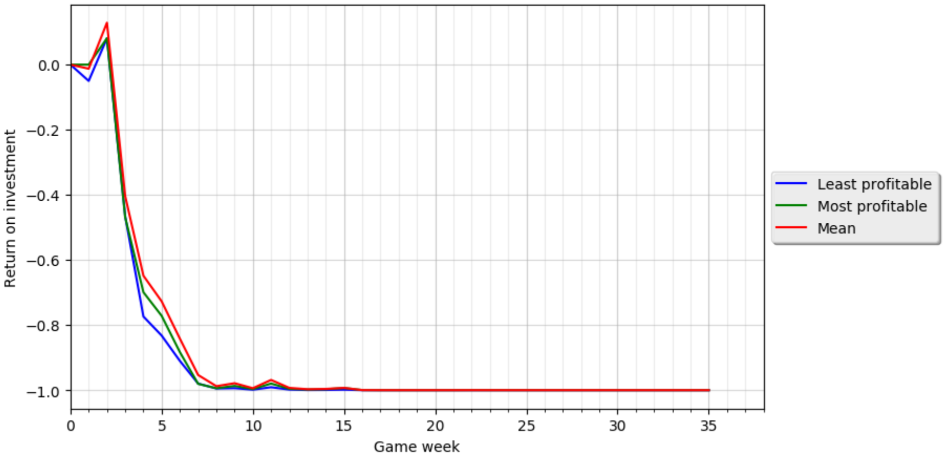


Figure 6.3.8: ROI over the span of the English Premier League season 2016-2017 using the head to head network and the Kelly ratio strategy.

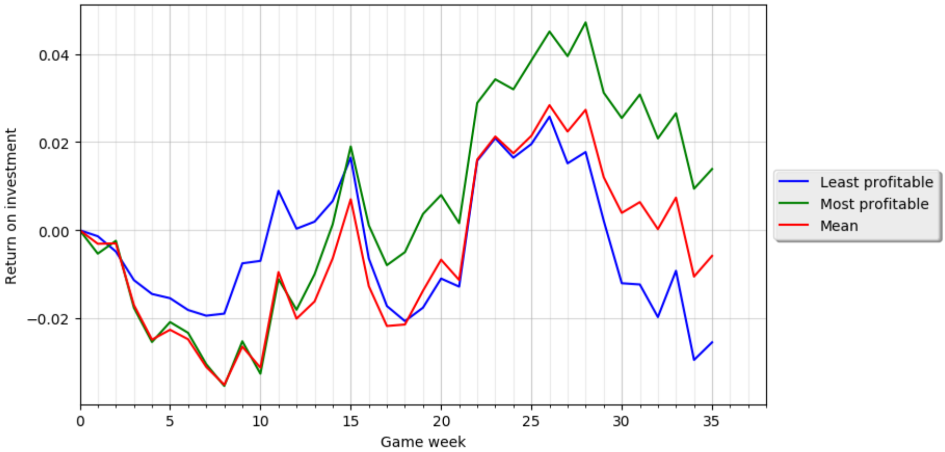


Figure 6.3.9: ROI over the span of the English Premier League season 2016-2017 using the head to head network and the variance adjusted strategy.

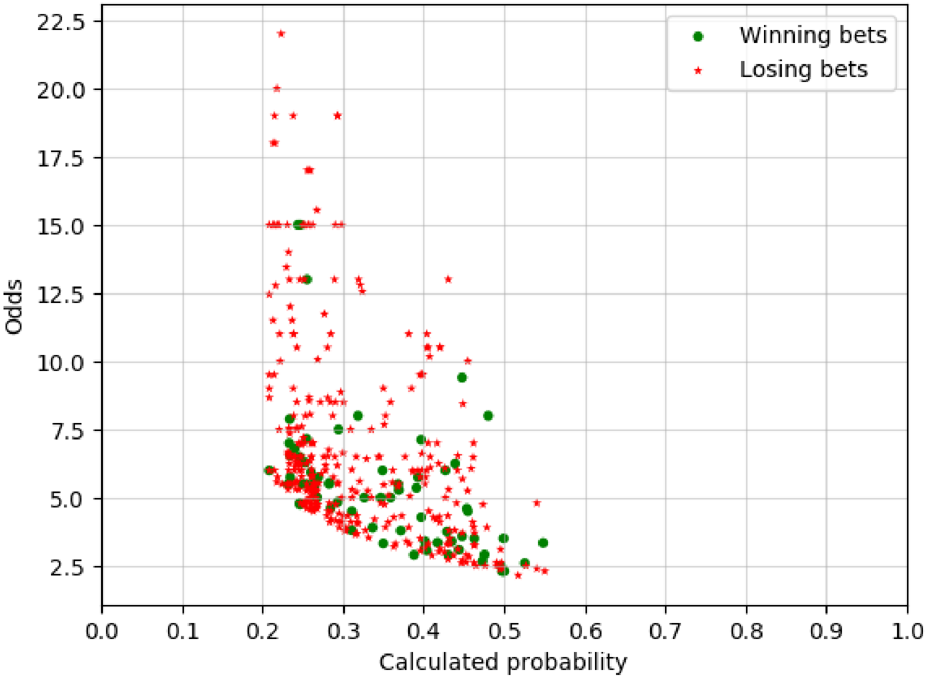


Figure 6.3.10: Offered odds and predicted probabilities for the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by the head to head network.

Summary

The network did not achieve consistent good results. The first season, the fixed bet and Kelly ratio strategies performed well, gaining profits for all instances. The second season, however, the same strategies achieved ROIs of -0.25 and -1.0, respectively.

Figures 6.3.5 and 6.3.10 show the connection between odds and probabilities predicted by the head to head network. The head to head network struggles with sparse predictions. This is likely due to how rapidly teams change in football. Between each season, and in January, teams have the opportunity to buy and sell players. A team being relegated from the English Premier League might be forced to sell their best players, reducing their strengths. Other teams receive a lot of money from their owners, making them able to improve their squad greatly between two seasons. This makes it difficult to predict the results of a match based on the previous meetings between the two teams. The training procedure therefore approximates the outcome distribution in the training data set. Over the two seasons, the prediction models won approximately 19.2% of all bets placed, with an average odds of 4.89.

6.4 Previous meetings

6.4.1 Network structure

Table 6.4.1 shows the RPS values and accuracy of the previous meetings network. In the evaluation data set, 38.1% of all matches ended in a home victory.

Hidden layer		**RPS values**			
Activation	**Size**	**Min**	**Max**	**Mean**	**Accuracy**
ReLU	32	0.0525	0.585	0.235	0.374
ReLU	64	0.0425	0.616	0.236	0.381
ReLU	128	0.0505	0.616	0.242	0.358
Sigmoid	32	0.125	0.457	0.228	0.374
Sigmoid	64	0.138	0.436	0.236	0.381
Sigmoid	128	0.133	0.371	0.228	0.381
Sigmoid	256	0.112	0.484	0.230	0.381
Tanh	32	0.0342	0.650	0.249	0.354
Tanh	64	0.0573	0.619	0.237	0.361
Tanh	128	0.0352	0.646	0.236	0.394
Tanh	256	0.0340	0.671	0.248	0.381

Table 6.4.1: Accuracy of the previous meetings network, with different hidden layer configurations. The row colored green shows the configuration with most promising results.

Using a hidden layer with 128 nodes activated by the tanh function yielded the most promising results, and will therefore be used when evaluating the profitability of the network. Table 6.4.2 shows the RPS values and prediction accuracy when evaluating the same configuration over the 2016-2017 season. Unfortunately, the network did not perform better than the benchmark model for any of the seasons.

RPS values			
Min	**Max**	**Mean**	**Accuracy**
0.0807	0.657	0.217	0.503

Table 6.4.2: Prediction accuracy of the previous meetings network for the 2016-2017 season of the English Premier League, using the most promising hidden layer configuration.

6.4.2 Betting results

English Premier League 2015-2016

Figures 6.4.1 to 6.4.4 show the development of the ROI generated by the previous meetings network over the English Premier League season 2015-2016.

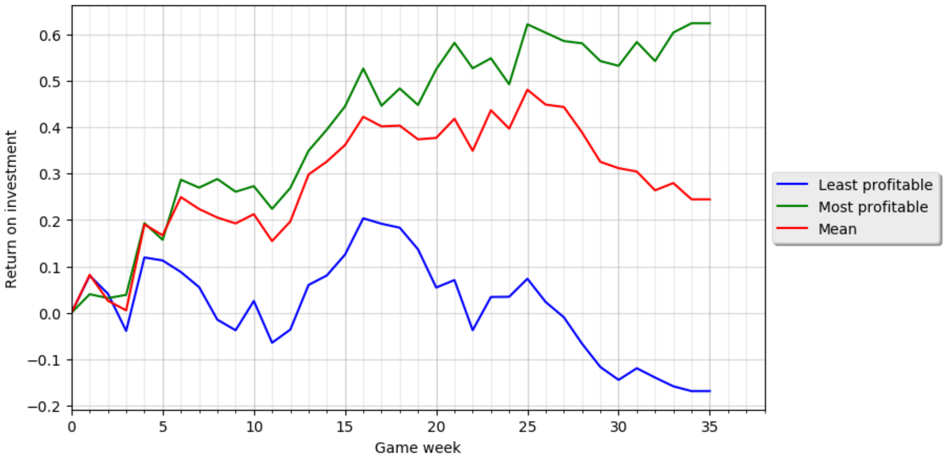


Figure 6.4.1: ROI over the span of the English Premier League season 2015-2016 using the previous meetings network and the fixed bet strategy.

Table 6.4.3 shows a summary of the ROI values achieved by the different strategies when used by the previous meetings network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

Figure 6.4.5 shows the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by a random instance of the previous meetings network.

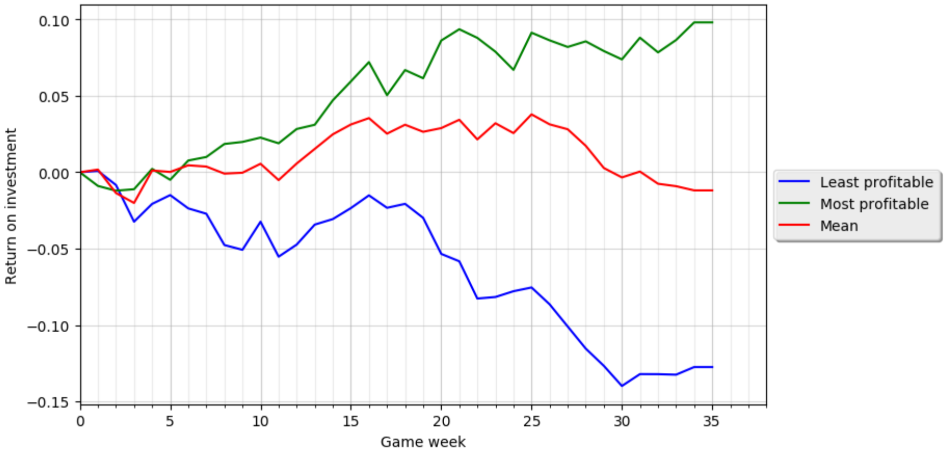


Figure 6.4.2: ROI over the span of the English Premier League season 2015-2016 using the previous meetings network and the fixed return strategy.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.17	0.62	0.25
Fixed return	-0.13	0.099	-0.012
Kelly ratio	-1.0	3.1	1.2
Variance adjusted	-0.17	0.075	-0.052

Table 6.4.3: Final ROI values for the four strategies when using the previous meetings network during the 2015-2016 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

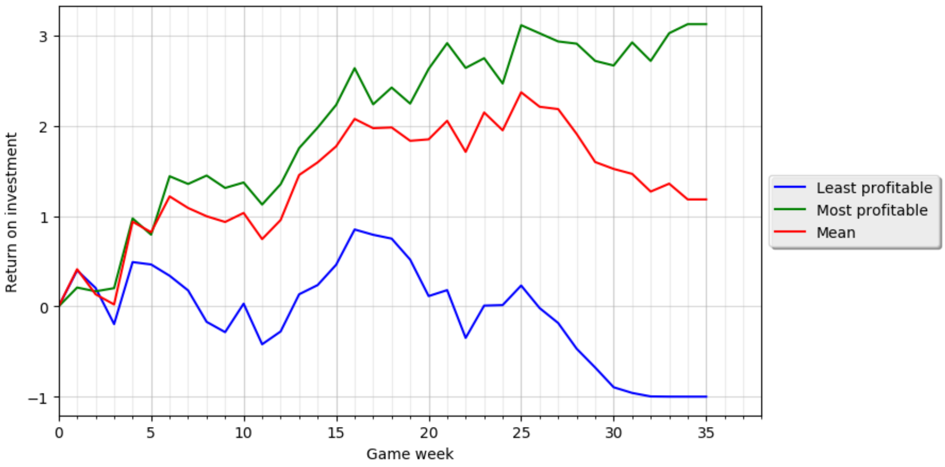


Figure 6.4.3: ROI over the span of the English Premier League season 2015-2016 using the previous meetings network and the Kelly ratio strategy.

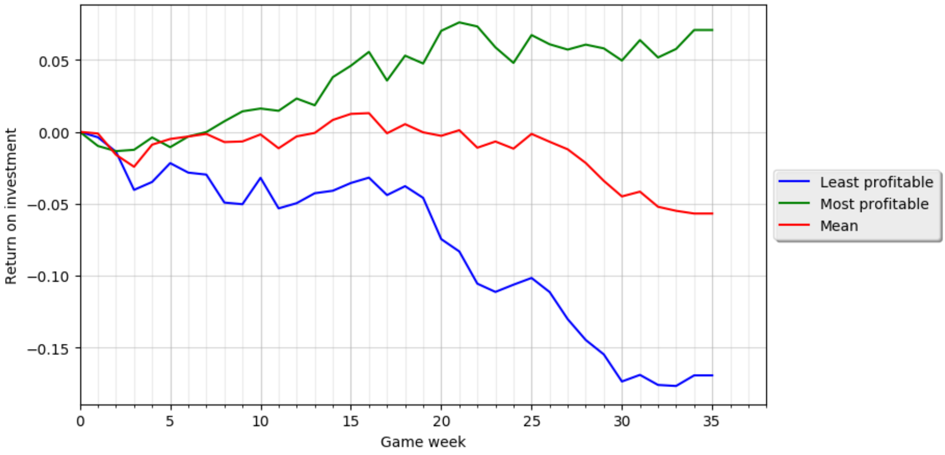


Figure 6.4.4: ROI over the span of the English Premier League season 2015-2016 using the previous meetings network and the variance adjusted strategy.

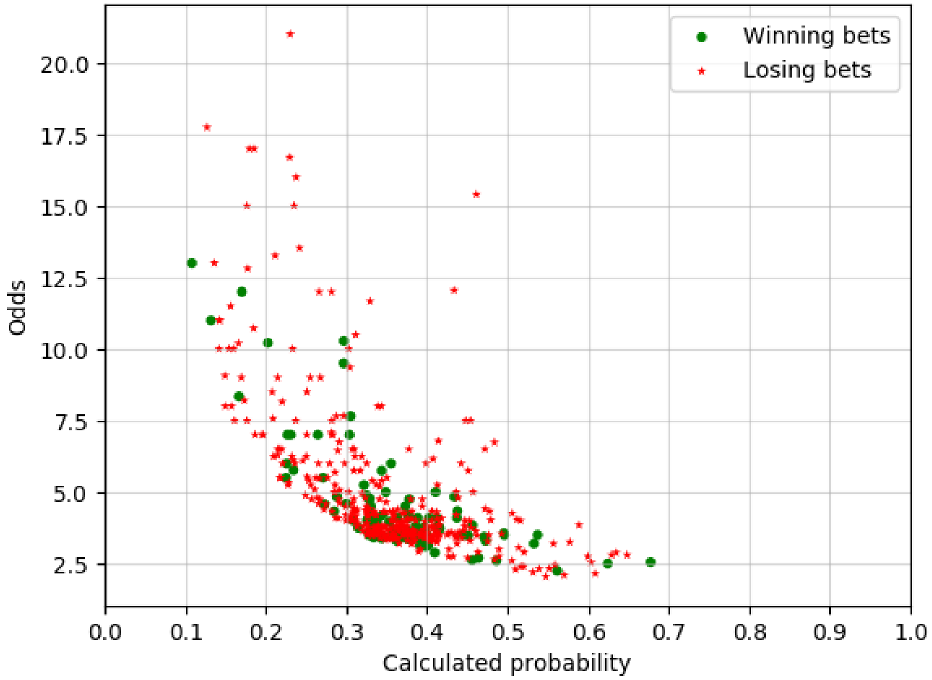


Figure 6.4.5: Offered odds and predicted probabilities for the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by the previous meetings network.

English Premier League 2016-2017

Figures 6.4.6 to 6.4.9 show the development of the ROI generated by the previous meetings network over the English Premier League season 2016-2017.

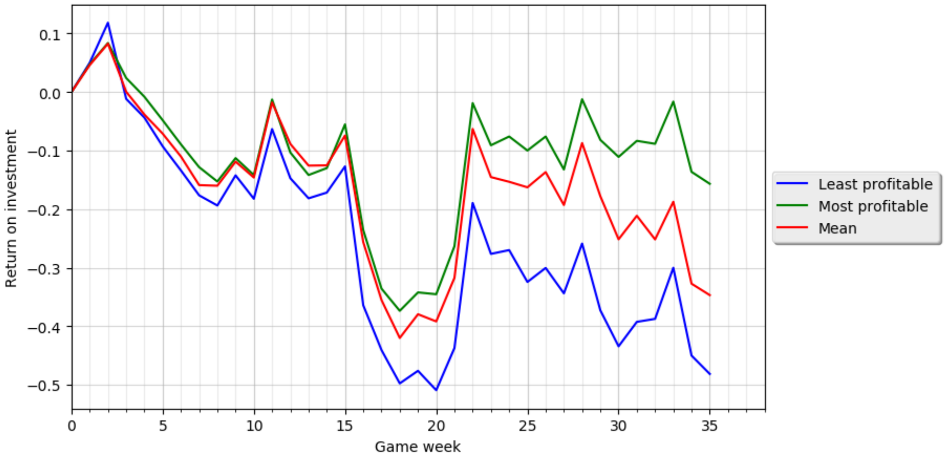


Figure 6.4.6: ROI over the span of the English Premier League season 2016-2017 using the previous meetings network and the fixed bet strategy.

Table 6.4.4 shows a summary of the ROI values achieved by the different strategies when used by the previous meetings network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.48	-0.15	-0.34
Fixed return	-0.09	0.0055	-0.050
Kelly ratio	-1.0	-1.0	-1.0
Variance adjusted	-0.058	0.027	-0.022

Table 6.4.4: Final ROI values for the four strategies when using the previous meetings network during the 2016-2017 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

Figure 6.4.10 shows the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by a random instance of the previous meetings network.

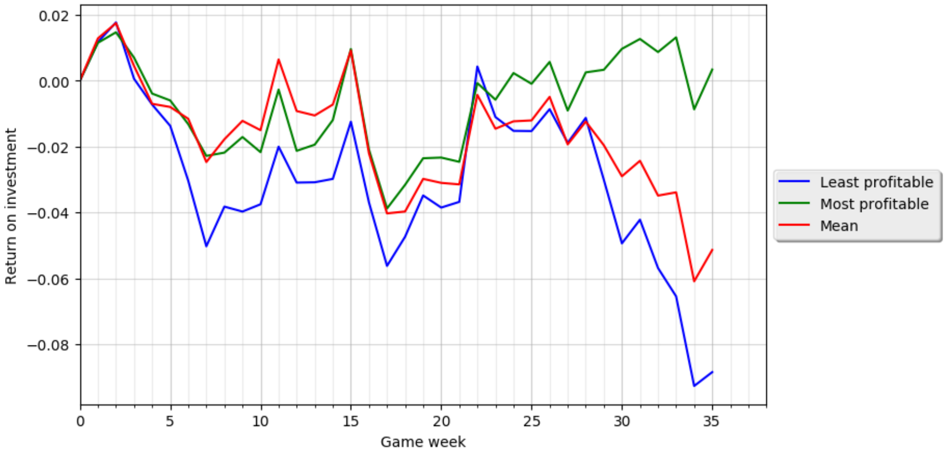


Figure 6.4.7: ROI over the span of the English Premier League season 2016-2017 using the previous meetings network and the fixed return strategy.

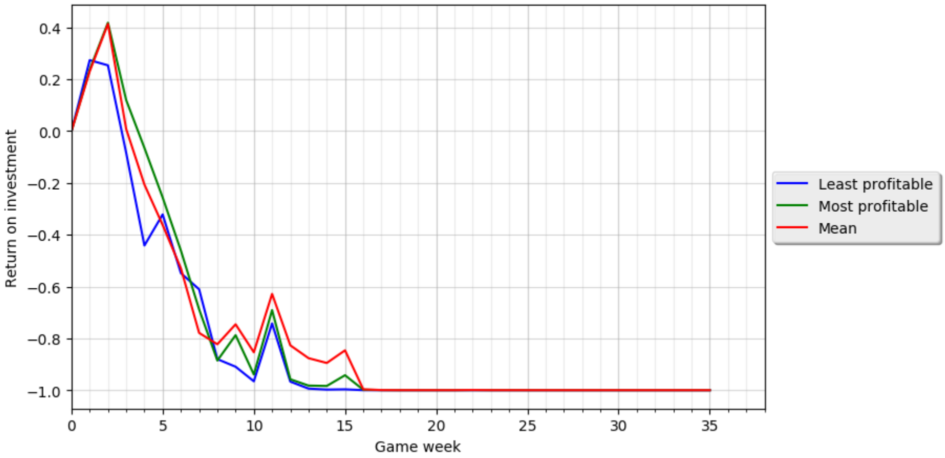


Figure 6.4.8: ROI over the span of the English Premier League season 2016-2017 using the previous meetings network and the Kelly ratio strategy.

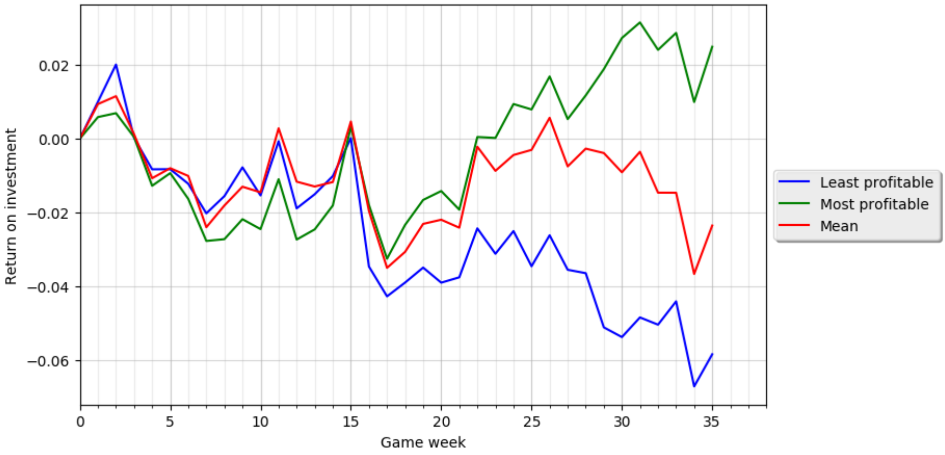


Figure 6.4.9: ROI over the span of the English Premier League season 2016-2017 using the previous meetings network and the variance adjusted strategy.

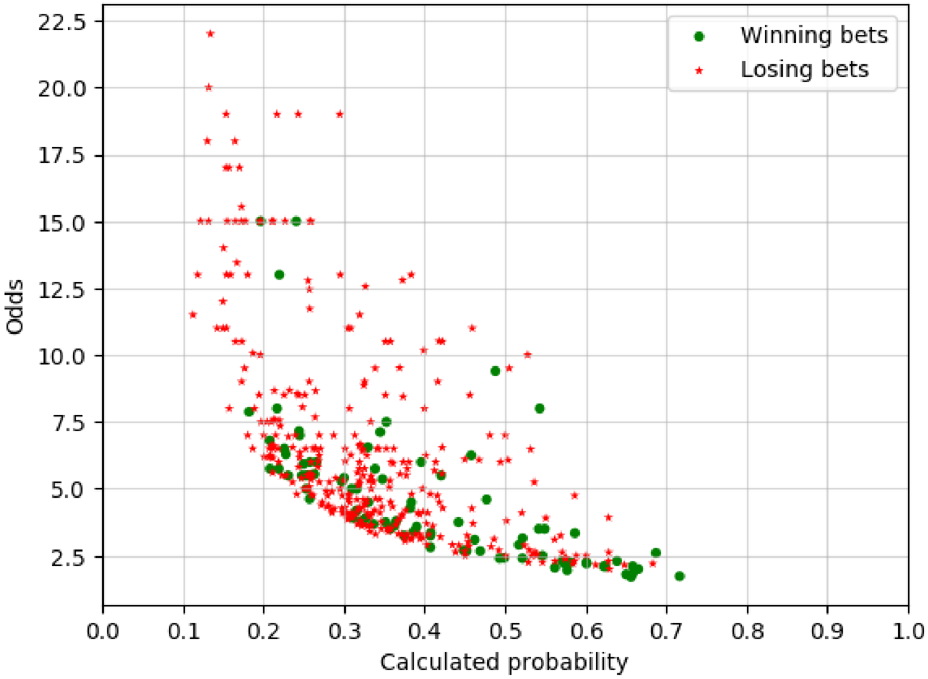


Figure 6.4.10: Offered odds and predicted probabilities for the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by the previous meetings network.

Summary

Similarly to the head to head network, the previous meetings network did not achieve consistent good results. The first season, the fixed bet and Kelly ratio strategies performed well, gaining a profit on average. The second season, however, the same strategies achieved ROIs of -0.34 and -1.0, respectively.

Figures 6.4.5 and 6.4.10 show the connection between odds and probabilities predicted by the previous meetings network. The previous meetings network has a more even spread than the head to head network. However, the previous meetings network tend to overestimate the probabilities across the board. The reason for the low spread is probably the same as for the head to head network. However, the previous meetings network also utilizes team ratings, accounting for potential team changes between the seasons. Over the two seasons, the prediction models won approximately 20.3% of all bets placed, with an average odds of 4.70.

6.5 Team characteristics

6.5.1 Network structure

Table 6.5.1 shows the RPS values and accuracy of the team characteristics network. In the evaluation data set, 41.3% of all matches ended in a home victory.

Hidden layer		**RPS values**			
Activation	**Size**	**Min**	**Max**	**Mean**	**Accuracy**
ReLU	32	0.0143	0.794	0.226	0.463
ReLU	64	0.00102	0.785	0.218	0.466
ReLU	128	0.0122	0.719	0.224	0.458
Sigmoid	32	0.0130	0.737	0.217	0.460
Sigmoid	64	0.0245	0.665	0.217	0.468
Sigmoid	128*	0.0169	0.706	0.213	0.469
Sigmoid	256*	0.0103	0.744	0.215	0.450
Tanh	32*	0.0152	0.775	0.223	0.445
Tanh	64*	0.00779	0.800	0.220	0.461
Tanh	128**	0.0102	0.698	0.216	0.463
Tanh	256**	0.0108	0.823	0.224	0.461

Table 6.5.1: Accuracy of the team characteristics network, with different hidden layer configurations. The row colored green shows the configuration with most promising results.

Using a hidden layer with 128 nodes activated by the sigmoid function yielded the most promising results, and will therefore be used when evaluating the profitability of the team characteristics network. Table 6.5.2 shows the RPS values and prediction accuracy when evaluating the same configuration over the 2016-2017 season. The network achieved RPS values similar to the benchmark model the first season. The second season, the network achieved slightly better values.

RPS values			
Min	**Max**	**Mean**	**Accuracy**
0.0171	0.700	0.187	0.609

Table 6.5.2: Prediction accuracy of the team characteristics network for the 2016-2017 season of the English Premier League, using the most promising hidden layer configuration.

6.5.2 Betting results

English Premier League 2015-2016

Figures 6.5.1 to 6.5.4 show the development of the ROI generated by the team characteristics network over the English Premier League season 2015-2016.

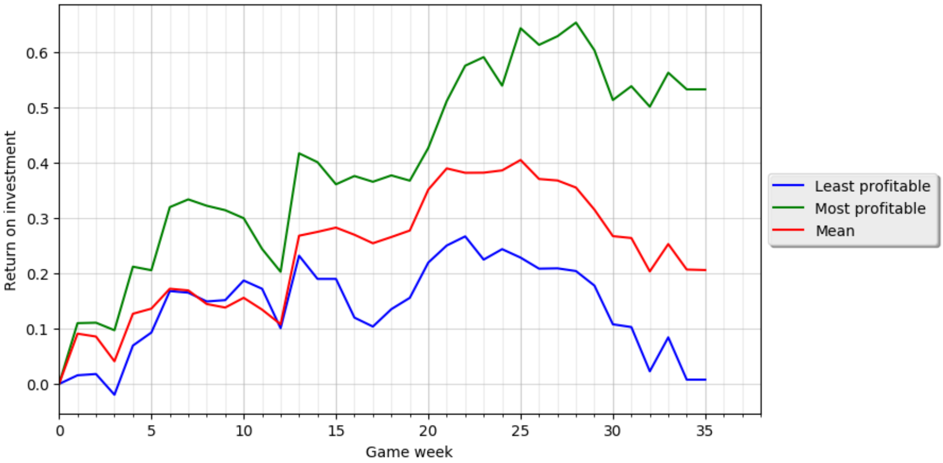


Figure 6.5.1: ROI over the span of the English Premier League season 2015-2016 using the team characteristics network and the fixed bet strategy.

Table 6.5.3 shows a summary of the ROI values achieved by the different strategies when used by the team characteristics network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

Figure 6.5.5 shows the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by a random instance of the team characteristics network.

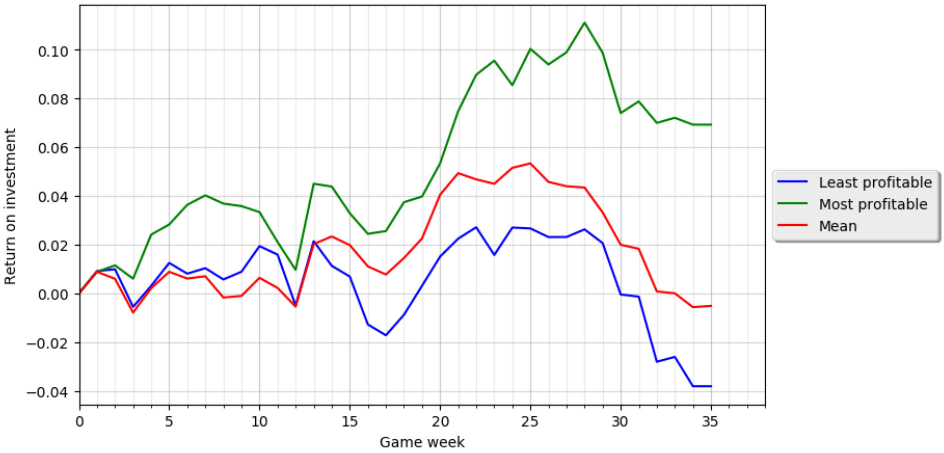


Figure 6.5.2: ROI over the span of the English Premier League season 2015-2016 using the team characteristics network and the fixed return strategy.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	0.0010	0.54	0.21
Fixed return	-0.038	0.070	-0.050
Kelly ratio	-0.25	2.6	0.80
Variance adjusted	-0.07	0.039	-0.032

Table 6.5.3: Final ROI values for the four strategies when using the team characteristics network during the 2015-2016 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

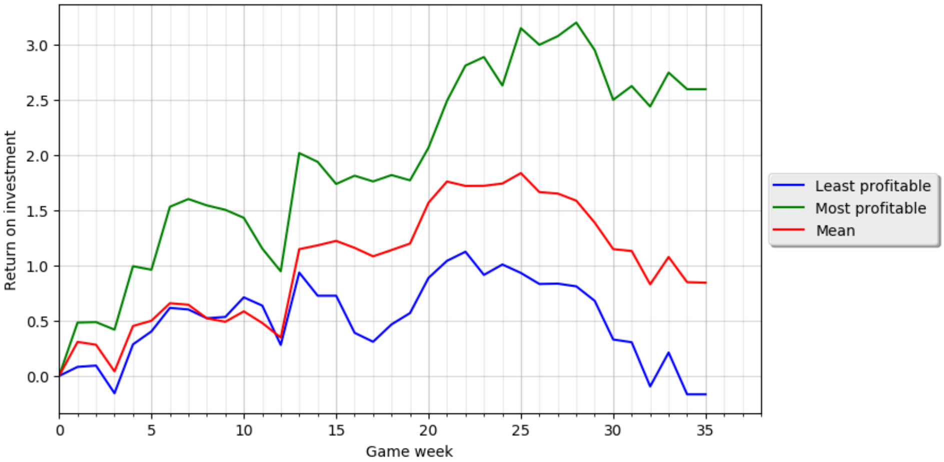


Figure 6.5.3: ROI over the span of the English Premier League season 2015-2016 using the team characteristics network and the Kelly ratio strategy.

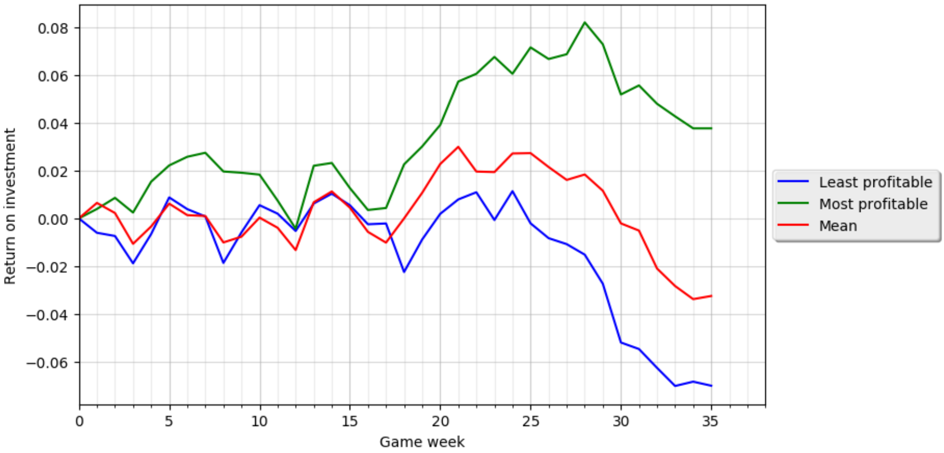


Figure 6.5.4: ROI over the span of the English Premier League season 2015-2016 using the team characteristics network and the variance adjusted strategy.

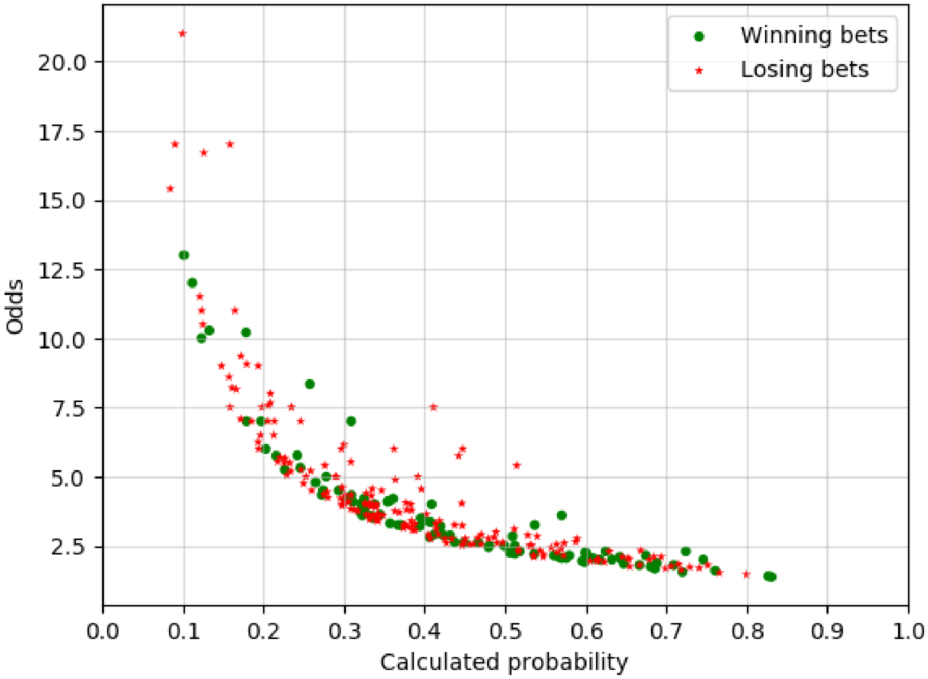


Figure 6.5.5: Offered odds and predicted probabilities for the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by the team characteristics network.

English Premier League 2016-2017

Figures 6.5.6 to 6.5.9 show the development of the ROI generated by the team characteristics network over the English Premier League season 2016-2017.

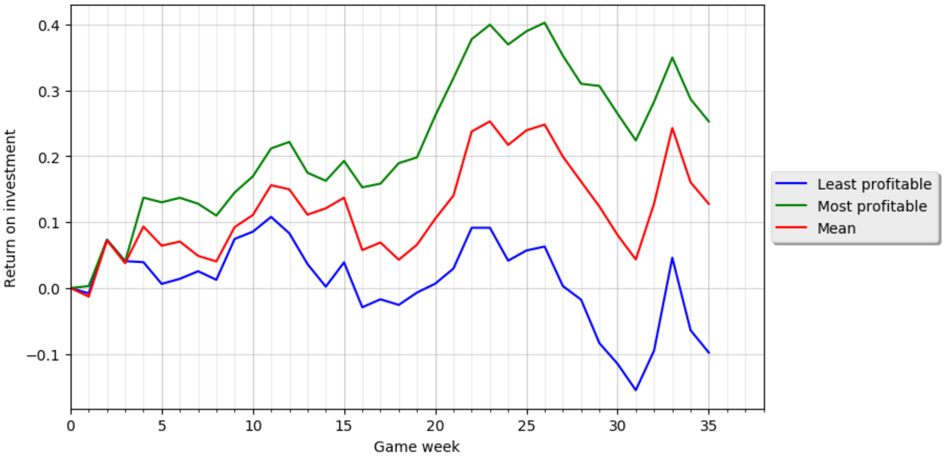


Figure 6.5.6: ROI over the span of the English Premier League season 2016-2017 using the team characteristics network and the fixed bet strategy.

Table 6.5.4 shows a summary of the ROI values achieved by the different strategies when used by the team characteristics network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.10	0.27	0.13
Fixed return	-0.010	0.14	0.080
Kelly ratio	-0.48	1.3	0.70
Variance adjusted	0.041	0.21	0.12

Table 6.5.4: Final ROI values for the four strategies when using the team characteristics network during the 2016-2017 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

Figure 6.5.10 shows the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by a random instance of the team characteristics network.

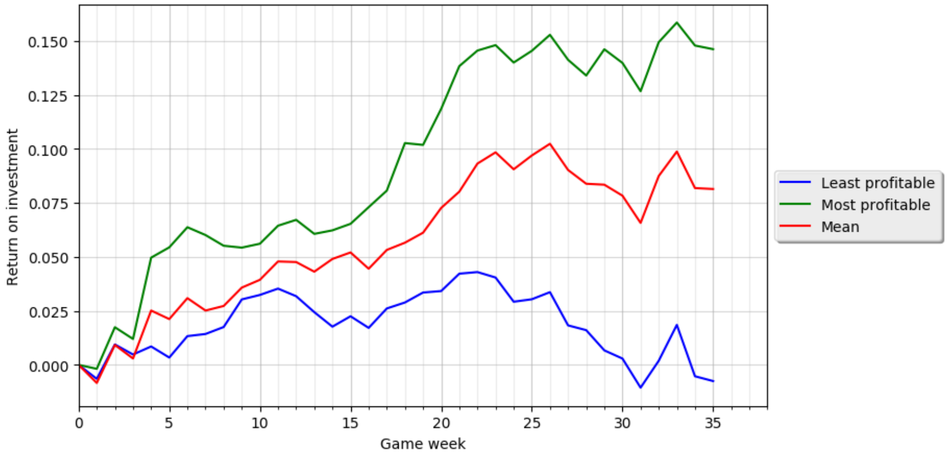


Figure 6.5.7: ROI over the span of the English Premier League season 2016-2017 using the team characteristics network and the fixed return strategy.

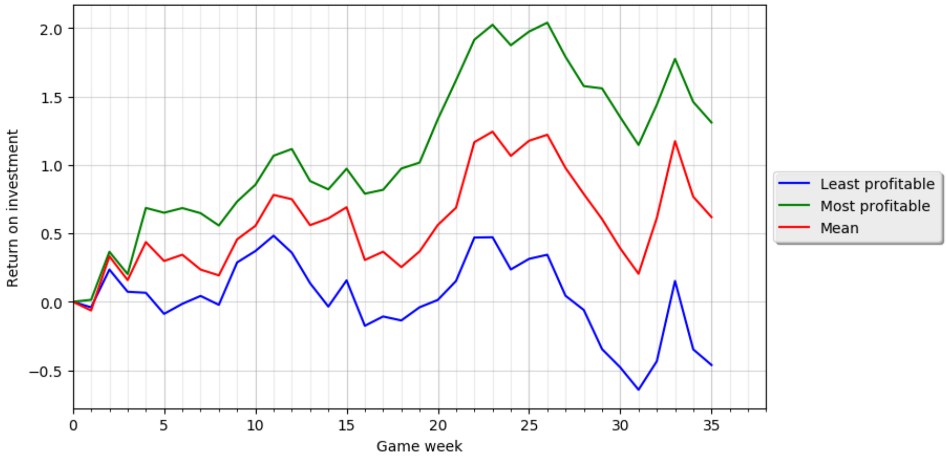


Figure 6.5.8: ROI over the span of the English Premier League season 2016-2017 using the team characteristics network and the Kelly ratio strategy.

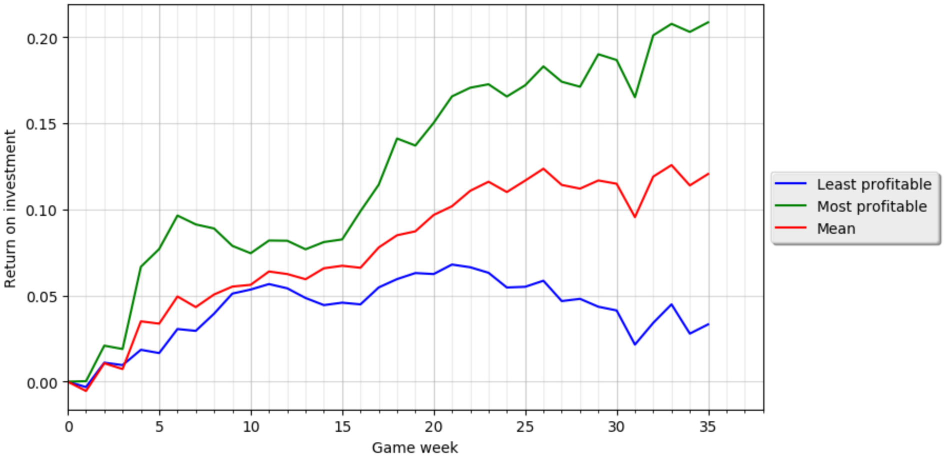


Figure 6.5.9: ROI over the span of the English Premier League season 2016-2017 using the team characteristics network and the variance adjusted strategy.

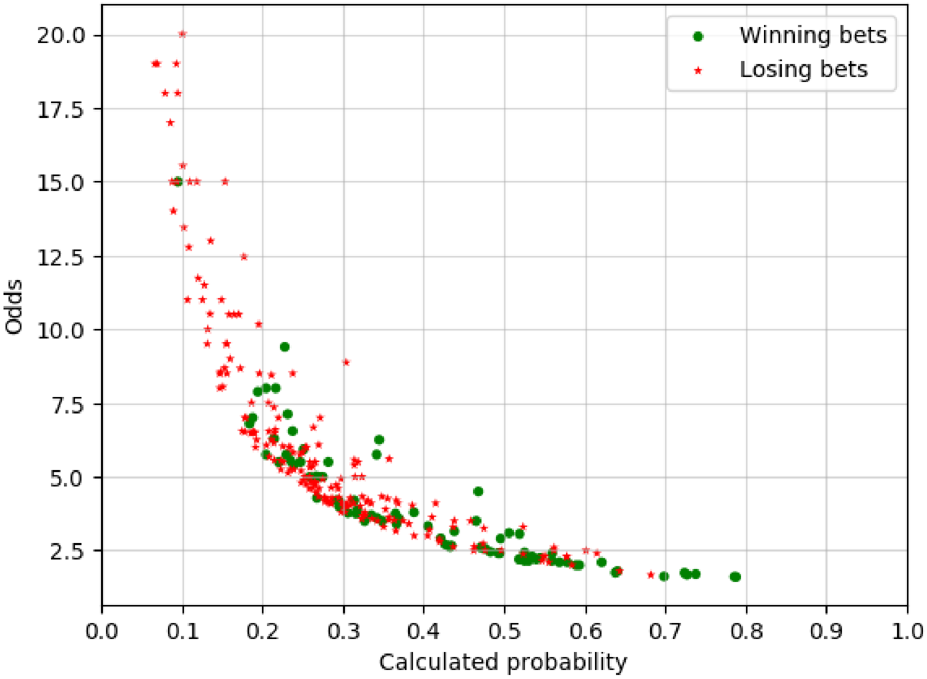


Figure 6.5.10: Offered odds and predicted probabilities for the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by the team characteristics network.

Summary

The team characteristics network achieved consistent good results, generating profits over both seasons using the fixed bet and Kelly ratio strategies. The network also generated profits the other strategies the first season.

Figures 6.5.5 and 6.5.10 show the connection between odds and probabilities predicted by the team characteristics network. The team characteristics network produce far better predictions than any other network. There is, however, a slight tendency to overestimate the probabilities in the lower fifth of the horizontal axis. Over the two seasons, the prediction models won approximately 29.4% of all bets placed, with an average odds of 3.82.

6.6 Team characteristics and strengths

6.6.1 Network structure

Table 6.6.1 shows the RPS values and accuracy of the team characteristics and strengths network. In the evaluation data set, 41.5% of all matches ended in a home victory.

Hidden layer		**RPS values**			
Activation	**Size**	**Min**	**Max**	**Mean**	**Accuracy**
ReLU	32	0.0124	0.761	0.219	0.434
ReLU	64*	0.00991	0.707	0.215	0.463
ReLU	128*	0.0113	0.728	0.216	0.459
Sigmoid	32	0.0275	0.663	0.214	0.441
Sigmoid	64	0.0243	0.711	0.214	0.460
Sigmoid	128	0.0219	0.714	0.214	0.463
Sigmoid	256*	0.0146	0.716	0.213	0.473
Sigmoid	512*	0.0182	0.689	0.214	0.457
Tanh	32	0.00974	0.738	0.219	0.465
Tanh	64*	0.00807	0.770	0.216	0.465
Tanh	128*	0.0129	0.764	0.215	0.457

Table 6.6.1: Accuracy of the team characteristics and strengths network, with different hidden layer configurations. The row colored green shows the configuration with most promising results.

Using a hidden layer with 256 nodes activated by the sigmoid function yielded the most promising results, and will therefore be used when evaluating the profitability of the team characteristics and strengths network. Table 6.6.2 shows the RPS values and prediction accuracy when evaluating the same configuration over the 2016-2017 season. The network achieved RPS values similar to the benchmark model the first season. The second season, the network achieved slightly better values.

RPS values			
Min	**Max**	**Mean**	**Accuracy**
0.0139	0.740	0.184	0.591

Table 6.6.2: Prediction accuracy of the team characteristics to strengths network for the 2016-2017 season of the English Premier League, using the most promising hidden layer configuration.

6.6.2 Betting results

English Premier League 2015-2016

Figures 6.6.1 to 6.6.4 show the development of the ROI generated by the team characteristics and strengths network over the English Premier League season 2015-2016.

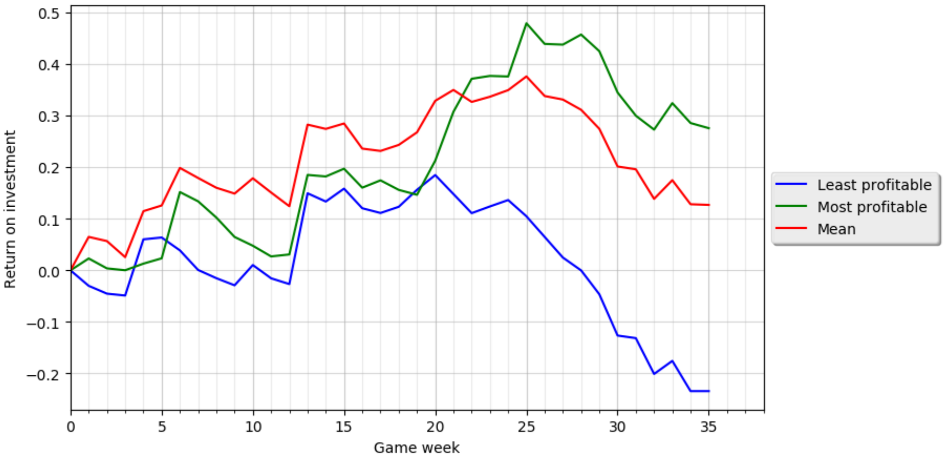


Figure 6.6.1: ROI over the span of the English Premier League season 2015-2016 using the team characteristics and strengths network and the fixed bet strategy.

Table 6.6.3 shows a summary of the ROI values achieved by the different strategies when used by the team characteristics and strengths network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

Figure 6.6.5 shows the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by a random instance of the team characteristics and strengths network.

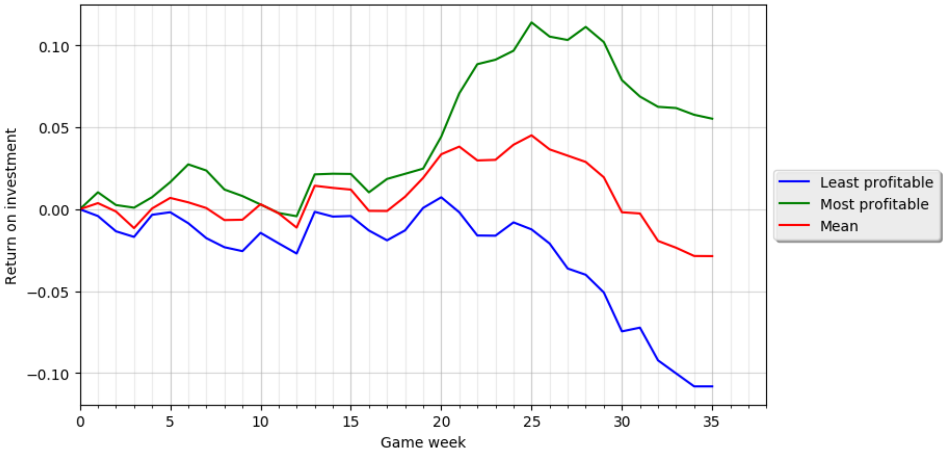


Figure 6.6.2: ROI over the span of the English Premier League season 2015-2016 using the team characteristics and strengths network and the fixed return strategy.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.24	0.28	0.12
Fixed return	-0.11	0.058	-0.025
Kelly ratio	-1.0	1.3	0.49
Variance adjusted	-0.14	0.052	-0.052

Table 6.6.3: Final ROI values for the four strategies when using the team characteristics and strengths network during the 2015-2016 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

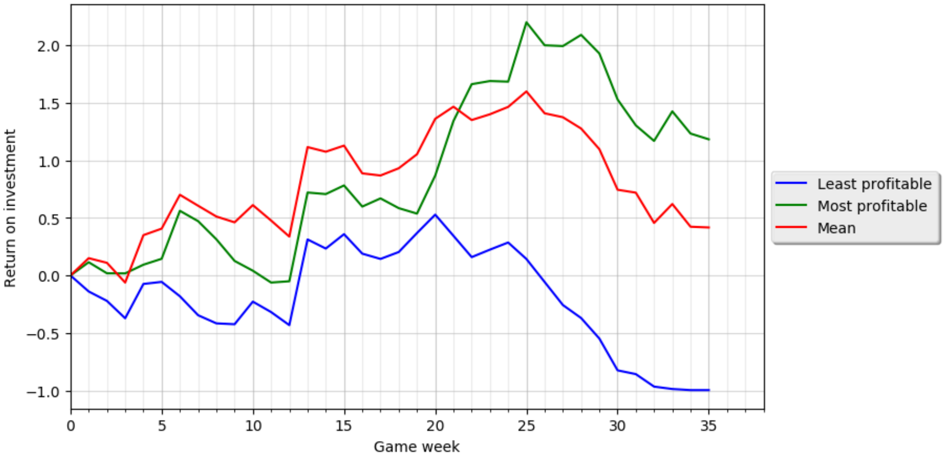


Figure 6.6.3: ROI over the span of the English Premier League season 2015-2016 using the team characteristics and strengths network and the Kelly ratio strategy.

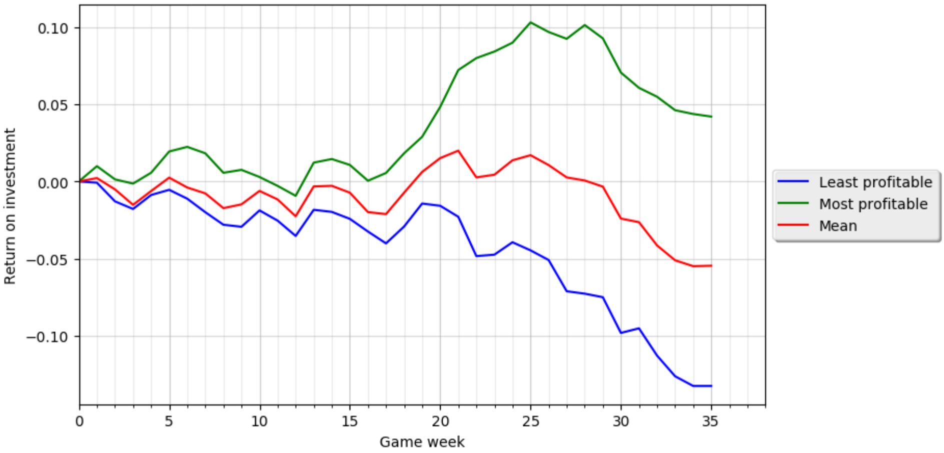


Figure 6.6.4: ROI over the span of the English Premier League season 2015-2016 using the team characteristics and strengths network and the variance adjusted strategy.

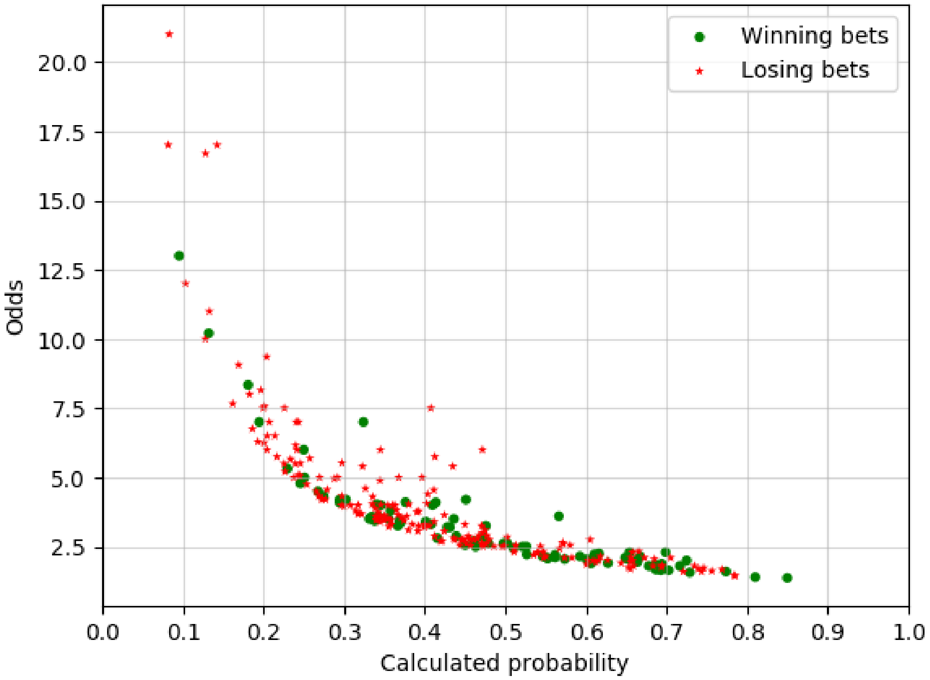


Figure 6.6.5: Offered odds and predicted probabilities for the bets placed during the 2015-2016 season of the English Premier League. The probabilities are generated by the team characteristics and strengths network.

English Premier League 2016-2017

Figures 6.6.6 to 6.6.9 show the development of the ROI generated by the team characteristics and strengths network over the English Premier League season 2016-2017.

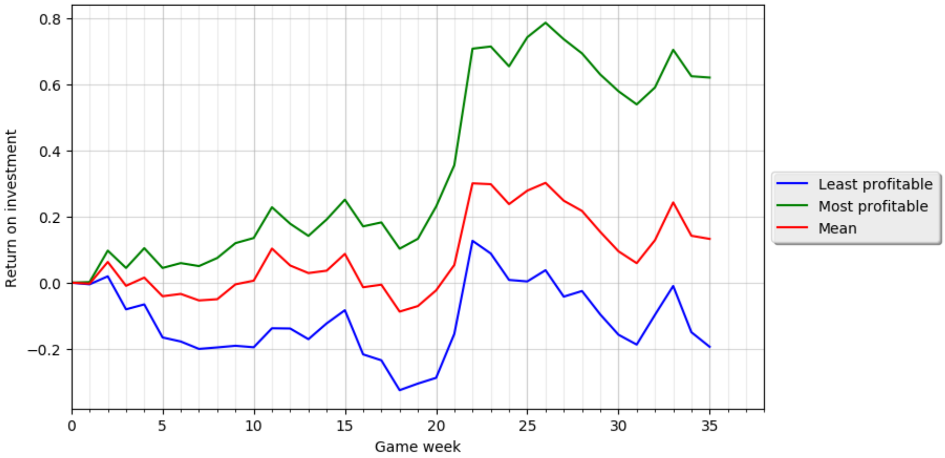


Figure 6.6.6: ROI over the span of the English Premier League season 2016-2017 using the team characteristics and strengths network and the fixed bet strategy.

Table 6.6.4 shows a summary of the ROI values achieved by the different strategies when used by the team characteristics and strengths network. The table shows the final ROI for the least profitable and most profitable simulations, together with the average final ROI.

	Final ROI		
Strategy	**Min**	**Max**	**Mean**
Fixed bet	-0.19	0.62	0.17
Fixed return	0.016	0.20	0.080
Kelly ratio	-0.1	3.1	0.80
Variance adjusted	0.045	0.21	0.13

Table 6.6.4: Final ROI values for the four strategies when using the team characteristics and strengths network during the 2016-2017 season of the English Premier League. The green colored cell was the most profitable strategy (on average).

Figure 6.6.10 shows the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by a random instance of the team characteristics and strengths network.

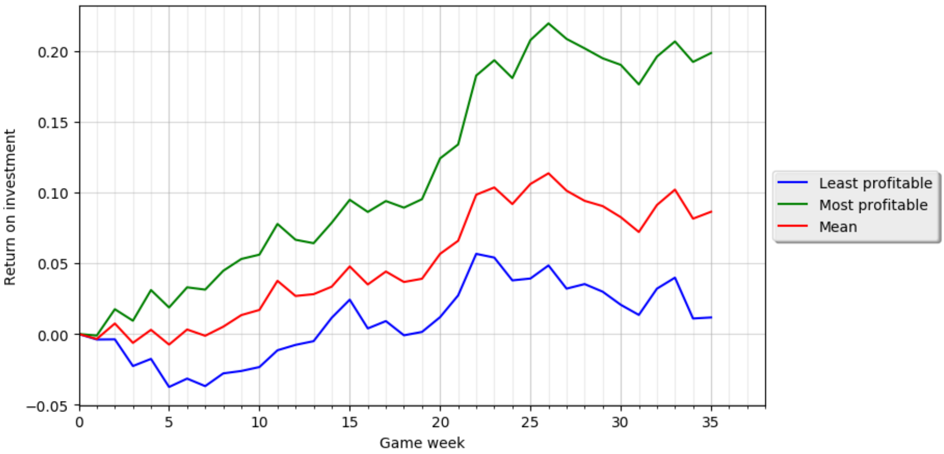


Figure 6.6.7: ROI over the span of the English Premier League season 2016-2017 using the team characteristics and strengths network and the fixed return strategy.

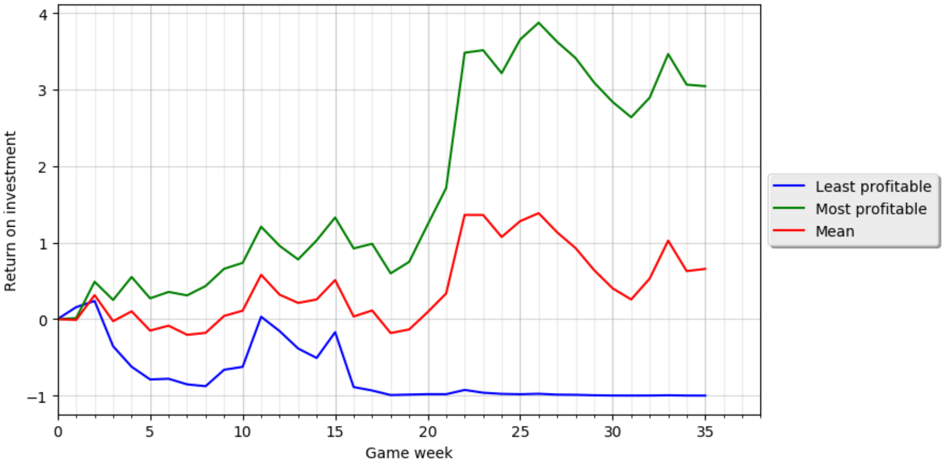


Figure 6.6.8: ROI over the span of the English Premier League season 2016-2017 using the team characteristics and strengths network and the Kelly ratio strategy.

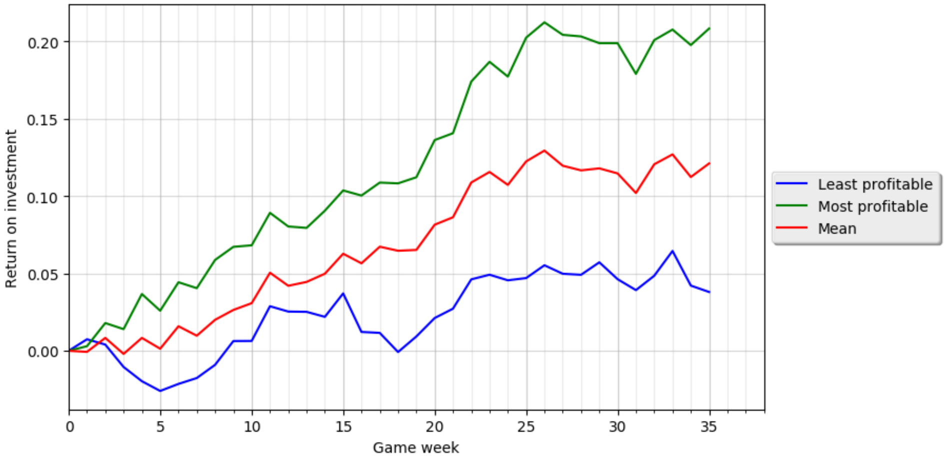


Figure 6.6.9: ROI over the span of the English Premier League season 2016-2017 using the team characteristics and strengths network and the variance adjusted strategy.

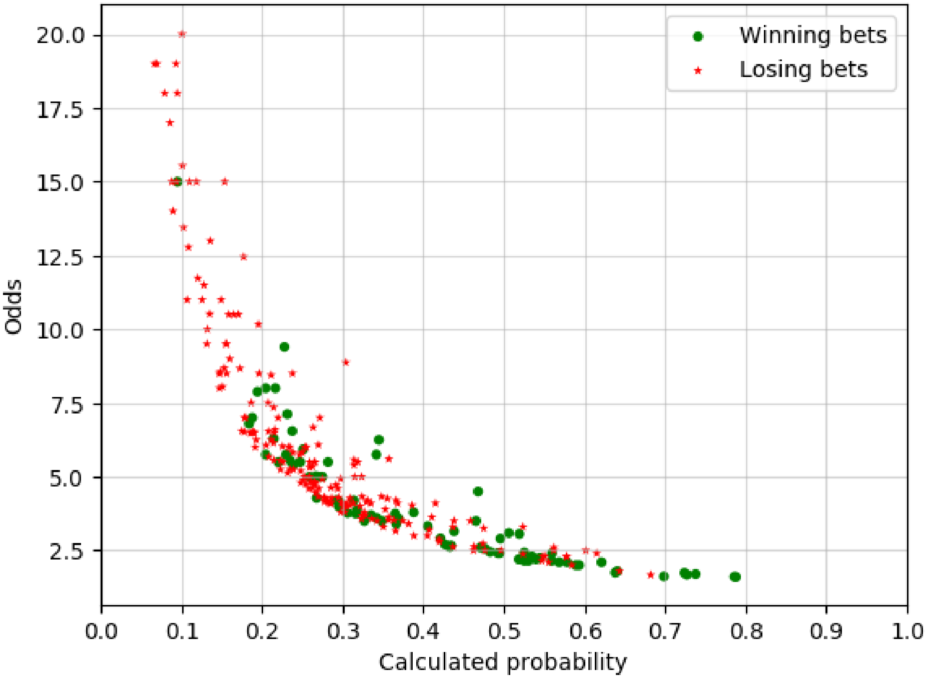


Figure 6.6.10: Offered odds and predicted probabilities for the bets placed during the 2016-2017 season of the English Premier League. The probabilities are generated by the team characteristics and strengths network.

Summary

Similarly to the team characteristics network, the team characteristics network achieved consistent good results. The network generated profits over both seasons using the fixed bet and Kelly ratio strategies. The network also generated profits the other strategies the first season.

Figures 6.6.5 and 6.6.10 show the connection between odds and probabilities predicted by the team characteristics and strengths network. The team characteristics and strengths has the same tendency to overestimate the probabilities in the lower half as the player ratings network. This is not surprising, as the two networks share features. Over the two seasons, the prediction models won approximately 28.4% of all bets placed, with an average odds of 3.67.

6.7 Comparing the results

Tables 6.7.1 and 6.7.2 show the average ROI for every combination of network and strategy for the two seasons.

Network	**Fixed bet**	**Fixed return**	**Kelly ratio**	**Variance adjusted**
Player ratings	0.068	-0.040	0.25	-0.073
Head to head	0.40	0.025	2.0	-0.0080
Previous meetings	0.25	-0.012	1.2	-0.052
Team characteristics	0.21	-0.050	0.80	-0.032
Team characteristics and strengths	0.12	-0.025	0.49	-0.052

Table 6.7.1: Comparison of the average ROI values for the networks. From the 2015-2016 season. Colored cells indicate profitable strategies.

Network	**Fixed bet**	**Fixed return**	**Kelly ratio**	**Variance adjusted**
Player ratings	-0.59	-0.095	-1.0	-0.053
Head to head	-0.20	-0.018	-1.0	-0.060
Previous meetings	-0.34	-0.050	-1.0	-0.022
Team characteristics	0.13	0.080	0.70	0.12
Team characteristics and strengths	0.17	0.080	0.80	0.13

Table 6.7.2: Comparison of the average ROI values for the networks. From the 2016-2017 season. Colored cells indicate profitable strategies.

Only two networks were able to generate a profit over both seasons: the networks based on team characteristics. The other networks produced varying results. The head to head network generated huge profits for the first season (40% on average using the fixed bet strategy, and 200% on average using the Kelly ratio strategy), while suffering losses the second season (20% on average using the fixed bet strategy, and going bankrupt for all instances using the Kelly ratio strategy).

The Kelly ratio strategy generated the highest profits, with good margin. The Kelly ratio strategy is, however, the only strategy that went bankrupt. When using the Kelly ratio strategy together with the player ratings network, all model instances were bankrupt within the eight first game weeks. The Kelly ratio strategy offers a high risk, high reward investment. The fixed return and variance adjusted strategies generated more stable results, losing at most 17%, and no more than 10% on average. The average profits were however not anywhere near the profits of the Kelly ratio strategy, with best average profits of 13% and 8%, respectively. The fixed bet strategy performed somewhere in-between the other strategies. For the worst performing networks, the fixed bet strategy suffered some huge losses. 78% of the initial bankroll was lost when used by the player ratings network during the 2016-2017 season. For the team characteristics based networks, the fixed bet strategy performed significantly better, losing at most 25% of the initial bankroll, with average profits of well above 10%.

6.8 Looking behind the results

The team characteristics network achieved the best overall results over the two seasons. The first season, it was clearly the most profitable network. The second season, it was barely beaten by the team characteristics and strengths network. By analyzing the odds-probability graphs for the team characteristics network, there are some clear patterns that might help increase the profitability further.

6.8.1 Placing no more than one bet per match

Table 6.8.1 shows all bets deemed feasible by a random instance of the team characteristics model during a random game week of the 2016-2017 season. Of the nine matches, a double bet was placed on four. That gives four bets that are guaranteed to fail. This pattern can be seen throughout all simulations, and is not specific for the highlighted game week alone.

To reduce the losses from placing more than one bet on a single match, one option is to only place the bet with the highest predicted probability. Table 6.8.2 shows how the average ROI values for the four strategies were affected when only placing the bet with the highest predicted probability. As the results show, only placing the ”safest” bet reduced the profitability of the network.

Another option is to only place the bet with the highest expected gain, $P_i * d_i$. Table 6.8.3 shows how the average ROI values for the four strategies were affected when only placing the bet with highest expected gain. Only placing the bet with highest expected gain increased the profitability of the network slightly.

Match	Predicted outcome	Odds	Probability
Tottenham 1 - 0 Sunderland	A	12.00	0.15
Crystal Palace 4 - 1 Stoke	A	4.20	0.35
Southampton 1 - 0 Swansea	A	6.17	0.28
Watford 3 - 1 Manchester United	H	6.00	0.25
Watford 3 - 1 Manchester United	D	4.20	0.28
Everton 3 - 1 Middlesbrough	A	6.19	0.19
Hull 1 - 4 Arsenal	H	6.50	0.25
Hull 1 - 4 Arsenal	D	4.20	0.29
Leicester 3 - 0 Burnley	D	4.60	0.27
Leicester 3 - 0 Burnley	A	8.87	0.32
Manchester City 4 - 0 Bournemouth	D	6.25	0.19
Manchester City 4 - 0 Bournemouth	A	12.78	0.16
Chelsea 1 - 2 Liverpool	H	2.30	0.54

Table 6.8.1: Bets deemed feasible by an instance of the team characteristics model. From game week 5 of the 2016-2017 season.

	2015-2016		2016-2017	
Strategy	**All feasible**	**Only best**	**All feasible**	**Only best**
Fixed bet	0.21	0.10	0.13	0.12
Fixed return	-0.050	-0.051	0.080	0.065
Kelly ratio	0.80	0.43	0.70	0.68
Variance Adjusted	-0.032	-0.051	0.12	0.090

Table 6.8.2: The effect of only allowing one bet per match. Only the bet with the highest predicted probability is placed. For the team characteristics network.

	2015-2016		2016-2017	
Strategy	**All feasible**	**Only best**	**All feasible**	**Only best**
Fixed bet	0.21	0.22	0.13	0.13
Fixed return	-0.050	-0.010	0.080	0.085
Kelly ratio	0.80	1.0	0.70	0.72
Variance Adjusted	-0.032	-0.030	0.12	0.14

Table 6.8.3: The effect of only allowing one bet per match. Only the bet with highest expected gain is placed. For the team characteristics network.

6.8.2 Setting an odds limit

By looking at Figure 6.5.5 and Figure 6.5.10, one can see that hardly any bets with odds above 13 are successful. That goes for both seasons. By rejecting bets with odds above 13, the profitability of the network is increased. Table 6.8.4 shows how the average ROI values for the four strategies were affected. Every strategy, over both seasons, had increased profits.

	2015-2016		**2016-2017**	
Strategy	**No limit**	**Max odds 13**	**No limit**	**Max odds 13**
Fixed bet	0.21	0.24	0.13	0.20
Fixed return	-0.050	-0.030	0.080	0.11
Kelly ratio	0.80	0.92	0.70	0.9
Variance Adjusted	-0.032	-0.028	0.12	0.13

Table 6.8.4: The effect of only allowing bets with odds less than 13. For the team characteristics network.

Chapter 7

Conclusion and future work

This chapter presents the conclusions drawn based on the results achieved in this report. The future work section presents suggestions for improvements of the prediction system.

7.1 Conclusion

This report presents a purely data-driven system for predicting football match outcomes, and for placing bets based on the predictions. The system stores large amounts of data that is easy to incorporate into the prediction model. The report shows that, even though football is a sport involving a lot of uncertainty and luck, it is possible to beat the betting market over the span of two consecutive seasons.

7.1.1 Model performance

The results from Chapter 6 show that the choice of model input is crucial to the model's accuracy and profitability. When comparing the head to head network with the previous meetings network, it becomes apparent that simple assumptions sometimes are enough, and that there might be no need to build a complex network. The same goes for the team characteristics network versus the team characteristics and strengths network.

The team characteristics model performed significantly better than most models presented in Section 2.1. Both Koopman and Lit (2015) and Rue and Salvesen (2000) generated profits well below the profits the team characteristics generated using the Kelly ratio strategy. However, when using the same strategies as the two other models, namely the fixed bet strategy and variance adjusted strategy, respectively, the team characteristics model did not perform as well. This highlights the importance of matching prediction model and betting strategy.

The model presented by A. C. Constantinou, N. E. Fenton, and Neil (2012) performed best of the models explored in Section 2.1. Using the same configuration: maximum odds and fixed bet strategy, the pi-football model performed significantly better than the team characteristics model (ROI of 0.83 versus 0.17). However, the team characteristics model performed better using the Kelly ratio strategy, achieving an average ROI of 0.75.

7.1.2 The betting strategies

The four betting strategies explored in this report can be compared to any other investment strategy. Strategies involving high risk have high potential gain, and at the same time higher chance of suffering losses. Strategies involving low risk have a lower probability of suffering losses, but at the same time lower potential gains. The Kelly ratio strategy is definitely the high-risk strategy, sometimes achieving impressive ROI, and sometimes going bankrupt. The fixed return strategy and variance adjusted strategy are more low-risk strategies, reducing the bet size for low-probability bets. The fixed bet strategy in an intermediate option, always placing bets of the same size. For the fixed bet, fixed return, and variance adjusted strategies to work, the prediction model needs to produce probabilities that closely match the implied probabilities of the bookmakers. The Kelly ratio strategy can to a greater extent survive on lucky strikes, generating a lot of profit.

7.1.3 Applying the predictions in the betting market

The most profitable prediction model, the team characteristics model, was able to generate an impressive average ROI of 0.75 over the two seasons using the Kelly ratio strategy. It should be noted that one instance of the network lost almost 50% of the initial investment over one season using the same strategy. The safer strategies, fixed return and variance adjusted, achieved average ROIs of 0.015 and 0.044, respectively. The intermediate strategy, fixed bet, achieved an average ROI of 0.17, while never losing more than 10% of the initial investment.

7.1.4 Fulfillment of goal, and answering research questions

Section 1.1 presented the goal of the report, along with two research questions stated in order to achieve the goal.

The first research question was to find out what a priori knowledge concerning a football can be fed to an ANN in order to predict the match outcome. Chapter 5 presents three data sources used for predicting match outcomes: player ratings, previous meetings between the teams, and team characteristics. The data sources are used to form five different ANNs. The data sources utilized in this report are only three of many sources, but as Chapter 6 shows, a single data source is enough to predict the outcome of a football match with up to 60% certainty.

The second research question asked how the outcome predictions can be used for generating a profit in betting. More specifically, how the choice of betting strategy affect the profitability of the network. Chapter 6 highlights the importance of matching prediction model and betting strategy correctly. Some combinations, as the team characteristics network and Kelly ratio strategy, can generate huge profits. Other combinations, on the other hand, as the player ratings network and fixed bet strategy, are catastrophic.

By answering the two research questions, the goal of the report has been fulfilled. The report shows that ANNs can indeed be used in order to profit from betting.

7.2 Future work

This section presents the author's suggestions for improvements of the prediction system.

7.2.1 Allowing different kinds of bets

As of now, the betting simulator only supports placing *1X2* type bets (bets on a single outcome).

Throughout the experiments in this report, bets have been placed on more than one outcome for several matches. This guarantees that at least one of the bets are unsuccessful. However, the odds are usually high enough to justify the choice.

Most bookmakers offer *double chance* bets. In double chance bets, one choose from three available bets, like in 1X2 bets, but with higher probabilities. The odds are, however, smaller than for single bets. The three available bets are a) home victory or draw b) away victory or draw, and c) home victory or away victory. Allowing double chance bets might increase the profitability of the prediction models, if the prediction model strongly suggests an outcome will **not** occur, and the odds suggest placing a double bet is better than two single bets.

Another bet type supported by most bookmakers is *draw no bet*. In draw no bet type bets, the money is refunded if the match ends with a draw. This might be an alternative to the double chance bets where the predicted probability of a draw is high enough, and the draw no bet odds offered are better than the double chance odds.

Systems with good RPS values will benefit from both double chance bets and draw no bet type bets. As mentioned in Section 2.3, the RPS system calculates the difference between the cumulative distributions of the predicted and observed probabilities. If the observed outcome is a home victory, predicting a draw is closer to the correct outcome than what predicting an away victory is. A prediction model that achieves good RPS values is therefore probably good at knowing what the final outcome will **not** be. This can be exploited if the corresponding double chance odds are high enough. Draw no bet type bets will be beneficial if the model overestimates the probability of drawn matches.

7.2.2 Considering betting biases

As mentioned in Section 2.4.3, bookmakers are biased. As of now, the betting simulator treats all bets equally, without taking any kind of bias into account.

To take the favorite-longshot into account, one can add a confidence threshold that increases with the odds. The same can be done with the predicted probability for the most-likely/least-likely bias. For the home-away bias, the required confidence needed to place a bet can be higher for away victories.

7.2.3 Include more bookmakers

As of now, the betting simulator only considers odds offered by seven bookmakers. This was mostly done for simplicity, as the historical match data presented at Football-Data (2016) only contains 1X2 odds from seven bookmakers. The data from Football-Data

(2016) was used as it is neatly stored in CSV files, and can easily be fetched without scraping any web page.

In the future, one should look into other web sites offering odds history, such as Odds-Portal (2016). Odds-Portal (2016) shows odds from more than 50 bookmakers (as of June 2017). For the match between Arsenal and Manchester United May 7, 2017, the Odds-Portal (2016) present odds from 55 bookmakers. The average odds offered were 2.09, 3.49, and 3.56 for home victory, draw, and away victory, respectively, whilst the highest odds were 2.16, 3.64, and 3.72.

Exploiting arbitrage opportunities

A way of securing extra profits is to exploit arbitrage opportunities. Arbitrage opportunities occur when the odds offered by different bookmakers vary enough to guarantee a profit. This method might go a bit beyond the scope of this report, as is does not require any prediction model. Nonetheless, it can still be used to increase the profitability of the system.

For the match between Arsenal and Everton May 21, 2017, the maximum available odds were 1.42, 6.20, 9.80, for home victory, draw, and away victory, respectively. The expected gain of the betting market was $\frac{1}{1.42} + \frac{1}{6.20} + \frac{1}{9.80} - 1 = -0,032$. The betting market was expected to suffer losses of 3.2%.

If the betting marked is expected to suffer losses, arbitrage is possible. Algorithm 2 shows how to secure profits from arbitrage.

Algorithm 2: How to secure profits from arbitrage

```
set desired total winnings, $C$;
for outcome $i \in \{0, 1, 2\}$ do
    place bet of size $C/d_i$ on outcome $i$
end
```

Table 7.2.1 shows how one could guarantee a profit for the match between Arsenal and Everton May 21, 2017. The desired total winnings, C, is set to 100 units.

	Home victory	**Draw**	**Away victory**
Best odds	1.42	6.20	9.80
Stake	$100/1.42 = 70.42$	$100/6.20 = 16.13$	$100/9.80 = 10.20$
Profit if win	29.58	83.88	90.16
Lost stakes	26.33	80.62	86.55
Total profit if win	3.25	3.25	3.61

Table 7.2.1: How to guarantee a profit for the match between Arsenal and Everton May 21, 2017.

7.2.4 Explore the impact of other available data

The prediction models constructed for this report utilizes three data sources: player ratings, previous meetings between the teams, and team characteristics. For future development of the prediction models, further exploring the data available at **www.whoscored.com** is a good start.

Section 2.2.3 presents interesting analyses of team characteristics. Bialkowski, Lucey, Carr, Yue, Sridharan, et al. (2014) mention how they in the future plan on using their findings for match outcome prediction. It would be interesting to explore the predictive properties the data from Figure 2.2.3 or Figure 2.2.4. Exploring the predictive properties of the ”significance” of different matches, as mentioned by J. Goddard and Asimakopoulos (2004), would also be interesting.

The team characteristics used in this report are all generated by the match events at **www.whoscored.com**. It would be interesting check if the events themselves hold better predicative information. By using the events themselves, the middle man is cut out. The underlying events might contain more predictive information than the aggregated characteristics.

References

[1] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, and Iain Matthews. “Win at home and draw away: Automatic formation analysis highlighting the differences in home and away team behaviors”. In: *Proceedings of 8th Annual MIT Sloan Sports Analytics Conference*. 2014.

[2] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, et al. “Identifying team style in soccer using formations learned from spatiotemporal tracking data”. In: *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE. 2014, pp. 9–14.

[3] Simon Borøy-Johnsen. *Using football match prediction to beat the bookmakers*. 2016.

[4] Léon Bottou. “Large-scale machine learning with stochastic gradient descent”. In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[5] A. Carron, T. M. Loughead, and S. R. Bray. “The home advantage in sport competitions: Courneya and Carron's (1992) conceptual framework a decade later”. In: *Journal of Sports Sciences* 23 (4 2005), pp. 395–407.

[6] M. Cattelan, C. Varin, and D. Firth. “Dynamic Bradley–Terry modelling of sports tournaments”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 62 (1 2013), pp. 135–150.

[7] Edgar F Codd. “Relational database: a practical foundation for productivity”. In: *Communications of the ACM* 25.2 (1982), pp. 109–117.

[8] A. C. Constantinou and N. E. Fenton. “Profiting from arbitrage and odds biases of the European football gambling market”. In: *Journal of Gambling Business and Economics* 7 (2 2013), pp. 41–70.

[9] A. C. Constantinou, N. E. Fenton, and M. Neil. “pi-football: A Bayesian network model for forecasting Association Football match outcomes”. In: *Knowledge-Based Systems* 36 (2012), pp. 322–339.

[10] Anthony C Constantinou, Norman E Fenton, et al. “Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models”. In: *Journal of Quantitative Analysis in Sports* 8.1 (2012), pp. 1559–0410.

[11] K. S. Courneya and A. Carron. “The Home Advantage in Sport Competitions: A Literature Review”. In: *Journal of Sport and Exercise Psychology* 14 (1 1992), pp. 13–27.

[12] DeepMind. *Solve intelligence. Use it to make the world a better place.* URL: `https://deepmind.com/about/` (visited on 05/25/2017).

[13] M. J. Dixon and S. G. Coles. “Modelling Association Football Scores and Inefficiencies in the Football Betting Market”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 46 (2 1997), pp. 265–280.

[14] M. Dixon and M. Robinson. “A birth process model for association football matches”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 47 (3 1998), pp. 523–538.

[15] Edward S Epstein. “A scoring system for probability forecasts of ranked categories”. In: *Journal of Applied Meteorology* 8.6 (1969), pp. 985–987.

[16] FIFA. *2014 FIFA World Cup Brazil™ in numbers*. 2014. URL: `http://www.fifa.com/worldcup/news/y=2014/m=9/news=2014-fifa-world-cup-braziltm-in-numbers-2443025.html` (visited on 12/14/2016).

[17] Football-Data. *Historical Football Results and Betting Odds Data*. URL: `http://football-data.co.uk/data.php` (visited on 12/07/2016).

[18] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks.” In: *Aistats*. Vol. 15. 106. 2011, p. 275.

[19] J. Goddard. “Who wins the football?” In: *Significance* 3 (1 2006), pp. 16–19.

[20] J. Goddard and I. Asimakopoulos. “Forecasting football results and the efficiency of fixed-odds betting”. In: *Journal of Forecasting* 23 (1 2004), pp. 51–66.

[21] John Goddard. “Regression models for forecasting goals and match results in association football”. In: *International Journal of forecasting* 21.2 (2005), pp. 331–340.

[22] Douglas M Hawkins. “The problem of overfitting”. In: *Journal of chemical information and computer sciences* 44.1 (2004), pp. 1–12.

[23] I. D. Hill. “Association Football and Statistical Inference”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 23 (2 1974), pp. 203–208.

[24] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. *Neural Networks for Machine Learning Lecture 6a Overview of mini–batch gradient descent.*

[25] N. Hirotsu and M. Wright. “An evaluation of characteristics of teams in association football by using a Markov process model”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 52 (4 2003), pp. 591–602.

[26] John J Hopfield. “Artificial neural networks”. In: *IEEE Circuits and Devices Magazine* 4.5 (1988), pp. 3–10.

[27] L. M. Hvattum and H. Arntzen. “Using ELO ratings for match result prediction in association football”. In: *International Journal of Forecasting* 26 (2010), pp. 460–470.

[28] D. Karlis and I. Ntzoufras. “Analysis of sports data by using bivariate Poisson models”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 52 (3 2003), pp. 381–393.

[29] Dimitris Karlis and Ioannis Ntzoufras. “Bayesian modelling of football outcomes: using the Skellam’s distribution for the goal difference”. In: *IMA Journal of Management Mathematics* 20.2 (2009), pp. 133–145.

[30] J. L. Kelly. “A new interpretation of information rate”. In: *IRE Transactions on Information Theory* 2 (3 1956), pp. 185–189.

[31] Keras. *Keras: Deep Learning library for Theano and TensorFlow*. URL: `https://keras.io/` (visited on 05/15/2017).

[32] Diederik Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[33] S. J. Koopman and R. Lit. “A dynamic bivariate Poisson model for analysing and forecasting match results in the English Premier League”. In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 178 (1 2015), pp. 167–186.

[34] H. W. Kuhn. “The Hungarian Method for the assignment problem”. In: *Naval Research Logistics Quarterly* 2 (1955), pp. 83–97.

[35] Helge Langseth. “Beating the bookie: A look at statistical models for prediction of football matches.” In: *SCAI*. 2013, pp. 165–174.

[36] Yongjin Lee and Seungjin Choi. “Minimum entropy, k-means, spectral clustering”. In: *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*. Vol. 1. IEEE. 2004, pp. 117–122.

[37] Todd M Loughead et al. “Facility familiarity and the home advantage in professional sports”. In: *International Journal of Sport and Exercise Psychology* 1.3 (2003), pp. 264–274.

[38] M. J. Maher. “Modelling association football scores”. In: *Statistica Neerlandica* 36 (3 1982), pp. 109–118.

[39] Maluuba. *News about Microsoft acquisition and investments in our Montreal AI lab*. URL: `http://www.maluuba.com/acquisition/` (visited on 05/25/2017).

[40] I. McHale and P. Scarf. “Modelling soccer matches using bivariate discrete distributions with general dependence structure”. In: *Statistica Neerlandica* 61 (4 2007), pp. 432–445.

[41] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[42] Alan M Nevill, Nigel J Balmer, and A Mark Williams. “The influence of crowd noise and experience upon refereeing decisions in football”. In: *Psychology of Sport and Exercise* 3.4 (2002), pp. 261–272.

[43] Alan M Nevill, Sue M Newell, and Sally Gale. “Factors associated with home advantage in English and Scottish soccer matches”. In: *Journal of Sports Sciences* 14.2 (1996), pp. 181–186.

[44] William S Noble. “What is a support vector machine?” In: *Nature biotechnology* 24.12 (2006), pp. 1565–1567.

[45] Odds-Portal. *Odds Portal: Odds Comparison, Sports Betting Odds*. 2016. URL: http://www.oddsportal.com/ (visited on 12/14/2016).

[46] Leif E Peterson. “K-nearest neighbor”. In: *Scholarpedia* 4.2 (2009), p. 1883.

[47] R. Pollard. “Home Advantage in Football: A Current Review of an Unsolved Puzzle”. In: *The Open Sports Sciences Journal* 1 (2008), pp. 12–14.

[48] R. Pollard and C. Reep. “Measuring the effectiveness of playing strategies at soccer”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 46 (4 1997), pp. 541–550.

[49] D. Ross. *Arpad Elo and the Elo Rating System*. 2007. URL: http://en.chessbase.com/post/arpad-elo-and-the-elo-rating-system (visited on 12/05/2016).

[50] H. Rue and Ø. Salvesen. “Prediction and Retrospective Analysis of Soccer Matches in a League”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 49 (3 2000), pp. 399–418.

[51] G. Shahtahmassebi and R. Moyeed. “An application of the generalized Poisson difference distribution to the Bayesian modelling of football scores”. In: *Statistica Neerlandica* 70 (3 2016), pp. 260–273.

[52] Nitish Srivastava et al. “Dropout: A simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[53] TensorFlow. *An open-source software library for Machine Intelligence*. URL: https://www.tensorflow.org/ (visited on 05/15/2017).

[54] Nikolaos Vlastakis, George Dotsis, and Raphael N Markellos. “How efficient is the European football betting market? Evidence from arbitrage and trading strategies”. In: *Journal of Forecasting* 28.5 (2009), pp. 426–444.

[55] WhoScored.com. *WhoScored Ratings Explained*. URL: https://www.whoscored.com/Explanations (visited on 05/25/2017).

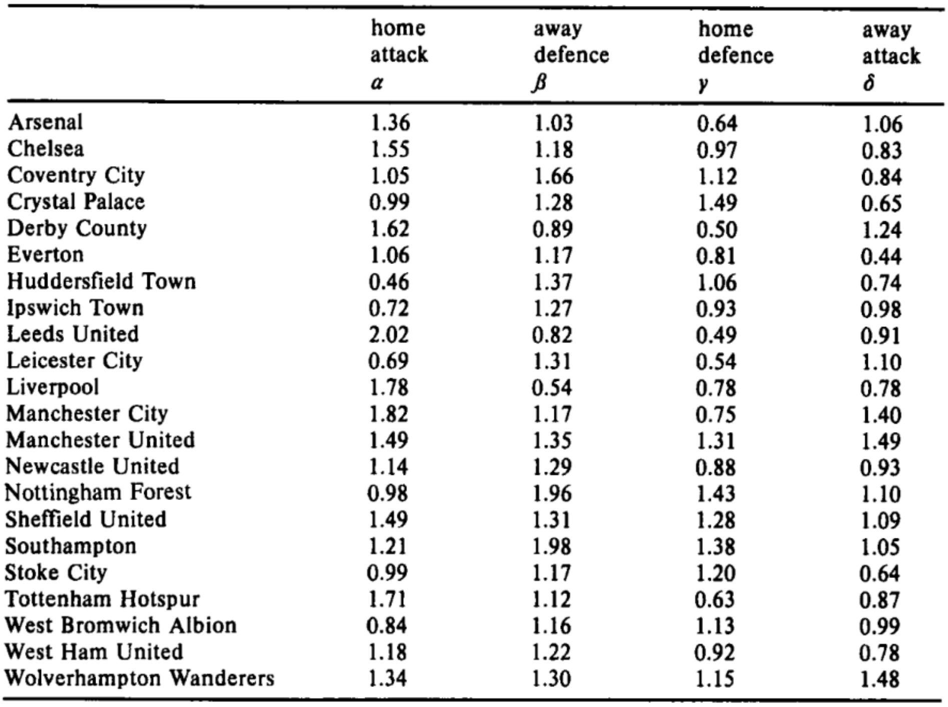
[56] Richard Williams et al. “Generalized ordered logit/partial proportional odds models for ordinal dependent variables”. In: *Stata Journal* 6.1 (2006), p. 58.

Appendices

Appendix A

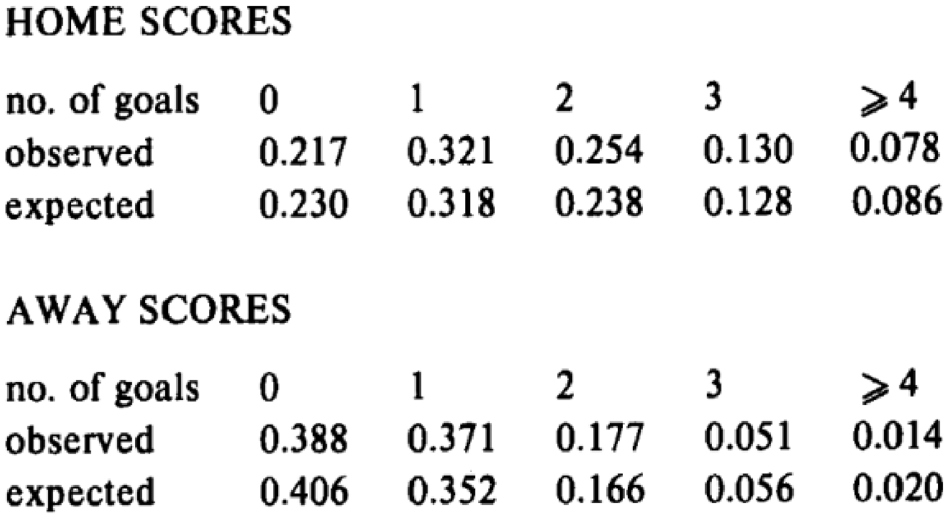
Results from previous models

A.1 Maher (1982)



	home attack α	away defence β	home defence γ	away attack δ
Arsenal	1.36	1.03	0.64	1.06
Chelsea	1.55	1.18	0.97	0.83
Coventry City	1.05	1.66	1.12	0.84
Crystal Palace	0.99	1.28	1.49	0.65
Derby County	1.62	0.89	0.50	1.24
Everton	1.06	1.17	0.81	0.44
Huddersfield Town	0.46	1.37	1.06	0.74
Ipswich Town	0.72	1.27	0.93	0.98
Leeds United	2.02	0.82	0.49	0.91
Leicester City	0.69	1.31	0.54	1.10
Liverpool	1.78	0.54	0.78	0.78
Manchester City	1.82	1.17	0.75	1.40
Manchester United	1.49	1.35	1.31	1.49
Newcastle United	1.14	1.29	0.88	0.93
Nottingham Forest	0.98	1.96	1.43	1.10
Sheffield United	1.49	1.31	1.28	1.09
Southampton	1.21	1.98	1.38	1.05
Stoke City	0.99	1.17	1.20	0.64
Tottenham Hotspur	1.71	1.12	0.63	0.87
West Bromwich Albion	0.84	1.16	1.13	0.99
West Ham United	1.18	1.22	0.92	0.78
Wolverhampton Wanderers	1.34	1.30	1.15	1.48

Figure A.1.1: Maximum likelihood estimates of the parameters for the English Division 1 1971-1972. Taken from Maher (1982).



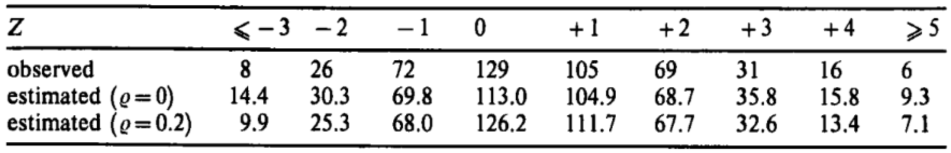
HOME SCORES

no. of goals	0	1	2	3	⩾4
observed	0.217	0.321	0.254	0.130	0.078
expected	0.230	0.318	0.238	0.128	0.086

AWAY SCORES

no. of goals	0	1	2	3	⩾4
observed	0.388	0.371	0.177	0.051	0.014
expected	0.406	0.352	0.166	0.056	0.020

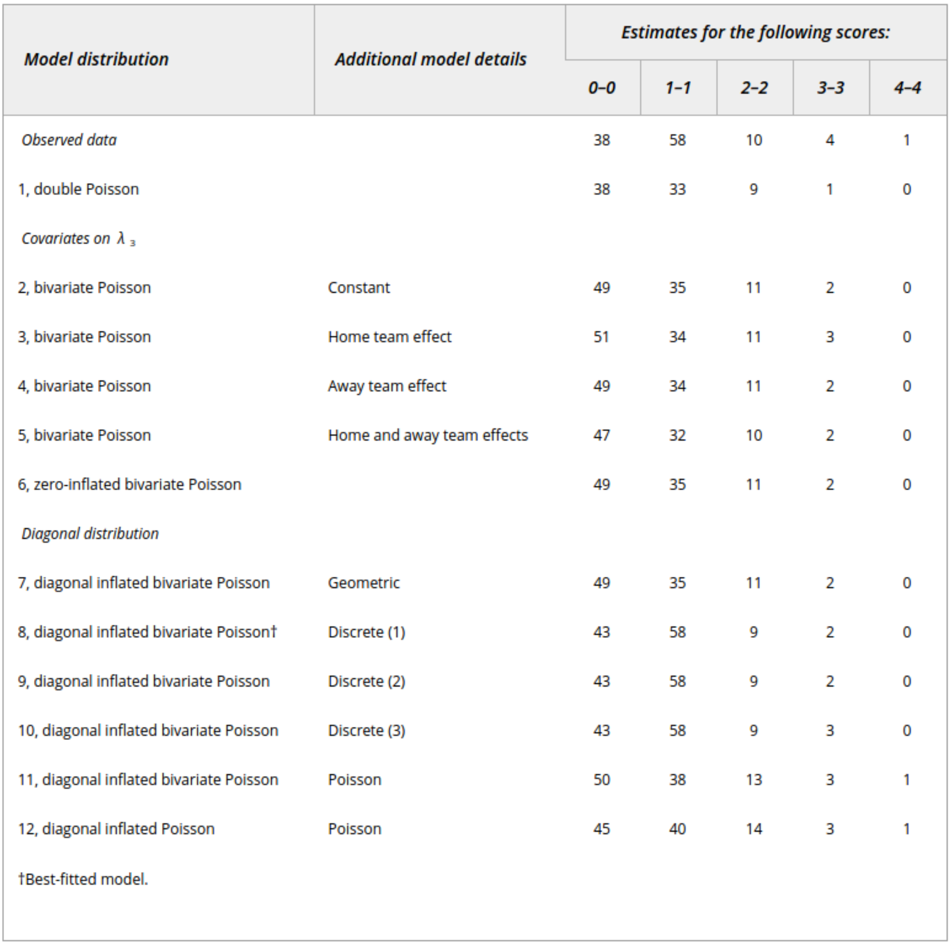
Figure A.1.2: Comparison of expected and observed number of goals scored for home and away teams for the English Division 1 1971-1972. Taken from Maher (1982).



Z	⩽−3	−2	−1	0	+1	+2	+3	+4	⩾5
observed	8	26	72	129	105	69	31	16	6
estimated ($\varrho=0$)	14.4	30.3	69.8	113.0	104.9	68.7	35.8	15.8	9.3
estimated ($\varrho=0.2$)	9.9	25.3	68.0	126.2	111.7	67.7	32.6	13.4	7.1

Figure A.1.3: Observed and estimated frequencies for Z, the goal difference, for the English Division 1 1971-1972 using different values for ϱ. Taken from Maher (1982).

A.2 D. Karlis and I. Ntzoufras (2003)



Model distribution	***Additional model details***	***Estimates for the following scores:***				
		0–0	***1–1***	***2–2***	***3–3***	***4–4***
Observed data		38	58	10	4	1
1, double Poisson		38	33	9	1	0
Covariates on λ_3						
2, bivariate Poisson	Constant	49	35	11	2	0
3, bivariate Poisson	Home team effect	51	34	11	3	0
4, bivariate Poisson	Away team effect	49	34	11	2	0
5, bivariate Poisson	Home and away team effects	47	32	10	2	0
6, zero-inflated bivariate Poisson		49	35	11	2	0
Diagonal distribution						
7, diagonal inflated bivariate Poisson	Geometric	49	35	11	2	0
8, diagonal inflated bivariate Poisson†	Discrete (1)	43	58	9	2	0
9, diagonal inflated bivariate Poisson	Discrete (2)	43	58	9	2	0
10, diagonal inflated bivariate Poisson	Discrete (3)	43	58	9	3	0
11, diagonal inflated bivariate Poisson	Poisson	50	38	13	3	1
12, diagonal inflated Poisson	Poisson	45	40	14	3	1

†Best-fitted model.

Figure A.2.1: The estimates by different versions of the model from D. Karlis and I. Ntzoufras (2003). Taken from D. Karlis and I. Ntzoufras (2003).

A.3 Koopman and Lit (2015)

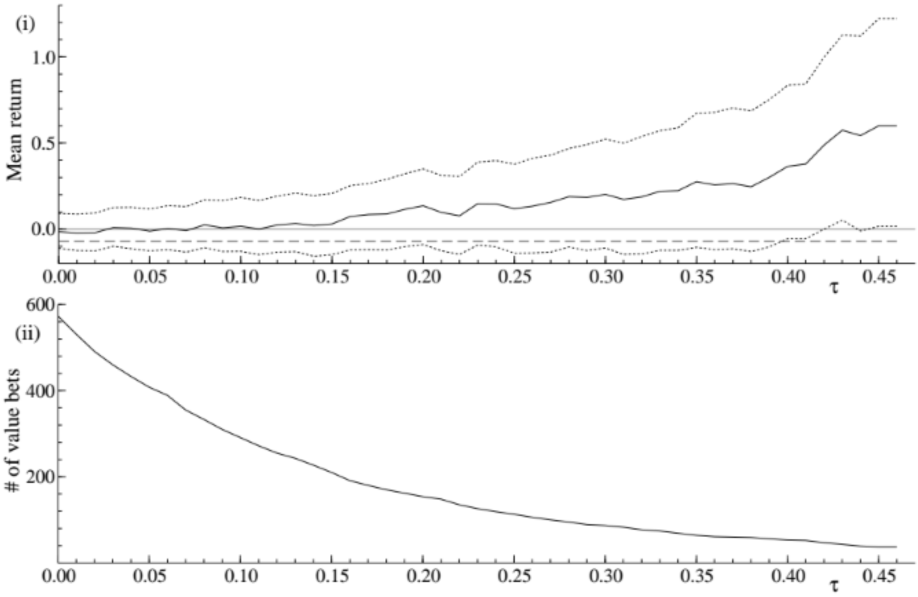


Figure A.3.1: (i) The average return from betting on match outcomes in the English Premier League 2010-2012 using different values for τ. Plotted together with 90% bootstrap confidence intervals. (ii) The number of feasible bets for different values of τ. Taken from Koopman and Lit (2015)

A.4 Rue and Salvesen (2000)

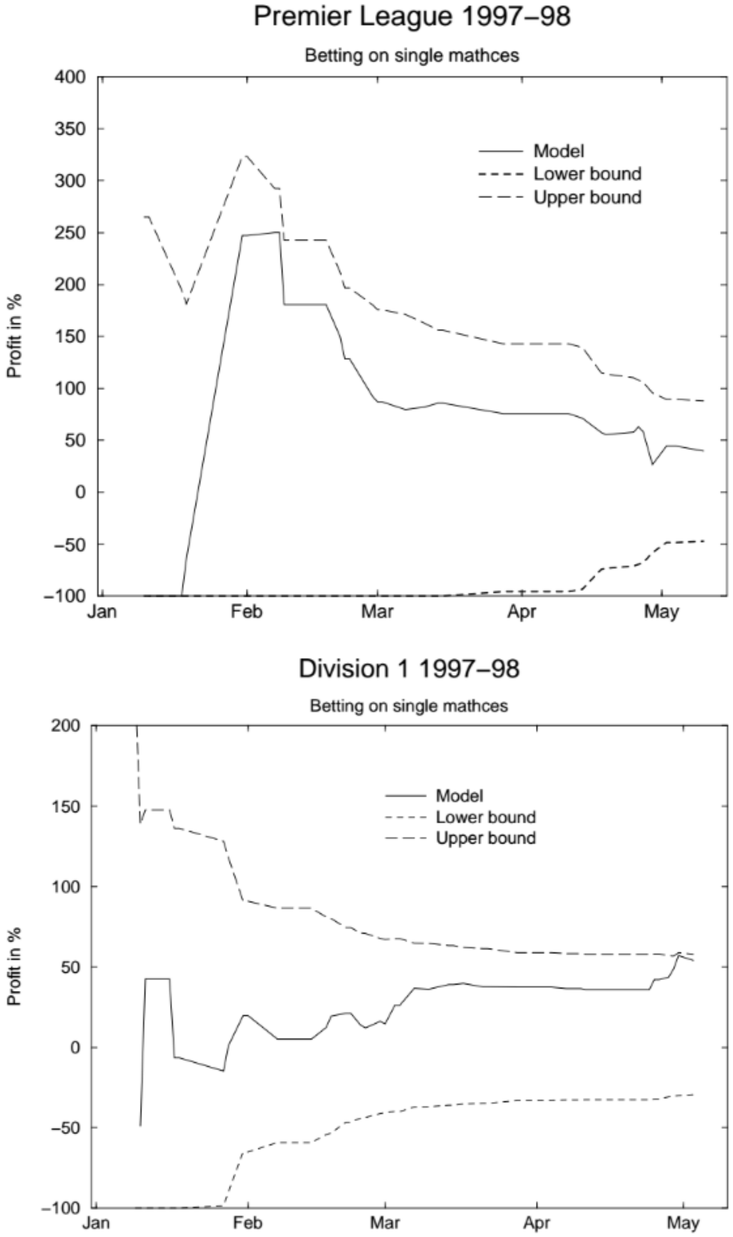


Figure A.4.1: The observed profit in the simulated betting experiments for the English Premier League and Division 1 1997-1998. Taken from Rue and Salvesen (2000).

A.5 Hvattum and Arntzen (2010)

	#BETS	UNIT BET		UNIT WIN		KELLY	
		BS	TROB	BS	TROB	BS	TROB
UNI	27 290	1.000	0.935	0.371	0.940	0.086	0.922
FRQ	16 892	1.000	0.928	0.448	0.938	0.090	0.915
GOD$_{\mathbf{b}}$	13 644	1.000	0.912	0.505	0.920	0.052	0.898
GOD$_{\mathbf{g}}$	14 161	1.000	0.945	0.535	0.947	0.056	0.927
ELO$_{\mathbf{b}}$	12 142	1.000	0.930	0.559	0.943	0.041	0.924
ELO$_{\mathbf{g}}$	12 152	1.000	0.939	0.568	0.954	0.040	0.940
AVG	5 594	1.000	0.954	0.318	0.967	0.009	0.946

Figure A.5.1: Average bet size (BS) and total return on bets (TROB) based on simulated betting on 14,927 matches from the English league system, using seven different betting strategies. Here, UNIT BET represents Fixed bet, UNIT WIN Fixed return, and KELLY the Kelly ratio strategy. Taken from Hvattum and Arntzen (2010).

A.6 A. C. Constantinou, N. E. Fenton, and Neil (2012)

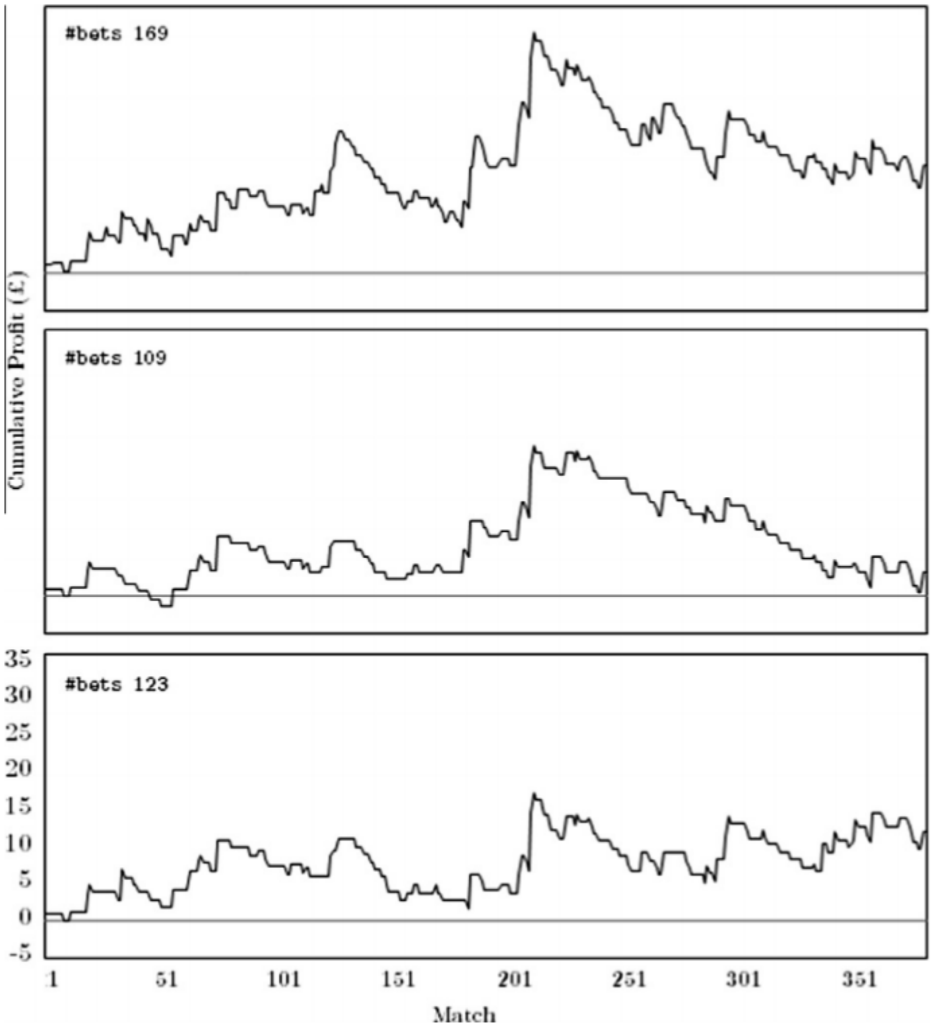
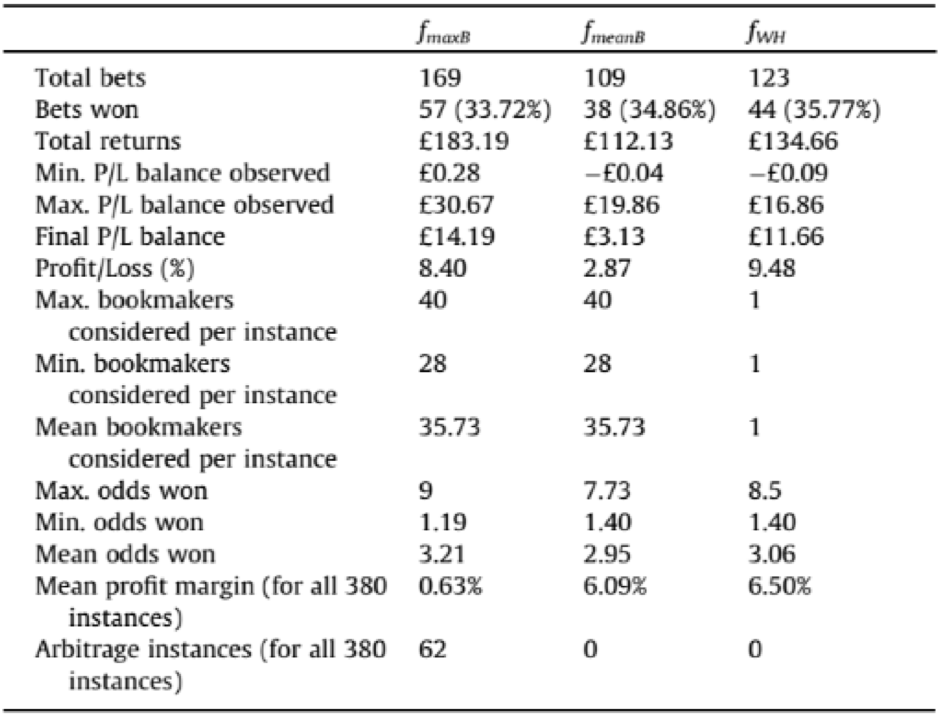


Figure A.6.1: Cumulative profits observed when simulating the Unit bet strategy at discrepancy levels of $\geq 5\%$ against (a) f_{maxB}, (b) f_{meanB}, and (c) f_{WH}. Taken from A. C. Constantinou, N. E. Fenton, and Neil (2012).



	f_{maxB}	f_{meanB}	f_{WH}
Total bets	169	109	123
Bets won	57 (33.72%)	38 (34.86%)	44 (35.77%)
Total returns	£183.19	£112.13	£134.66
Min. P/L balance observed	£0.28	−£0.04	−£0.09
Max. P/L balance observed	£30.67	£19.86	£16.86
Final P/L balance	£14.19	£3.13	£11.66
Profit/Loss (%)	8.40	2.87	9.48
Max. bookmakers considered per instance	40	40	1
Min. bookmakers considered per instance	28	28	1
Mean bookmakers considered per instance	35.73	35.73	1
Max. odds won	9	7.73	8.5
Min. odds won	1.19	1.40	1.40
Mean odds won	3.21	2.95	3.06
Mean profit margin (for all 380 instances)	0.63%	6.09%	6.50%
Arbitrage instances (for all 380 instances)	62	0	0

Figure A.6.2: Betting simulation stats against (a) f_{maxB}, (b) f_{meanB}, and (c) f_{WH} at discrepancy levels of $\geq 5\%$. Taken from A. C. Constantinou, N. E. Fenton, and Neil (2012).

Appendix B

www.whoscored.com data

B.1 Match events

Name	**Description**
Start	The match started
End	The match ended
Formation set	Initial formation information for a team
Corner awarded	A team was awarded a corner
Offside provoked	A team was caught offside
Formation change	A team made a change in their formation
Turnover	A team made a turnover
Cross not claimed	A cross was not claimed by the team
Shield ball opp	The opposing team shielded the ball
Save	A keeper made a save
Saved shot	A keeper saved a shot
Penalty faced	A keeper faced a penalty
Keeper sweeper	A keeper swept the ball
Claim	A keeper claimed the ball
Punch	A keeper punched the ball
Smother	A keeper made a smother save
Keeper pickup	A keeper picked up the ball
Pass	A player made a pass
Goal	A player scored a goal
Ball recovery	A player recovered the ball from the opposing team
Ball touch	A player made a touch on the ball
Take on	A player took on another player
Tackle	A player made a tackle
Dispossessed	A player lost possession of the ball
Interception	A player intercepted a pass
Clearance	A player made a clearance
Blocked pass	A player blocked a pass
Aerial	A player went into an aerial duel
Foul	A player made a foul
Missed shots	A player missed a shot
Challenge	A player challenged another player for the ball
Card	A player was awarded a card
Error	A player made en error
Substitution off	A player was substituted off
Substitution on	A player was substituted on
Shot on post	A player hit a goal post with a shot
Good skill	A player showed good skills
Chance missed	A player missed a chance
Offside pass	A pass offside provoking was made

Table B.1.1: Different event types in the detailed matches at **www.whoscored.com**.

Name	**Description**	**Mandatory?**
ID	Unique event ID	✗
Event ID	ID relative to the match	✗
Minute	What minute the event occurs	✓
Second	What second the event occurs	✓
Player ID	ID of the player the event concerns	✗
Related player ID	ID of other player the event concerns	✗
X	X coordinate of where the event occurs	✗
Y	Y coordinate of where the event occurs	✗
End x	X coordinate of where the event ends	✗
End y	Y coordinate of where the event ends	✗
Goal mouth x	X coordinate of where the ball entered the goal	✗
Goal mouth y	Y coordinate of where the ball entered the goal	✗
Period	What match half the event occurs	✗
Type	What type of event it is (Table B.1.1)	✗
Outcome	Whether the event is successful or unsuccessful	✗
Qualifiers	List of different event properties	✓
Is touch	Whether event is a touch	✗
Is goal	Whether event is a goal	✗
Is own goal	Whether event is an own goal	✗
Is penalty	Whether event is a penalty	✗
Is touch	Whether event is a touch	✗

Table B.1.2: Properties of a detailed event at **www.whoscored.com**.

B.2 Player metrics

Name	**Description**
Tackles successful	Number of successful tackles
Tackles unsuccessful	Number of unsuccessful tackles
Tackles total	Total number of tackles
Shots on target	Number of shots on target
Shots off target	Number of shots off target
Shots blocked	Number of shots blocked
Shots on post	Number of shots to hit a goal post
Shots total	Total number of shots
Dribbles won	Number of dribbles won
Dribbles lost	Number of dribbles lost
Dribbles attempted	Total number of dribbles attempted
Dribbled past	Number of times the player was dribbled past
Passes accurate	Number of accurate passes
Passes key	Number of key passes
Passes total	Total number of passes
Touches	Number of touches
Possessions	Number of times the player was in possession of the ball
Interceptions	Number of interceptions made
Fouls committed	Number of fouls committed
Claims high	Number of claims made high on the pitch
Clearances	Number of clearances made
Parried safe	Number of times the player successfully parried the ball
Parried danger	Number of times the player parried the ball into a dangerous situation
Errors	Number of errors made
Dispossessed	Number of times the player lost the ball
Offsides caught	Number of times the player was caught offside
Corners accurate	Number of accurate corners kicked
Corners total	Total number of corners kicked
Collected	Number of times the player collected the ball
Total saves	Total number of saves made
Offensive aerials	Number of offensive aerials participated in
Defensive aerials	Number of defensive aerials participated in
Aerials won	Number of aerials won
Aerials total	Total number of aerials

Table B.2.1: List of player metrics in the detailed matches at **www.whoscored.com**.

B.3 Views

URL post-fix	Description
Preview	Probable lineups, missing players, and top players.
Show	Last matches between the competing teams. Current table positions (if applicable). Last matches for the competing teams. Team characteristics at the time of the match.
TeamStatistics	Team characteristics at the time of the match (the same as in the `Live` view). Per-match statistics, listing average number of goals, shots, cards, corners, fouls, etc. for the competing teams. Situational statistics, listing summary of goals for/against the teams, passing history, and cards history. Positional statistics, highlighting where on the pitch the teams usually play and shoot.
PlayerStatistics	Height, weight, matches played, minutes played, ratings, and numbers from Table B.2.1 for the players in the competing teams' squads.
Betting	Odds (if match is not finished). Summary of the teams' recent results. Summary of goals scored by the teams (how many, scoring times, and number of clean sheets and goal-less games).
Live	”Match Centre”: overview of the players, with their positions, substitutions, final ratings, and most important events. ”Match Commentary”: timeline containing textual descriptions of how the match progressed. ”Chalkboard”: spatiotemporal overview over player events (shots, passes, dribbles, clearances, saves, etc.). ”Heatmaps”: heatmaps, highlighting where on the pitch the different players contributed to the match. ”Live Stream”: live stream of the match (if available)
MatchReport	Textual summary of the teams' strengths, weaknesses, and styles. Situational report, listing the number of attempts, passes, and card situations.
LiveStream	Live stream of the match (if available).

Table B.3.1: List of different views for matches at **www.whoscored.com**.

URL post-fix	Description
Show	List of recent matches. List of coming matches. Current squad, with summary of the numbers from Table B.2.1 for each player. Team characteristics. Top players. Formation summary (seasonal, last match, and a list of what positions the current squad members have played).
Fixtures	List of the team's matches the current season.
Statistics	Match statistics (average number of cards, possession, passes, shots, tackles, etc.). Situational statistics (where the team usually scores from, what kind of passes are usually made, and how the team is usually awarded cards). Positional statistics, including where on the pitch the team usually plays and where on the pitch the team usually shoot for the goal.
RefereeStatistics	List of all referees that have conducted the team, including number of matches the have conducted, cards they have issued, and penalties they have awarded.
History	List of team squads, with summary of the numbers from Table B.2.1 for each player. Option show statistics from different seasons.

Table B.3.2: List of different views for teams at **www.whoscored.com**.

URL post-fix	Description
Show	Player information (name, height, weight, full name, age, nationality, current team, shirt number, positions). Tournaments the player currently participates in, with summary of the numbers from Table B.2.1. List of different playing positions the player has had. Player characteristics. List of the last 10 matches the player participated in.
Fixtures	List of matches the current season where the player has participated.
History	Summary of the player's career, including competitions, matches played, minutes played, and the numbers from Table B.2.1

Table B.3.3: List of different views for players at **www.whoscored.com**.

B.4 Match header

Description
Home team ID
Away team ID
Home team name
Away team name
Kickoff time
Start of match date
Match status
Elapsed time (HT for half time, FT for full time, EAT for extra after time, PEN for after penalties)
Half time score
Full time score
Score after extra time
Score after penalties
Final score
Home team country
Away team country

Table B.4.1: Ordered list of fields in **www.whoscored.com** match header.